INTRODUCTION TO STATISTICS WITH R



Edward Chi
Cerritos College



Note to Students and Instructors

April 27, 2023

Dear Students and Instructors,

This textbook is an initial attempt at creating a free introduction to statistics textbook that incorporates the free and popular statistical software, R. This book is a collection of sections pulled from other free textbooks published on LibreTexts. Because some sections are from one author's book and other sections are from another author's book, there are some inconsistencies. I apologize in advance for the confusion this will cause and thank you for your understanding.

Sincerely,

W. Edward Chi

Introduction to Statistics with R

This text is disseminated via the Open Education Resource (OER) LibreTexts Project (https://LibreTexts.org) and like the thousands of other texts available within this powerful platform, it is freely available for reading, printing, and "consuming."

The LibreTexts mission is to bring together students, faculty, and scholars in a collaborative effort to provide an accessible, and comprehensive platform that empowers our community to develop, curate, adapt, and adopt openly licensed resources and technologies; through these efforts we can reduce the financial burden born from traditional educational resource costs, ensuring education is more accessible for students and communities worldwide.

Most, but not all, pages in the library have licenses that may allow individuals to make changes, save, and print this book. Carefully consult the applicable license(s) before pursuing such effects. Instructors can adopt existing LibreTexts texts or Remix them to quickly build course-specific resources to meet the needs of their students. Unlike traditional textbooks, LibreTexts' web based origins allow powerful integration of advanced features and new technologies to support learning.



LibreTexts is the adaptable, user-friendly non-profit open education resource platform that educators trust for creating, customizing, and sharing accessible, interactive textbooks, adaptive homework, and ancillary materials. We collaborate with individuals and organizations to champion open education initiatives, support institutional publishing programs, drive curriculum development projects, and more.

The LibreTexts libraries are Powered by NICE CXone Expert and was supported by the Department of Education Open Textbook Pilot Project, the California Education Learning Lab, the UC Davis Office of the Provost, the UC Davis Library, the California State University Affordable Learning Solutions Program, and Merlot. This material is based upon work supported by the National Science Foundation under Grant No. 1246120, 1525057, and 1413739.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation nor the US Department of Education.

Have questions or comments? For information about adoptions or adaptions contact info@LibreTexts.org or visit our main website at https://LibreTexts.org.

This text was compiled on 04/19/2026

TABLE OF CONTENTS

Note to Students and Instructors

Licensing

1: Basics

- 1.1: Introduction
  - 1.1.1: What Is Statistical Thinking?
  - 1.1.2: Dealing with Statistics Anxiety
  - 1.1.3: What Can Statistics Do for Us?
  - 1.1.4: The Big Ideas of Statistics
  - 1.1.5: Causality and Statistics
- 1.2: Working with Data
  - 1.2.1: What Are Data?
  - 1.2.2: Data Basics
  - 1.2.3: Scales of Measurement
  - 1.2.4: What Makes a Good Measurement?
  - 1.2.5: Overview of Data Collection Principles
  - 1.2.6: Observational Studies and Sampling Strategies
  - 1.2.7: Experiments
  - 1.2.8: How Not to Do Statistics
  - 1.2.9: Exercises

2: Introduction to R

- 2.1: Why Programming Is Hard to Learn
- 2.2: Using RStudio
- 2.3: Installing R
- 2.4: Getting Started with R
- 2.5: Variables
- 2.6: Functions
- 2.7: Letting RStudio Help You with Your Commands
- 2.8: Vectors
- 2.9: Math with Vectors
- 2.10: Data Frames
- 2.11: Using R Libraries
- 2.12: Installing and Loading Packages
- 2.13: Using Comments
- 2.14: Navigating the File System
- 2.15: Loading and Saving Data
- 2.16: Useful Things to Know about Variables
- 2.17: Factors
- 2.18: Data frames
- 2.19: Suggested Readings and Videos

3: Summarizing Data Visually

- 3.1: Qualitative Data
- 3.2: Quantitative Data
- 3.3: Other Graphical Representations of Data
- 3.4: Statistical Literacy

4: Summarizing Data Visually Using R

- 4.1: An Overview of R Graphics
- 4.2: An Introduction to Plotting
- 4.3: Histograms
- 4.4: Stem and Leaf Plots
- 4.5: Scatterplots
- 4.6: Bar Graphs
- 4.7: Saving Image Files Using R and Rstudio
- 4.8: Summary

5: Summarizing Data With Numbers

- 5.1: Central Tendency
- 5.2: What is Central Tendency
- 5.3: Measures of Central Tendency
- 5.4: Median and Mean
- 5.5: Measures of the Location of the Data
- 5.6: Additional Measures
- 5.7: Comparing Measures
- 5.8: Variability
- 5.9: Measures of Variability
- 5.10: Shapes of Distributions
- 5.11: Effects of Linear Transformations
- 5.12: Variance Sum Law I - Uncorrelated Variables
- 5.13: Statistical Literacy
- 5.14: Case Study- Using Stents to Prevent Strokes
- 5.15: Measures of the Location of the Data (Exercises)
- 5.E: Summarizing Distributions (Exercises)

6: Describing Data With Numbers Using R

- 6.1: Measures of Central Tendency
- 6.2: Measures of Variability
- 6.3: Skew and Kurtosis
- 6.4: Getting an Overall Summary of a Variable
- 6.5: Descriptive Statistics Separately for each Group
- 6.6: Standard Scores
- 6.7: Epilogue- Good Descriptive Statistics Are Descriptive!

7: Introduction to Probability

- 7.1: How are Probability and Statistics Different?
- 7.2: What Does Probability Mean?
- 7.3: Basic Probability Theory

- 7.4: The Binomial Distribution
- 7.5: The Normal Distribution
- 7.6: Other Useful Distributions
- 7.7: Summary
- 7.8: Statistical Literacy
- 7.E: Probability (Exercises)

8: Estimating Unknown Quantities from a Sample

- 8.1: Samples, Populations and Sampling
- 8.2: The Law of Large Numbers
- 8.3: Sampling Distributions and the Central Limit Theorem
- 8.4: Estimating Population Parameters
- 8.5: Estimating a Confidence Interval
- 8.6: Summary
- 8.7: Statistical Literacy
- 8.E: Estimation (Exercises)

9: Hypothesis Testing

- 9.1: A Menagerie of Hypotheses
- 9.2: Two Types of Errors
- 9.3: Test Statistics and Sampling Distributions
- 9.4: Making Decisions
- 9.5: The p value of a test
- 9.6: Reporting the Results of a Hypothesis Test
- 9.7: Running the Hypothesis Test in Practice
- 9.8: Effect Size, Sample Size and Power
- 9.9: Some Issues to Consider
- 9.10: Misconceptions of Hypothesis Testing
- 9.11: Summary
- 9.12: Statistical Literacy
- 9.13: Logic of Hypothesis Testing (Exercises)

10: Categorical Data Analysis

- 10.1: The χ2 Goodness-of-fit Test
- 10.2: The χ2 test of independence (or association)
- 10.3: The Continuity Correction
- 10.4: Effect Size
- 10.5: Assumptions of the Test(s)
- 10.6: The Most Typical Way to Do Chi-square Tests in R
- 10.7: The Fisher Exact Test
- 10.8: The McNemar Test
- 10.9: What's the Difference Between McNemar and Independence?
- 10.10: Summary
- 10.11: Statistical Literacy
- 10.12: Chi Square (Exercises)

11: Comparing Two Means

- 11.1: The one-sample z-test
- 11.2: The One-sample t-test
- 11.3: The Independent Samples t-test (Student Test)
- 11.4: The Independent Samples t-test (Welch Test)
- 11.5: The Paired-samples t-test
- 11.6: One Sided Tests
- 11.7: Using the t.test() Function
- 11.8: Effect Size
- 11.9: Checking the Normality of a Sample
- 11.10: Testing Non-normal Data with Wilcoxon Tests
- 11.11: Summary
- 11.12: Statistical Literacy
- 11.E: Tests of Means (Exercises)

12: Comparing Several Means (One-way ANOVA)

- 12.1: Summary
- 12.2: An Illustrative Data Set
- 12.3: How ANOVA Works
- 12.4: Running an ANOVA in R
- 12.5: Effect Size
- 12.6: Multiple Comparisons and Post Hoc Tests
- 12.7: Assumptions of One-way ANOVA
- 12.8: Checking the Homogeneity of Variance Assumption
- 12.9: Removing the Homogeneity of Variance Assumption
- 12.10: Checking the Normality Assumption
- 12.11: Removing the Normality Assumption
- 12.12: On the Relationship Between ANOVA and the Student t Test

13: Introduction to Linear Regression

- 13.1: Prelude to Linear Regression
- 13.2: Line Fitting, Residuals, and Correlation
- 13.3: Fitting a Line by Least Squares Regression
- 13.4: Types of Outliers in Linear Regression
- 13.5: Inference for Linear Regression
- 13.6: Exercises

14: Multiple and Logistic Regression

- 14.1: Introduction to Multiple Regression
- 14.2: Model Selection
- 14.3: Checking Model Assumptions using Graphs
- 14.4: Introduction to Logistic Regression
- 14.5: Exercises
- 14.6: Statistical Literacy
- 14.E: Regression (Exercises)

15: Regression in R

- 15.1: What Is a Linear Regression Model?
- 15.2: Estimating a Linear Regression Model
- 15.3: Multiple Linear Regression
- 15.4: Quantifying the Fit of the Regression Model
- 15.5: Hypothesis Tests for Regression Models
- 15.6: Correlations
- 15.7: Handling Missing Values
- 15.8: Testing the Significance of a Correlation
- 15.9: Regarding Regression Coefficients
- 15.10: Assumptions of Regression
- 15.11: Model Checking
- 15.12: Model Selection
- 15.13: Summary

16: Research Design

- 16.1: Scientific Method
- 16.2: Measurement
- 16.3: Data Collection
- 16.4: Sampling Bias
- 16.5: Experimental Designs
- 16.6: Causation
- 16.7: Statistical Literacy
- 16.E: Research Design (Exercises)

17: Preparing Datasets and Other Pragmatic Matters

- 17.1: Tabulating and Cross-tabulating Data
- 17.2: Transforming and Recoding a Variable
- 17.3: A few More Mathematical Functions and Operations
- 17.4: Extracting a Subset of a Vector
- 17.5: Extracting a Subset of a Data Frame
- 17.6: Sorting, Flipping and Merging Data
- 17.7: Reshaping a Data Frame
- 17.8: Working with Text
- 17.9: Reading Unusual Data Files
- 17.10: Coercing Data from One Class to Another
- 17.11: Other Useful Data Structures
- 17.12: Miscellaneous Topics
- 17.13: Summary

18: Basic Programming

- 18.1: Scripts
- 18.2: Loops
- 18.3: Conditional Statements
- 18.4: Writing Functions
- 18.5: Implicit Loops
- 18.6: Summary

19: Bayesian Statistics

- 19.1: Probabilistic Reasoning by Rational Agents
- 19.2: Bayesian Hypothesis Tests
- 19.3: Why Be a Bayesian?
- 19.4: Evidentiary Standards You Can Believe
- 19.5: The p-value Is a Lie.
- 19.6: Bayesian Analysis of Contingency Tables
- 19.7: Bayesian t-tests
- 19.8: Bayesian Regression
- 19.9: Bayesian ANOVA
- 19.10: Summary

20: Case Studies and Data

- 20.1: Angry Moods
- 20.2: Flatulence
- 20.3: Physicians Reactions
- 20.4: Teacher Ratings
- 20.5: Diet and Health
- 20.6: Smiles and Leniency
- 20.7: Animal Research
- 20.8: ADHD Treatment
- 20.9: Weapons and Aggression
- 20.10: SAT and College GPA
- 20.11: Stereograms
- 20.12: Driving
- 20.13: Stroop Interference
- 20.14: TV Violence
- 20.15: Obesity and Bias
- 20.16: Shaking and Stirring Martinis
- 20.17: Adolescent Lifestyle Choices
- 20.18: Chocolate and Body Weight
- 20.19: Bedroom TV and Hispanic Children
- 20.20: Weight and Sleep Apnea
- 20.21: Misusing SEM
- 20.22: School Gardens and Vegetable Consumption
- 20.23: TV and Hypertension
- 20.24: Dietary Supplements
- 20.25: Young People and Binge Drinking
- 20.26: Sugar Consumption in the US Diet
- 20.27: Nutrition Information Sources and Older Adults
- 20.28: Mind Set - Exercise and the Placebo Effect
- 20.29: Predicting Present and Future Affect
- 20.30: Exercise and Memory
- 20.31: Parental Recognition of Child Obesity
- 20.32: Educational Attainment and Racial, Ethnic, and Gender Disparity

21: Math Review for Introductory Statistics

- 00: Front Matter
  - TitlePage
  - InfoPage
  - Table of Contents
  - Licensing
- 21.1: Decimals Fractions and Percents
  - 21.1.1: Comparing Fractions, Decimals, and Percents
  - 21.1.2: Converting Between Fractions, Decimals and Percents
  - 21.1.3: Decimals- Rounding and Scientific Notation
  - 21.1.4: Using Fractions, Decimals and Percents to Describe Charts
- 21.2: The Number Line
  - 21.2.1: Distance between Two Points on a Number Line
  - 21.2.2: Plotting Points and Intervals on the Number Line
  - 21.2.3: Represent an Inequality as an Interval on a Number Line
  - 21.2.4: The Midpoint
- 21.3: Operations on Numbers
  - 21.3.1: Area of a Rectangle
  - 21.3.2: Factorials and Combination Notation
  - 21.3.3: Order of Operations
  - 21.3.4: Order of Operations in Expressions and Formulas
  - 21.3.5: Perform Signed Number Arithmetic
  - 21.3.6: Powers and Roots
  - 21.3.7: Using Summation Notation
- 21.4: Sets
  - 21.4.1: Set Notation
  - 21.4.2: The Complement of a Set
  - 21.4.3: The Union and Intersection of Two Sets
  - 21.4.4: Venn Diagrams
- 21.5: Expressions, Equations and Inequalities
  - 21.5.1: Evaluate Algebraic Expressions
  - 21.5.2: Inequalities and Midpoints
  - 21.5.3: Solve Equations with Roots
  - 21.5.4: Solving Linear Equations in One Variable
- 21.6: Graphing Points and Lines in Two Dimensions
  - 21.6.1: Finding Residuals
  - 21.6.2: Find the Equation of a Line given its Graph
  - 21.6.3: Find y given x and the Equation of a Line
  - 21.6.4: Graph a Line given its Equation
  - 21.6.5: Interpreting the Slope of a Line
  - 21.6.6: Interpreting the y-intercept of a Line
  - 21.6.7: Plot an Ordered Pair
- Index
- Glossary
- Detailed Licensing

Index

Glossary

Detailed Licensing

Detailed Licensing

Licensing

*A detailed breakdown of this resource's licensing can be found in **Back Matter/Detailed Licensing**.*

CHAPTER OVERVIEW

1: Basics

1.1: Introduction

1.1.1: What Is Statistical Thinking?

1.1.2: Dealing with Statistics Anxiety

1.1.3: What Can Statistics Do for Us?

1.1.4: The Big Ideas of Statistics

1.1.5: Causality and Statistics

1.2: Working with Data

1.2.1: What Are Data?

1.2.2: Data Basics

1.2.3: Scales of Measurement

1.2.4: What Makes a Good Measurement?

1.2.5: Overview of Data Collection Principles

1.2.6: Observational Studies and Sampling Strategies

1.2.7: Experiments

1.2.8: How Not to Do Statistics

1.2.9: Exercises

1: Basics is shared under a not declared license and was authored, remixed, and/or curated by LibreTexts.

SECTION OVERVIEW

1.1: Introduction

Learning Objectives

Having read this chapter, you should be able to:

- Describe the central goals and fundamental concepts of statistics
- Describe the difference between experimental and observational research with regard to what can be inferred about causality
- Explain how randomization provides the ability to make inferences about causation.

“Statistical thinking will one day be as necessary for efficient citizenship as the ability to read and write.” - H.G. Wells

1.1.1: What Is Statistical Thinking?

1.1.2: Dealing with Statistics Anxiety

1.1.3: What Can Statistics Do for Us?

1.1.4: The Big Ideas of Statistics

1.1.5: Causality and Statistics

This page titled 1.1: Introduction is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

1.1.1: What Is Statistical Thinking?

Statistical thinking is a way of understanding a complex world by describing it in relatively simple terms that nonetheless capture essential aspects of its structure, and that also provide us some idea of how uncertain we are about our knowledge. The foundations of statistical thinking come primarily from mathematics and statistics, but also from computer science, psychology, and other fields of study.

We can distinguish statistical thinking from other forms of thinking that are less likely to describe the world accurately. In particular, human intuition often tries to answer the same questions that we can answer using statistical thinking, but often gets the answer wrong. For example, in recent years most Americans have reported that they think that violent crime was worse compared to the previous year (Pew Research Center). However, a statistical analysis of the actual crime data shows that in fact violent crime has steadily *decreased* since the 1990's. Intuition fails us because we rely upon best guesses (which psychologists refer to as *heuristics*) that can often get it wrong. For example, humans often judge the prevalence of some event (like violent crime) using an *availability heuristic* – that is, how easily can we think of an example of violent crime. For this reason, our judgments of increasing crime rates may be more reflective of increasing news coverage, in spite of an actual decrease in the rate of crime. Statistical thinking provides us with the tools to more accurately understand the world and overcome the fallibility of human intuition.

This page titled 1.1.1: What Is Statistical Thinking? is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

- **1.1: What Is Statistical Thinking?** by Russell A. Poldrack is licensed CC BY-NC 4.0. Original source: https://statsthinking21.github.io/statsthinking21-core-site.

1.1.2: Dealing with Statistics Anxiety

Many people come to their first statistics class with a lot of trepidation and anxiety, especially once they hear that they will also have to learn to code in order to analyze data. In my class I give students a survey prior to the first session in order to measure their attitude towards statistics, asking them to rate a number of statments on a scale of 1 (strongly disagree) to 7 (strongly agree). One of the items on the survey is “The thought of being enrolled in a statistics course makes me nervous”. In the most recent class, almost two-thirds of the class responded with a five or higher, and about one-fourth of the students said that they strongly agreed with the statement. So if you feel nervous about starting to learn statistics, you are not alone.

Anxiety feels uncomfortable, but psychology tells us that this kind of emotional arousal can actually help us perform *better* on many tasks, by focusing our attention So if you start to feel anxious about the material in this course, remind yourself that many others in the class are feeling similarly, and that the arousal could actually help you perform better (even if it doesn’t seem like it!).

This page titled 1.1.2: Dealing with Statistics Anxiety is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

- **1.2: Dealing with Statistics Anxiety** by Russell A. Poldrack is licensed CC BY-NC 4.0. Original source: https://statsthinking21.github.io/statsthinking21-core-site.

1.1.3: What Can Statistics Do for Us?

There are three major things that we can do with statistics:

- *Describe*: The world is complex and we often need to describe it in a simplified way that we can understand.
- *Decide*: We often need to make decisions based on data, usually in the face of uncertainty.
- *Predict*: We often wish to make predictions about new situations based on our knowledge of previous situations.

Let's look at an example of these in action, centered on a question that many of us are interested in: How do we decide what's healthy to eat?
There are many different sources of guidance, from government dietary guidelines to diet books to bloggers.
Let's focus in on a specific question: Is saturated fat in our diet a bad thing?

One way that we might answer this question is common sense.
If we eat fat then it's going to turn straight into fat in our bodies, right?
And we have all seen photos of arteries clogged with fat, so eating fat is going to clog our arteries, right?

Another way that we might answer this question is by listening to authority figures. The Dietary Guidelines from the US Food and Drug Administration have as one of their Key Recommendations that "A healthy eating pattern limits saturated fats".You might hope that these guidelines would be based on good science, and in some cases they are, but as Nina Teicholz outlined in her book "Big Fat Surprise"(Teicholz 2014), this particular recommendation seems to be based more on the dogma of nutrition researchers than on actual evidence.

Finally, we might look at actual scientific research. Let's start by looking at a large study called the PURE study, which has examined diets and health outcomes (including death) in more than 135,000 people from 18 different countries. In one of the analyses of this dataset (published in *The Lancet* in 2017; Dehghan et al. (2017)), the PURE investigators reported an analysis of how intake of various classes of macronutrients (including saturated fats and carbohydrates) was related to the likelihood of dying during the time that people were followed. People were followed for a *median* of 7.4 years, meaning that half of the people in the study were followed for less and half were followed for more than 7.4 years. Figure 1.1 plots some of the data from the study (extracted from the paper), showing the relationship between the intake of both saturated fats and carbohydrates and the risk of dying from any cause.

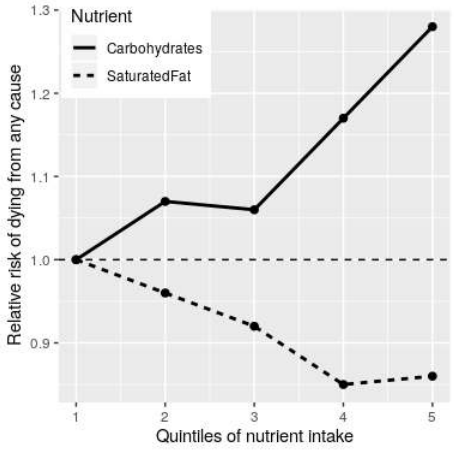


Figure 1.1: A plot of data from the PURE study, showing the relationship between death from any cause and the relative intake of saturated fats and carbohydrates.

This plot is based on ten numbers. To obtain these numbers, the researchers split the group of 135,335 study participants (which we call the "sample") into 5 groups ("quintiles") after ordering them in terms of their intake of either of the nutrients; the first quintile contains the 20% of people with the lowest intake, and the 5th quintile contains the 20% with the highest intake. The researchers then computed how often people in each of those groups died during the time they were being followed. The figure expresses this in terms of the *relative risk* of dying in comparison to the lowest quintile: If this number is greater than 1 it means that people in the group are *more* likely to die than are people in the lowest quintile, whereas if it's less than one it means that people in the group are

less likely to die. The figure is pretty clear: People who ate more saturated fat were *less* likely to die during the study, with the lowest death rate seen for people who were in the fourth quintile (that is, who ate more fat than the lowest 60% but less than the top 20%). The opposite is seen for carbohydrates; the more carbs a person ate, the more likely they were to die during the study. This example shows how we can use statistics to *describe* a complex dataset in terms of a much simpler set of numbers; if we had to look at the data from each of the study participants at the same time, we would be overloaded with data and it would be hard to see the pattern that emerges when they are described more simply.

The numbers in Figure 1.1 seem to show that deaths decrease with saturated fat and increase with carbohydrate intake, but we also know that there is a lot of uncertainty in the data; there are some people who died early even though they ate a low-carb diet, and, similarly, some people who ate a ton of carbs but lived to a ripe old age. Given this variability, we want to *decide* whether the relationships that we see in the data are large enough that we wouldn't expect them to occur randomly if there was not truly a relationship between diet and longevity. Statistics provide us with the tools to make these kinds of decisions, and often people from the outside view this as *the* main purpose of statistics. But as we will see throughout the book, this need for black-and-white decisions based on fuzzy evidence has often led researchers astray.

Based on the data we would also like to make predictions about future outcomes. For example, a life insurance company might want to use data about a particular person's intake of fat and carbohydrate to predict how long they are likely to live. An important aspect of prediction is that it requires us to generalize from the data we already have to some other situation, often in the future; if our conclusions were limited to the specific people in the study at a particular time, then the study would not be very useful. In general, researchers must assume that their particular sample is representative of a larger *population*, which requires that they obtain the sample in a way that provides an unbiased picture of the population. For example, if the PURE study had recruited all of its participants from religious sects that practice vegetarianism, then we probably wouldn't want to generalize the results to people who follow different dietary standards.

This page titled 1.1.3: What Can Statistics Do for Us? is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

- **1.3: What Can Statistics Do for Us?** by Russell A. Poldrack is licensed CC BY-NC 4.0. Original source: https://statsthinking21.github.io/statsthinking21-core-site.

1.1.4: The Big Ideas of Statistics

There are a number of very basic ideas that cut through nearly all aspects of statistical thinking. Several of these are outlined by Stigler (2016) in his outstanding book "The Seven Pillars of Statistical Wisdom", which I have augmented here.

1.4.1 Learning from data

One way to think of statistics is as a set of tools that enable us to learn from data. In any situation, we start with a set of ideas or *hypotheses* about what might be the case. In the PURE study, the researchers may have started out with the expectation that eating more fat would lead to higher death rates, given the prevailing negative dogma about saturated fats. Later in the course we will introduce the idea of *prior knowledge*, which is meant to reflect the knowledge that we bring to a situation. This prior knowledge can vary in its strength, often based on our amount of experience; if I visit a restaurant for the first time I am likely to have a weak expectation of how good it will be, but if I visit a restaurant where I have eaten ten times before, my expectations will be much stronger. Similarly, if I look at a restaurant review site and see that a restaurant's average rating of four stars is only based on three reviews, I will have a weaker expectation than I would if it was based on 300 reviews.

Statistics provides us with a way to describe how new data can be best used to update our beliefs, and in this way there are deep links between statistics and psychology. In fact, many theories of human and animal learning from psychology are closely aligned with ideas from the new field of *machine learning*. Machine learning is a field at the interface of statistics and computer science that focuses on how to build computer algorithms that can learn from experience. While statistics and machine learning often try to solve the same problems, researchers from these fields often take very different approaches; the famous statistician Leo Breiman once referred to them as "The Two Cultures" to reflect how different their approaches can be (Breiman 2001). In this book I will try to blend the two cultures together because both approaches provide useful tools for thinking about data.

1.4.2 Aggregation

Another way to think of statistics is "the science of throwing away data". In the example of the PURE study above, we took more than 100,000 numbers and condensed them into ten. It is this kind of *aggregation* that is one of the most important concepts in statistics. When it was first advanced, this was revolutionary: If we throw out all of the details about every one of the participants, then how can we be sure that we aren't missing something important?

As we will see, statistics provides us ways to characterize the structure of aggregates of data, and with theoretical foundations that explain why this usually works well. However, it's also important to keep in mind that aggregation can go too far, and later we will encounter cases where a summary can provide a misleading picture of the data being summarized.

1.4.3 Uncertainty

The world is an uncertain place. We now know that cigarette smoking causes lung cancer, but this causation is probabilistic: A 68-year-old man who smoked two packs a day for the past 50 years and continues to smoke has a 15% (1 out of 7) risk of getting lung cancer, which is much higher than the chance of lung cancer in a nonsmoker. However, it also means that there will be many people who smoke their entire lives and never get lung cancer. Statistics provides us with the tools to characterize uncertainty, to make decisions under uncertainty, and to make predictions whose uncertainty we can quantify.

One often sees journalists write that scientific researchers have "proven" some hypothesis. But statistical analysis can never "prove" a hypothesis, in the sense of demonstrating that it must be true (as one would in a logical or mathematical proof). Statistics can provide us with evidence, but it's always tentative and subject to the uncertainty that is always present in the real world.

1.4.4 Sampling

The concept of aggregation implies that we can make useful insights by collapsing across data – but how much data do we need? The idea of *sampling* says that we can summarize an entire population based on just a small number of samples from the population, as long as those samples are obtained in the right way. For example, the PURE study enrolled a sample of about 135,000 people, but its goal was to provide insights about the billions of humans who make up the population from which those people were sampled. As we already discussed above, the way that the study sample is obtained is critical, as it determines how broadly we can generalize the results. Another fundamental insight about sampling is that while larger samples are always better (in

terms of their ability to accurately represent the entire population), there are diminishing returns as the sample gets larger. In fact, the rate at which the benefit of larger samples decreases follows a simple mathematical rule, growing as the square root of the sample size, such that in order to double the quality of our data we need to quadruple the size of our sample.

This page titled 1.1.4: The Big Ideas of Statistics is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

- **1.4: The Big Ideas of Statistics** by Russell A. Poldrack is licensed CC BY-NC 4.0. Original source: https://statsthinking21.github.io/statsthinking21-core-site.

1.1.5: Causality and Statistics

The PURE study seemed to provide pretty strong evidence for a positive relationship between eating saturated fat and living longer, but this doesn't tell us what we really want to know: If we eat more saturated fat, will that cause us to live longer? This is because we don't know whether there is a direct causal relationship between eating saturated fat and living longer. The data are consistent with such a relationship, but they are equally consistent with some other factor causing both higher saturated fat and longer life. For example, it is likely that people who are richer eat more saturated fat and richer people tend to live longer, but their longer life is not necessarily due to fat intake — it could instead be due to better health care, reduced psychological stress, better food quality, or many other factors. The PURE study investigators tried to account for these factors, but we can't be certain that their efforts completely removed the effects of other variables. The fact that other factors may explain the relationship between saturated fat intake and death is an example of why introductory statistics classes often teach that "correlation does not imply causation", though the renowned data visualization expert Edward Tufte has added, "but it sure is a hint."

Although observational research (like the PURE study) cannot conclusively demonstrate causal relations, we generally think that causation can be demonstrated using studies that experimentally control and manipulate a specific factor. In medicine, such a study is referred to as a *randomized controlled trial* (RCT). Let's say that we wanted to do an RCT to examine whether increasing saturated fat intake increases life span. To do this, we would sample a group of people, and then assign them to either a treatment group (which would be told to increase their saturated fat intake) or a control group (who would be told to keep eating the same as before). It is essential that we assign the individuals to these groups randomly. Otherwise, people who choose the treatment might be different in some way than people who choose the control group – for example, they might be more likely to engage in other healthy behaviors as well. We would then follow the participants over time and see how many people in each group died. Because we randomized the participants to treatment or control groups, we can be reasonably confident that there are no other differences between the groups that would *confound* the treatment effect; however, we still can't be certain because sometimes randomization yields treatment versus control groups that *do* vary in some important way. Researchers often try to address these confounds using statistical analyses, but removing the influence of a confound from the data can be very difficult.

A number of RCTs have examined the question of whether changing saturated fat intake results in better health and longer life. These trials have focused on *reducing* saturated fat because of the strong dogma amongst nutrition researchers that saturated fat is deadly; most of these researchers would have probably argued that it was not ethical to cause people to eat *more* saturated fat! However, the RCTs have show a very consistent pattern: Overall there is no appreciable effect on death rates of reducing saturated fat intake.

This page titled 1.1.5: Causality and Statistics is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

- **1.5: Causality and Statistics** by Russell A. Poldrack is licensed CC BY-NC 4.0. Original source: https://statsthinking21.github.io/statsthinking21-core-site.

SECTION OVERVIEW

1.2: Working with Data

Learning Objectives

Having read this chapter, you should be able to:

- Distinguish between different types of variables (quantitative/qualitative, binary/integer/real, discrete/continuous) and give examples of each of these kinds of variables
- Distinguish between the concepts of reliability and validity and apply each concept to a particular dataset

1.2.1: What Are Data?

1.2.2: Data Basics

1.2.3: Scales of Measurement

1.2.4: What Makes a Good Measurement?

1.2.5: Overview of Data Collection Principles

1.2.6: Observational Studies and Sampling Strategies

1.2.7: Experiments

1.2.8: How Not to Do Statistics

1.2.9: Exercises

This page titled 1.2: Working with Data is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

1.2.1: What Are Data?

The first important point about data is that data *are* - meaning that the word "data" is plural (though some people disagree with me on this). You might also wonder how to pronounce "data" – I say "day-tah" but I know many people who say "dah-tah" and I have been able to remain friends with them in spite of this. Now if I heard them say "the data is" then that would be bigger issue…

2.1.1 Qualitative data

Data are composed of *variables*, where a variable reflects a unique measurement or quantity. Some variables are *qualitative*, meaning that they describe a quality rather than a numeric quantity. For example, in my stats course I generally give an introductory survey, both to obtain data to use in class and to learn more about the students. One of the questions that I ask is "What is your favorite food?", to which some of the answers have been: blueberries, chocolate, tamales, pasta, pizza, and mango. Those data are not intrinsically numerical; we could assign numbers to each one (1=blueberries, 2=chocolate, etc), but we would just be using the numbers as labels rather than as real numbers; for example, it wouldn't make sense to add the numbers together in this case. However, we will often code qualitative data using numbers in order to make them easier to work with, as you will see later.

2.1.2 Quantitative data

More commonly in statistics we will work with *quantitative* data, meaning data that are numerical. For example, here Table 2.1 shows the results from another question that I ask in my introductory class, which is "Why are you taking this class?"

Table 2.1: Counts of the prevalence of different responses to the question "Why are you taking this class?"

Why are you taking this class?	Number of students
It fulfills a degree plan requirement	105
It fulfills a General Education Breadth Requirement	32
It is not required but I am interested in the topic	11
Other	4

Note that the students' answers were qualitative, but we generated a quantitative summary of them by counting how many students gave each response.

2.1.2.1 Types of numbers

There are several different types of numbers that we work with in statistics. It's important to understand these differences, in part because programming languages like R often distinguish between them.

Binary numbers. The simplest are binary numbers – that is, zero or one. We will often use binary numbers to represent whether something is true or false, or present or absent. For example, I might ask 10 people if they have ever experienced a migraine headache, recording their answers as "Yes" or "No". It's often useful to instead use *logical* values, which take the value of either `TRUE` or `FALSE`. We can create these by testing whether each value is equal to "Yes", which we can do using the `==` symbol. This will return the value `TRUE` for any matching "Yes" values, and `FALSE` otherwise. These are useful to R knows how to interpret them natively, whereas it doesn't know what "Yes" and "No" mean.

In general, most programming languages treat truth values and binary numbers equivalently. The number 1 is equal to the logical value `TRUE`, and the number zero is equal to the logical value `FALSE`.

Integers. Integers are whole numbers with no fractional or decimal part. We most commonly encounter integers when we count things, but they also often occur in psychological measurement. For example, in my introductory survey I administer a set of questions about attitudes towards statistics (such as "Statistics seems very mysterious to me."), on which the students respond with a number between 1 ("Disagree strongly") and 7 ("Agree strongly").

Real numbers. Most commonly in statistics we work with real numbers, which have a fractional/decimal part. For example, we might measure someone's weight, which can be measured to an arbitrary level of precision, from whole pounds down to

micrograms.

This page titled 1.2.1: What Are Data? is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

- **2.1: What Are Data?** by Russell A. Poldrack is licensed CC BY-NC 4.0. Original source: https://statsthinking21.github.io/statsthinking21-core-site.

1.2.2: Data Basics

Effective presentation and description of data is a first step in most analyses. This section introduces one structure for organizing data as well as some terminology that will be used throughout this book.

Observations, variables, and data matrices

Table 1.3 displays rows 1, 2, 3, and 50 of a data set concerning 50 emails received during early 2012. These observations will be referred to as the email50 data set, and they are a random sample from a larger data set that we will see in Section 1.7.

Table 1.3: Four rows from the email 50 data matrix.

	spam	num_char	line_breaks	format	number
1	no	21,705	551	html	small
2	no	7,011	183	html	big
3	yes	631	28	text	none
⋮	⋮	⋮	⋮	⋮	⋮
50	no	15,829	242	html	small

Each row in the table represents a single email or **case** (a *case is also sometimes called a unit of observation or an observational unit.).* The columns represent characteristics, called **variables**, for each of the emails. For example, the first row represents email 1, which is a not spam, contains 21,705 characters, 551 line breaks, is written in HTML format, and contains only small numbers.

In practice, it is especially important to ask clarifying questions to ensure important aspects of the data are understood. For instance, it is always important to be sure we know what each variable means and the units of measurement. Descriptions of all five email variables are given in Table 1.4.

Table 1.4: Variables and their descriptions for the email 50 data set.

variable	description
spam	Specifies whether the message was spam
num_char	The number of characters in the email
line_breaks	The number of line breaks in the email (not including text wrapping)
format	Indicates if the email contained special formatting, such as bolding, tables, or links, which would indicate the message is in HTML format
number	Indicates whether the email contained no number, a small number (under1 million), or a large number

The data in Table 1.3 represent a **data matrix**, which is a common way to organize data. Each row of a data matrix corresponds to a unique case, and each column corresponds to a variable. A data matrix for the stroke study introduced in Section 1.1 is shown in Table 1.1, where the cases were patients and there were three variables recorded for each patient.

Data matrices are a convenient way to record and store data. If another individual or case is added to the data set, an additional row can be easily added. Similarly, another column can be added for a new variable.

Exercise 1.2.2.1

Exercise 1.2 We consider a publicly available data set that summarizes information about the 3,143 counties in the United states, and we call this the county data set. This data set includes information about each county: its name, the state where it resides, its population in 2000 and 2010, per capita federal spending, poverty rate, and ve additional characteristics. How might these data be organized in a data matrix? Reminder: look in the footnotes for answers to in-text exercises.[5]

[5]*Each county may be viewed as a case, and there are eleven pieces of information recorded for each case. A table with 3,143 rows and 11 columns could hold these data, where each row represents a county and each column represents a particular piece of information.*

Seven rows of the county data set are shown in Table 1.5, and the variables are summarized in Table 1.6. These data were collected from the US Census website.[6]

[6]*quickfacts.census.gov/qfd/index.html*

Table 1.5: Seven rows from the county data set.

	name	state	pop 2000	pop 2010	fed spend	poverty	home owner-ship	multiu-nit	income	med income	smoking ban
1	Autau-ga	AL	43671	54571	6.068	10.6	77.5	7.2	24568	53255	none
2	Baldw-in	AL	140415	182265	6.140	12.2	76.7	22.6	26469	50147	none
3	Barbo-ur	AL	29038	27457	8.752	25.0	68.0	11.1	15875	33219	none
4	Bibb	AL	20826	22915	7.122	12.6	82.9	6.6	19918	41770	none
5	Blount	AL	51024	57322	5.131	13.4	82.0	3.7	21070	45549	none
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
3142	Wash-akie	WY	8289	8533	8.714	5.6	70.9	10.0	28557	48379	none
3143	West-on	WY	6644	7208	6.695	7.9	77.9	6.5	28463	53853	none

Table 1.6: Variables and their descriptions for the county data set.

variable description
name County name state State where the county resides (also including the District of Columbia) pop2000 Population in 2000 pop2010 Population in 2010 fed_spend Federal spending per capita poverty Percent of the population in poverty homeownership Percent of the population that lives in their own home or lives with the owner (e.g. children living with parents who own the home) multiunit Percent of living units that are in multi-unit structures (e.g. apartments) income Income per capita med_income Median household income for the county, where a household's income equals the total income of its occupants who are 15 years or older smoking_ban Type of county-wide smoking ban in place at the end of 2011, which takes one of three values: none, partial, or comprehensive, where a comprehensive ban means smoking was not permitted in restaurants, bars, or workplaces, and partial means smoking was banned in at least one of those three locations

Types of variables

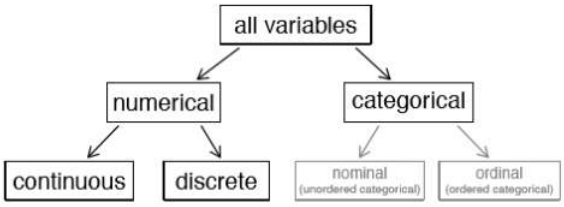


Figure 1.7: Breakdown of variables into their respective types.

Examine the fed spend, pop2010, state, and smoking ban variables in the county data set. Each of these variables is inherently different from the other three yet many of them share certain characteristics.

First consider fed spend, which is said to be a **numerical** variable since it can takea wide range of numerical values, and it is sensible to add, subtract, or take averages with those values. On the other hand, we would not classify a variable reporting telephone area codes as numerical since their average, sum, and difference have no clear meaning.

The pop2010 variable is also numerical, although it seems to be a little different than fed spend. This variable of the population count can only take whole non-negative numbers (0, 1, 2, ...). For this reason, the population variable is said to be discrete since it can only take numerical values with jumps. On the other hand, the federal spending variable is said to be **continuous**.

The variable state can take up to 51 values after accounting for Washington, DC: AL, ..., and WY. Because the responses themselves are categories, state is called a **categorical** variable,7 and the possible values are called the variable's **levels**.

Finally, consider the smoking ban variable, which describes the type of county-wide smoking ban and takes values none, partial, or comprehensive in each county. This variable seems to be a hybrid: it is a categorical variable but the levels have a natural ordering. A variable with these properties is called an **ordinal** variable. To simplify analyses, any ordinal variables in this book will be treated as categorical variables.

Example 1.3 Data were collected about students in a statistics course. Three variables were recorded for each student: number of siblings, student height, and whether the student had previously taken a statistics course. Classify each of the variables as continuous numerical, discrete numerical, or categorical.

The number of siblings and student height represent numerical variables. Because the number of siblings is a count, it is discrete. Height varies continuously, so it is a continuous numerical variable. The last variable classi es students into two categories - those who have and those who have not taken a statistics course - which makes this variable categorical.

Exercise 1.2.2.1

Exercise 1.4 Consider the variables group and outcome (at 30 days) from the stent study in Section 1.1. Are these numerical or categorical variables?[8]

[8]*There are only two possible values for each variable, and in both cases they describe categories. Thus, each are categorical variables.*

[7]*Sometimes also called a nominal variable.*

Relationships between variables

Many analyses are motivated by a researcher looking for a relationship between two or more variables. A social scientist may like to answer some of the following questions:

1. Is federal spending, on average, higher or lower in counties with high rates of poverty?
2. If homeownership is lower than the national average in one county, will the percent of multi-unit structures in that county likely be above or below the national average?
3. Which counties have a higher average income: those that enact one or more smoking bans or those that do not?

To answer these questions, data must be collected, such as the county data set shown in Table 1.5. Examining summary statistics could provide insights for each of the three questions about counties. Additionally, graphs can be used to visually summarize data and are useful for answering such questions as well.

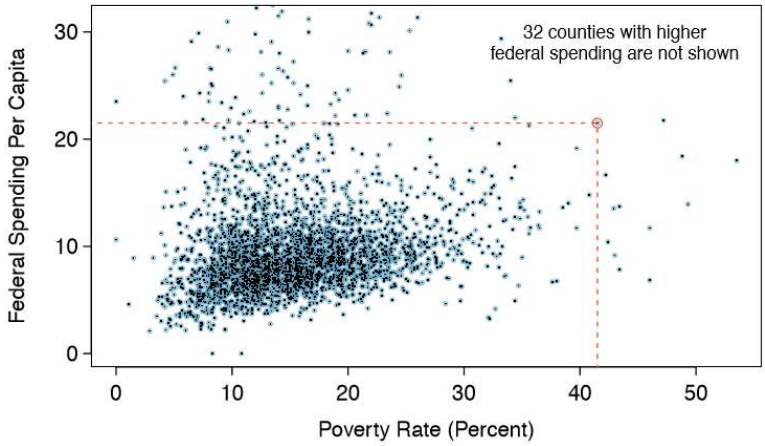


Figure 1.8: A scatterplot showing fed spend against poverty. Owsley County of Kentucky, with a poverty rate of 41.5% and federal spending of $21.50 per capita, is highlighted.

Scatterplots are one type of graph used to study the relationship between two numerical variables. Figure 1.8 compares the variables fed spend and poverty. Each point on the plot represents a single county. For instance, the highlighted dot corresponds to County 1088 in the county data set: Owsley County, Kentucky, which had a poverty rate of 41.5% and federal spending of $21.50 per capita. The scatterplot suggests a relationship between the two variables: counties with a high poverty rate also tend to have slightly more federal spending. We might brainstorm as to why this relationship exists and investigate each idea to determine which is the most reasonable explanation.

Exercise 1.2.2.1

Exercise 1.5 Examine the variables in the email50 data set, which are described in Table 1.4 on page 4. Create two questions about the relationships between these variables that are of interest to you.[9]

[9]*Two sample questions: (1) Intuition suggests that if there are many line breaks in an email then there would tend to also be many characters: does this hold true? (2) Is there a connection between whether an email format is plain text (versus HTML) and whether it is a spam message?*

The fed_spend and poverty variables are said to be associated because the plot shows a discernible pattern. When two variables show some connection with one another, they are called associated variables. Associated variables can also be called dependent variables and vice-versa.

Example 1.2.2.1

Example 1.6 This example examines the relationship between homeownership and the percent of units in multi-unit structures (e.g. apartments, condos), which is visualized using a scatterplot in Figure 1.9. Are these variables associated?

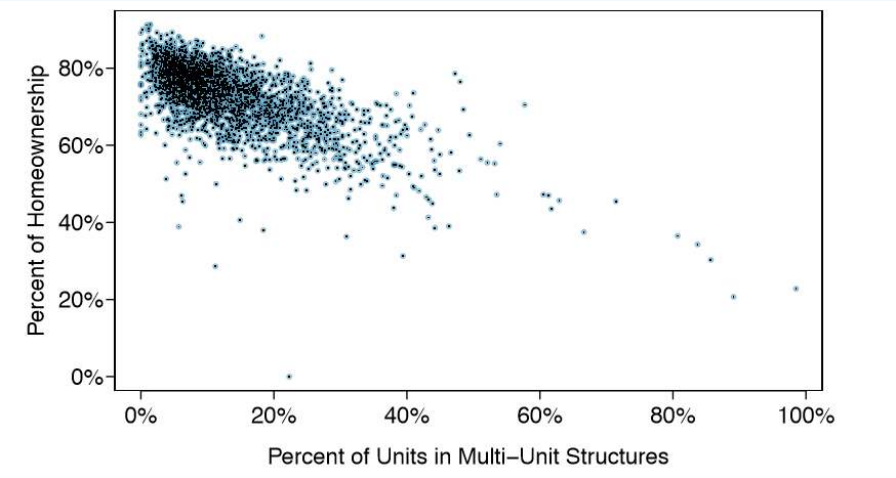


Figure 1.9: A scatterplot of homeownership versus the percent of units that are in multi-unit structures for all 3,143 counties.

Solution

It appears that the larger the fraction of units in multi-unit structures, the lower the homeownership rate. Since there is some relationship between the variables, they are associated.

Because there is a downward trend in Figure 1.9 { counties with more units in multiunit structures are associated with lower homeownership - these variables are said to be negatively associated. A positive association is shown in the relationship between the poverty and fed spend variables represented in Figure 1.8, where counties with higher poverty rates tend to receive more federal spending per capita.

If two variables are not associated, then they are said to be independent. That is, two variables are **independent** if there is no evident relationship between the two.

Associated or independent, never both

A pair of variables are either related in some way (associated) or not (independent). No pair of variables is both associated and independent.

This page titled 1.2.2: Data Basics is shared under a CC BY-SA 3.0 license and was authored, remixed, and/or curated by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel via source content that was edited to the style and standards of the LibreTexts platform.

- **1.3: Data Basics** by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel is licensed CC BY-SA 3.0. Original source: https://www.openintro.org/book/os.

1.2.3: Scales of Measurement

2.4.1 Scales of measurement

All variables must take on at least two different possible values (otherwise they would be a *constant* rather than a variable), but different values of the variable can relate to each other in different ways, which we refer to as *scales of measurement*. There are four ways in which the different values of a variable can differ.

- *Identity*: Each value of the variable has a unique meaning.
- *Magnitude*: The values of the variable reflect different magnitudes and have an ordered relationship to one another — that is, some values are larger and some are smaller.
- *Equal intervals*: Units along the scale of measurement are equal to one another. This means, for example, that the difference between 1 and 2 would be equal in its magnitude to the difference between 19 and 20.
- *Absolute zero*: The scale has a true meaningful zero point. For example, for many measurements of physical quantities such as height or weight, this is the complete absence of the thing being measured.

There are four different scales of measurement that go along with these different ways that values of a variable can differ.

Nominal scale. A nominal variable satisfies the criterion of identity, such that each value of the variable represents something different, but the numbers simply serve as qualitative labels as discussed above. For example, we might ask people for their political party affiliation, and then code those as numbers: 1 = “Republican”, 2 = “Democrat”, 3 = “Libertarian”, and so on. However, the different numbers do not have any ordered relationship with one another.

Ordinal scale. An ordinal variable satisfies the criteria of identity and magnitude, such that the values can be ordered in terms of their magnitude. For example, we might ask a person with chronic pain to complete a form every day assessing how bad their pain is, using a 1-7 numeric scale. Note that while the person is presumably feeling more pain on a day when they report a 6 versus a day when they report a 3, it wouldn’t make sense to say that their pain is twice as bad on the former versus the latter day; the ordering gives us information about relative magnitude, but the differences between values are not necessarily equal in magnitude.

Interval scale. An interval scale has all of the features of an ordinal scale, but in addition the intervals between units on the measurement scale can be treated as equal. A standard example is physical temperature measured in Celsius or Farenheit; the physical difference between 10 and 20 degrees is the same as the physical difference between 90 and 100 degrees, but each scale can also take on negative values.

Ratio scale. A ratio scale variable has all four of the features outlined above: identity, magnitude, equal intervals, and absolute zero. The difference between a ratio scale variable and an interval scale variable is that the ratio scale variable has a true zero point. Examples of ratio scale variables include physical height and weight, along with temperature measured in Kelvin.

There are two important reasons that we must pay attention to the scale of measurement of a variable. First, the scale determines what kind of mathematical operations we can apply to the data (see Table 2.2). A nominal variable can only be compared for equality; that is, do two observations on that variable have the same numeric value? It would not make sense to apply other mathematical operations to a nominal variable, since they don’t really function as numbers in a nominal variable, but rather as labels. With ordinal variables, we can also test whether one value is greater or lesser than another, but we can’t do any arithmetic. Interval and ratio variables allow us to perform arithmetic; with interval variables we can only add or subtract values, whereas with ratio variables we can also multiply and divide values.

Table 2.2: Different scales of measurement admit different types of numeric operations

	Equal/not equal	>/<	+/-	Multiply/divide
Nominal	OK			
Ordinal	OK	OK		
Interval	OK	OK	OK	
Ratio	OK	OK	OK	OK

These constraints also imply that there are certain kinds of statistics that we can compute on each type of variable. Statistics that simply involve counting of different values (such as the most common value, known as the *mode*), can be calculated on any of the variable types. Other statistics are based on ordering or ranking of values (such as the *median*, which is the middle value when all of the values are ordered by their magnitude), and these require that the value at least be on an ordinal scale. Finally, statistics that involve adding up values (such as the average, or *mean*), require that the variables be at least on an interval scale. Having said that, we should note that it's quite common for researchers to compute the mean of variables that are only ordinal (such as responses on personality tests), but this can sometimes be problematic.

This page titled 1.2.3: Scales of Measurement is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

- **2.4: Appendix** by Russell A. Poldrack is licensed CC BY-NC 4.0. Original source: https://statsthinking21.github.io/statsthinking21-core-site.

1.2.4: What Makes a Good Measurement?

In many fields such as psychology, the thing that we are measuring is not a physical feature, but instead is an unobservable theoretical concept, which we usually refer to as a *construct*. For example, let's say that I want to test how well you understand the distinction between the four different scales of measurement described above. I could give you a pop quiz that would ask you several questions about these concepts and count how many you got right. This test might or might not be a good measurement of the construct of your actual knowledge — for example, if I were to write the test in a confusing way or use language that you don't understand, then the test might suggest you don't understand the concepts when really you do. On the other hand, if I give a multiple choice test with very obvious wrong answers, then you might be able to perform well on the test even if you don't actually understand the material.

It is usually impossible to measure a construct without some amount of error. In the example above, you might know the answer but you might mis-read the question and get it wrong. In other cases there is error intrinsic to the thing being measured, such as when we measure how long it takes a person to respond on a simple reaction time test, which will vary from trial to trial for many reasons. We generally want our measurement error to be as low as possible.

Sometimes there is a standard against which other measurements can be tested, which we might refer to as a "gold standard" — for example, measurement of sleep can be done using many different devices (such as devices that measure movement in bed), but they are generally considered inferior to the gold standard of polysomnography (which uses measurement of brain waves to quantify the amount of time a person spends in each stage of sleep). Often the gold standard is more difficult or expensive to perform, and the cheaper method is used even though it might have greater error.

When we think about what makes a good measurement, we usually distinguish two different aspects of a good measurement.

2.5.1 Reliability

Reliability refers to the consistency of our measurements. One common form of reliability, known as "test-retest reliability", measures how well the measurements agree if the same measurement is performed twice. For example, I might give you a questionnaire about your attitude towards statistics today, repeat this same questionnaire tomorrow, and compare your answers on the two days; we would hope that they would be very similar to one another, unless something happened in between the two tests that should have changed your view of statistics (like reading this book!).

Another way to assess reliability comes in cases where the data includes subjective judgments. For example, let's say that a researcher wants to determine whether a treatment changes how well an autistic child interacts with other children, which is measured by having experts watch the child and rate their interactions with the other children. In this case we would like to make sure that the answers don't depend on the individual rater — that is, we would like for there to be high *inter-rater reliability*. This can be assessed by having more than one rater perform the rating, and then comparing their ratings to make sure that they agree well with one another.

Reliability is important if we want to compare one measurement to another. The relationship between two different variables can't be any stronger than the relationship between either of the variables and itself (i.e., its reliability). This means that an unreliable measure can never have a strong statistical relationship with any other measure. For this reason, researchers developing a new measurement (such as a new survey) will often go to great lengths to establish and improve its reliability.

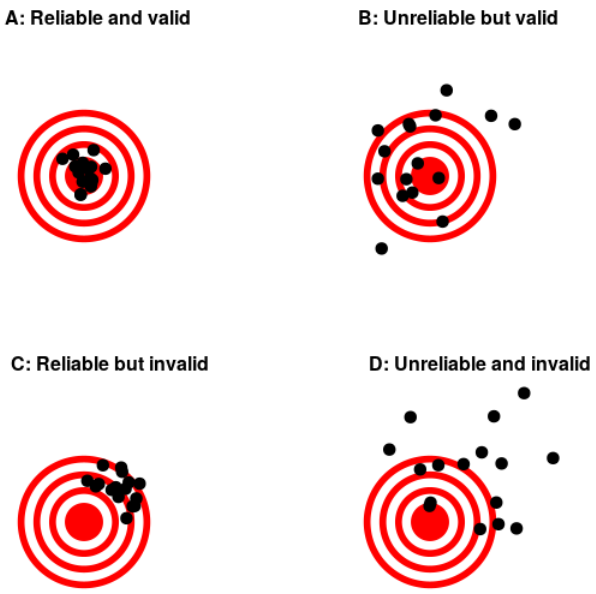


Figure 2.1: A figure demonstrating the distinction between reliability and validity, using shots at a bullseye. Reliability refers to the consistency of location of shots, and validity refers to the accuracy of the shots with respect to the center of the bullseye.

2.5.2 Validity

Reliability is important, but on its own it's not enough: After all, I could create a perfectly reliable measurement on a personality test by re-coding every answer using the same number, regardless of how the person actually answers. We want our measurements to also be *valid* — that is, we want to make sure that we are actually measuring the construct that we think we are measuring (Figure 2.1). There are many different types of validity that are commonly discussed; we will focus on three of them.

Face validity. Does the measurement make sense on its face? If I were to tell you that I was going to measure a person's blood pressure by looking at the color of their tongue, you would probably think that this was not a valid measure on its face. On the other hand, using a blood pressure cuff would have face validity. This is usually a first reality check before we dive into more complicated aspects of validity.

Construct validity. Is the measurement related to other measurements in an appropriate way? This is often subdivided into two aspects. *Convergent validity* means that the measurement should be closely related to other measures that are thought to reflect the same construct. Let's say that I am interested in measuring how extroverted a person is using a questionnaire or an interview. Convergent validity would be demonstrated if both of these different measurements are closely related to one another. On the other hand, measurements thought to reflect different constructs should be unrelated, known as *divergent validity*. If my theory of personality says that extraversion and conscientiousness are two distinct constructs, then I should also see that my measurements of extraversion are *unrelated* to measurements of conscientiousness.

Predictive validity. If our measurements are truly valid, then they should also be predictive of other outcomes. For example, let's say that we think that the psychological trait of sensation seeking (the desire for new experiences) is related to risk taking in the real world. To test for predictive validity of a measurement of sensation seeking, we would test how well scores on the test predict scores on a different survey that measures real-world risk taking.

This page titled 1.2.4: What Makes a Good Measurement? is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

- **2.5: What Makes a Good Measurement?** by Russell A. Poldrack is licensed CC BY-NC 4.0. Original source: https://statsthinking21.github.io/statsthinking21-core-site.

1.2.5: Overview of Data Collection Principles

The first step in conducting research is to identify topics or questions that are to be investigated. A clearly laid out research question is helpful in identifying what subjects or cases should be studied and what variables are important. It is also important to consider how data are collected so that they are reliable and help achieve the research goals.

Populations and samples

Consider the following three research questions:

1. What is the average mercury content in sword sh in the Atlantic Ocean?
2. Over the last 5 years, what is the average time to degree for Duke undergraduate students?
3. Does a new drug reduce the number of deaths in patients with severe heart disease?

Each research question refers to a target population. In the rst question, the target **population** is all sword sh in the Atlantic ocean, and each sh represents a case. Often times, it is too expensive to collect data for every case in a population. Instead, a sample is taken. A **sample** represents a subset of the cases and is often a small fraction of the population. For instance, 60 sword sh (or some other number) in the population might be selected, and this sample data may be used to provide an estimate of the population average and answer the research question.

Exercise

Exercise 1.7 For the second and third questions above, identify the target population and what represents an individual case.[10]

Anecdotal Evidence

Consider the following possible responses to the three research questions:

1. A man on the news got mercury poisoning from eating sword sh, so the average mercury concentration in sword sh must be dangerously high.
2. I met two students who took more than 7 years to graduate from Duke, so it must take longer to graduate at Duke than at many other colleges.
3. My friend's dad had a heart attack and died after they gave him a new heart disease drug, so the drug must not work.

Each of the conclusions are based on some data. However, there are two problems. First, the data only represent one or two cases. Second, and more importantly, it is unclear whether these cases are actually representative of the population. Data collected in this haphazard fashion are called **anecdotal evidence**.

[10] *(2) Notice that the rst question is only relevant to students who complete their degree; the average cannot be computed using a student who never nished her degree. Thus, only Duke undergraduate students who have graduated in the last ve years represent cases in the population under consideration. Each such student would represent an individual case. (3) A person with severe heart disease represents a case. The population includes all people with severe heart disease.*



Figure 1.10: In February 2010, some media pundits cited one large snow storm as valid evidence against global warming. As comedian Jon Stewart pointed out, “It’s one storm, in one region, of one country.”

Anecdotal evidence

Be careful of data collected in a haphazard fashion. Such evidence may be true and veri able, but it may only represent extraordinary cases.

Anecdotal evidence typically is composed of unusual cases that we recall based on their striking characteristics. For instance, we are more likely to remember the two people we met who took 7 years to graduate than the six others who graduated in four years. Instead of looking at the most unusual cases, we should examine a sample of many cases that represent the population.

Sampling from a Population

We might try to estimate the time to graduation for Duke undergraduates in the last 5 years by collecting a sample of students. All graduates in the last 5 years represent the population, and graduates who are selected for review are collectively called the sample. In general, we always seek to randomly select a sample from a population. The most basic type of random selection is equivalent to how raffles are conducted. For example, in selecting graduates, we could write each graduate's name on a raffle ticket and draw 100 tickets. The selected names would represent a random sample of 100 graduates. Why pick a sample randomly? Why not just pick a sample by hand? Consider the following scenario.

Example

Example 1.8 Suppose we ask a student who happens to be majoring in nutrition to select several graduates for the study. What kind of students do you think she might collect? Do you think her sample would be representative of all graduates?

Perhaps she would pick a disproportionate number of graduates from health-related fields. Or perhaps her selection would be well-representative of the population. When selecting samples by hand, we run the risk of picking a biased sample, even if that bias is unintentional or difficult to discern.

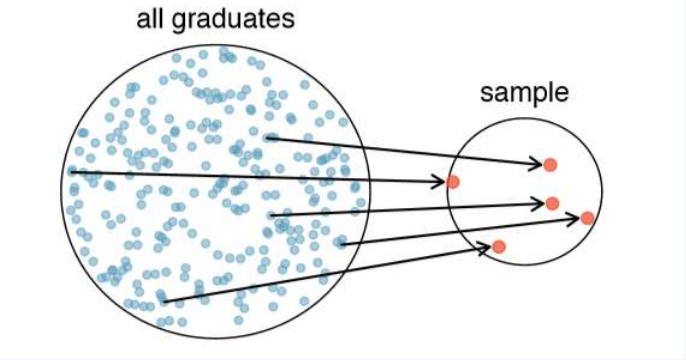


Figure 1.11: In this graphic, five graduates are randomly selected from the population to be included in the sample.

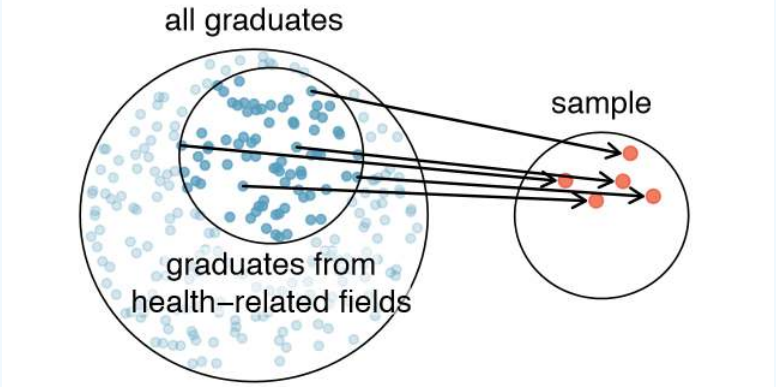


Figure 1.12: Instead of sampling from all graduates equally, a nutrition major might inadvertently pick graduates with health related majors disproportionally often.

If someone was permitted to pick and choose exactly which graduates were included in the sample, it is entirely possible that the sample could be skewed to that person's interests, which may be entirely unintentional. This introduces bias into a sample. Sampling randomly helps resolve this problem. The most basic random sample is called a **simple random sample**, and it is the equivalent of using a raffle to select cases. This means that each case in the population has an equal chance of being included and there is no implied connection between the cases in the sample.

The act of taking a simple random sample helps minimize bias, however, bias can crop up in other ways. Even when people are picked at random, e.g. for surveys, caution must be exercised if the **non-response** is high. For instance, if only 30% of the people randomly sampled for a survey actually respond, then it is unclear whether the results are **representative** of the entire population. This **non-response bias** can skew results.

Another common downfall is a **convenience sample**, where individuals who are easily accessible are more likely to be included in the sample. For instance, if a political survey is done by stopping people walking in the Bronx, this will not represent all of New York City. It is often diffcult to discern what sub-population a convenience sample represents.

Exercise

Exercise 1.9 We can easily access ratings for products, sellers, and companies through websites. These ratings are based only on those people who go out of their way to provide a rating. If 50% of online reviews for a product are negative, do you think this means that 50% of buyers are dissatisfied with the product?[11]

[11]*Answers will vary. From our own anecdotal experiences, we believe people tend to rant more about products that fell below expectations than rave about those that perform as expected. For this reason, we suspect there is a negative bias in product ratings on sites like Amazon. However, since our experiences may not be representative, we also keep an open mind should data on the subject become available.*

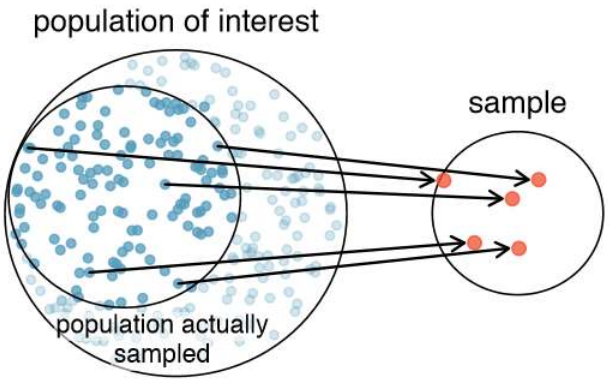


Figure 1.13: Due to the possibility of non-response, surveys studies may only reach a certain group within the population. It is difficult, and often times impossible, to completely x this problem.

Explanatory and Response Variables

Consider the following question from page 7 for the county data set:

(1) Is federal spending, on average, higher or lower in counties with high rates of poverty?

If we suspect poverty might a ect spending in a county, then poverty is the explanatory variable and federal spending is the response variable in the relationship.[12] If there are many variables, it may be possible to consider a number of them as explanatory variables.

TIP: Explanatory and response variables

To identify the explanatory variable in a pair of variables, identify which of the two is suspected of a ecting the other and plan an appropriate analysis.

$$\text{explanatory variable} \xrightarrow{\text{might affect}} \text{response variable} \tag{1.2.5.1}$$

Caution: association does not imply causation

Labeling variables as explanatory and response does not guarantee the relationship between the two is actually causal, even if there is an association identi ed between the two variables. We use these labels only to keep track of which variable we suspect a ects the other.

In some cases, there is no explanatory or response variable. Consider the following question from page 7:

(2) If homeownership is lower than the national average in one county, will the percent of multi-unit structures in that county likely be above or below the national average?

It is difficult to decide which of these variables should be considered the explanatory and response variable, i.e. the direction is ambiguous, so no explanatory or response labels are suggested here.

[12]*Sometimes the explanatory variable is called the independent variable and the response variable is called the dependent variable. However, this becomes confusing since a pair of variables might be independent or dependent, so we avoid this language.*

Introducing observational studies and experiments

There are two primary types of data collection: observational studies and experiments.

Researchers perform an **observational study** when they collect data in a way that does not directly interfere with how the data arise. For instance, researchers may collect information via surveys, review medical or company records, or follow a **cohort** of many similar individuals to study why certain diseases might develop. In each of these situations, researchers merely observe the data that arise. In general, observational studies can provide evidence of a naturally occurring association between variables, but they cannot by themselves show a causal connection.

When researchers want to investigate the possibility of a causal connection, they conduct an **experiment**. Usually there will be both an explanatory and a response variable. For instance, we may suspect administering a drug will reduce mortality in heart attack patients over the following year. To check if there really is a causal connection between the explanatory variable and the response, researchers will collect a sample of individuals and split them into groups. The individuals in each group are assigned a treatment. When individuals are randomly assigned to a group, the experiment is called a **randomized experiment**. For example, each heart attack patient in the drug trial could be randomly assigned, perhaps by flipping a coin, into one of two groups: the first group receives a **placebo** (fake treatment) and the second group receives the drug. See the case study in Section 1.1 for another example of an experiment, though that study did not employ a placebo.

TIP: association $\neq$ causation

In general, association does not imply causation, and causation can only be inferred from a randomized experiment.

This page titled 1.2.5: Overview of Data Collection Principles is shared under a CC BY-SA 3.0 license and was authored, remixed, and/or curated by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel via source content that was edited to the style and standards of the LibreTexts platform.

- **1.4: Overview of Data Collection Principles** by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel is licensed CC BY-SA 3.0. Original source: https://www.openintro.org/book/os.

1.2.6: Observational Studies and Sampling Strategies

Observational Studies

Generally, data in observational studies are collected only by monitoring what occurs, what occurs, while experiments require the primary explanatory variable in a study be assigned for each subject by the researchers. Making causal conclusions based on experiments is often reasonable. However, making the same causal conclusions based on observational data can be treacherous and is not recommended. Thus, observational studies are generally only sufficient to show associations.

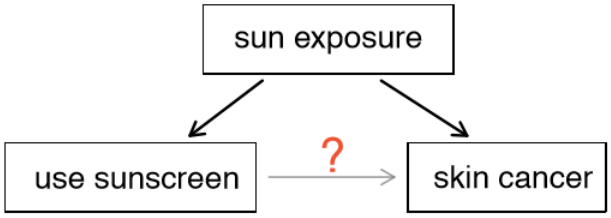
Exercise 1.2.6.1

Suppose an observational study tracked sunscreen use and skin cancer, and it was found that the more sunscreen someone used, the more likely the person was to have skin cancer. Does this mean sunscreen causes skin cancer?

Solution

No. See the paragraph following the exercise for an explanation.

Some previous research tells us that using sunscreen actually reduces skin cancer risk, so maybe there is another variable that can explain this hypothetical association between sunscreen usage and skin cancer. One important piece of information that is absent is sun exposure. If someone is out in the sun all day, she is more likely to use sunscreen and more likely to get skin cancer. Exposure to the sun is unaccounted for in the simple investigation.



Sun exposure is what is called a **confounding variable** (also called a lurking variable, confounding factor, or a confounder), which is a variable that is correlated with both the explanatory and response variables. While one method to justify making causal conclusions from observational studies is to exhaust the search for confounding variables, there is no guarantee that all confounding variables can be examined or measured. In the same way, the county data set is an observational study with confounding variables, and its data cannot easily be used to make causal conclusions.

Exercise 1.2.6.2

Figure 1.9 shows a negative association between the homeownership rate and the percentage of multi-unit structures in a county. However, it is unreasonable to conclude that there is a causal relationship between the two variables. Suggest one or more other variables that might explain the relationship visible in Figure 1.9.

Solution

Answers will vary. Population density may be important. If a county is very dense, then this may require a larger fraction of residents to live in multi-unit structures. Additionally, the high density may contribute to increases in property value, making homeownership infeasible for many residents.

Observational studies come in two forms: prospective and retrospective studies. A **prospective study** identifies individuals and collects information as events unfold. For instance, medical researchers may identify and follow a group of similar individuals over many years to assess the possible influences of behavior on cancer risk. One example of such a study is The Nurses Health Study, started in 1976 and expanded in 1989. This prospective study recruits registered nurses and then collects data from them using questionnaires. **Retrospective studies** collect data after events have taken place, e.g. researchers may review past events in medical records. Some data sets, such as county, may contain both rospectively- and retrospectively-collected variables. Local governments

prospectively collect some variables as events unfolded (e.g. retails sales) while the federal government retrospectively collected others during the 2010 census (e.g. county population counts).

Three Sampling Methods

Almost all statistical methods are based on the notion of implied randomness. If observational data are not collected in a random framework from a population, these statistical methods are not reliable. Here we consider three random sampling techniques: simple, stratified, and cluster sampling. Figure 1.14 provides a graphical representation of these techniques.

Simple random sampling is probably the most intuitive form of random sampling. Consider the salaries of Major League Baseball (MLB) players, where each player is a member of one of the league's 30 teams. To take a simple random sample of 120 baseball players and their salaries from the 2010 season, we could write the names of that season's 828 players onto slips of paper, drop the slips into a bucket, shake the bucket around until we are sure the names are all mixed up, then draw out slips until we have the sample of 120 players. In general, a sample is referred to as "simple random" if each case in the population has an equal chance of being included in the nal sample and knowing that a case is included in a sample does not provide useful information about which other cases are included.

Stratified sampling is a divide-and-conquer sampling strategy. The population is divided into groups called *strata*. The strata are chosen so that similar cases are grouped together, then a second sampling method, usually simple random sampling, is employed within each stratum. In the baseball salary example, the teams could represent the strata; some teams have a lot more money (we're looking at you, Yankees). Then we might randomly sample 4 players from each team for a total of 120 players.

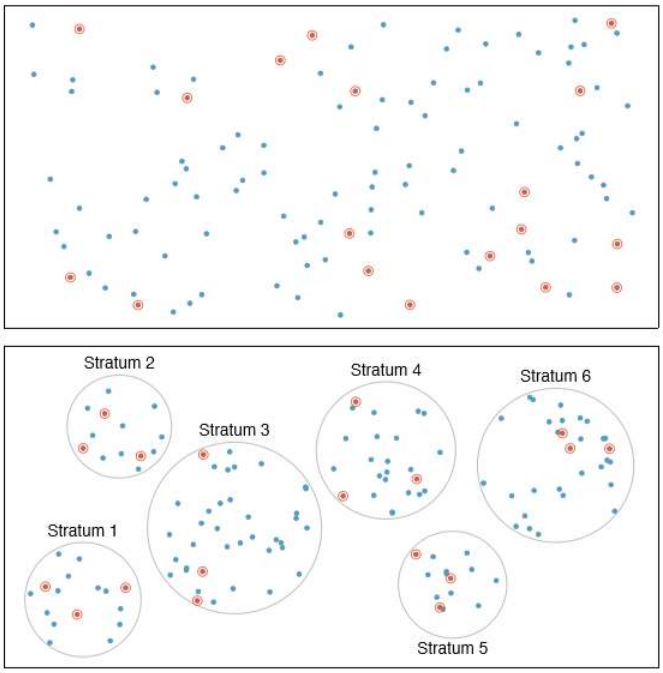


Figure 1.14: Examples of simple random, stratified, and cluster sampling. In the top panel, simple random sampling was used to randomly select the 18 cases. In the middle panel, stratified sampling was used: cases were grouped into strata, and then simple random sampling was employed within each stratum. In the bottom panel, cluster sampling was used, where data were binned into nine clusters, three of the clusters were randomly selected, and six cases were randomly sampled in each of these clusters.

Stratified sampling is especially useful when the cases in each stratum are very similar with respect to the outcome of interest. The downside is that analyzing data from a stratified sample is a more complex task than analyzing data from a simple random sample. The analysis methods introduced in this book would need to be extended to analyze data collected using stratified sampling.

Example 1.2.6.1

Why would it be good for cases within each stratum to be very similar?

Solution

We might get a more stable estimate for the subpopulation in a stratum if the cases are very similar. These improved estimates for each subpopulation will help us build a reliable estimate for the full population.

A *cluster sample* is much like a two-stage simple random sample. We break up the population into many groups, called *clusters*. Then we sample a fixed number of clusters and collect a simple random sample within each cluster. This technique is similar to stratified sampling in its process, except that there is no requirement in cluster sampling to sample from every cluster. Stratified sampling requires observations be sampled from every stratum.

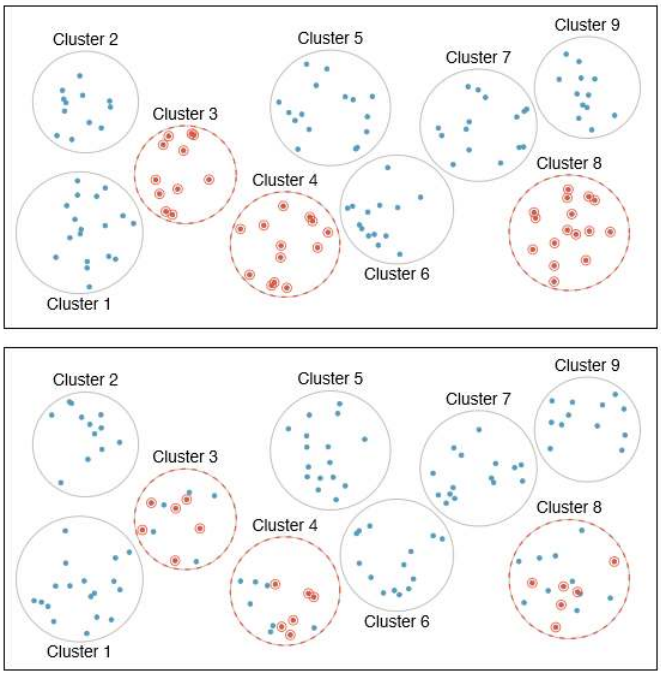


Figure 1.15: Examples of cluster and multistage sampling. In the top panel, cluster sampling was used. Here, data were binned into nine clusters, three of these clusters were sampled, and all observations within these three cluster were included in the sample. In the bottom panel, multistage sampling was used. It di↵ers from cluster sampling in that of the clusters selected, we randomly select a subset of each cluster to be included in the sample.

Sometimes cluster sampling can be a more economical random sampling technique than the alternatives. Also, unlike stratified sampling, cluster sampling is most helpful when there is a lot of case-to-case variability within a cluster but the clusters themselves don't look very different from one another. For example, if neighborhoods represented clusters, then this sampling method works best when the neighborhoods are very diverse. A downside of cluster sampling is that more advanced analysis techniques are typically required, though the methods in this book can be extended to handle such data.

Example 1.2.6.3

Suppose we are interested in estimating the malaria rate in a densely tropical portion of rural Indonesia. We learn that there are 30 villages in that part of the Indonesian jungle, each more or less similar to the next. Our goal is to test 150 individuals for malaria. What sampling method should be employed?

Solution

A simple random sample would likely draw individuals from all 30 villages, which could make data collection extremely expensive. Stratified sampling would be a challenge since it is unclear how we would build strata of similar individuals. However, cluster sampling seems like a very good idea. First, we might randomly select half the villages, then randomly select 10 people from each. This would probably reduce our data collection costs substantially in comparison to a simple random sample and would still give us reliable information.

This page titled 1.2.6: Observational Studies and Sampling Strategies is shared under a CC BY-SA 3.0 license and was authored, remixed, and/or curated by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel via source content that was edited to the style and standards of the

LibreTexts platform.

- **1.5: Observational Studies and Sampling Strategies** by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel is licensed CC BY-SA 3.0. Original source: https://www.openintro.org/book/os.

1.2.7: Experiments

Studies where the researchers assign treatments to cases are called **experiments**. When this assignment includes randomization, e.g. using a coin ip to decide which treatment a patient receives, it is called a **randomized experiment**. Randomized experiments are fundamentally important when trying to show a causal connection between two variables.

Principles of experimental design

Randomized experiments are generally built on four principles.

Controlling. Researchers assign treatments to cases, and they do their best to **control** any other differences in the groups. For example, when patients take a drug in pill form, some patients take the pill with only a sip of water while others may have it with an entire glass of water. To control for water consumption, a doctor may ask all patients to drink a 12 ounce glass of water with the pill.

Randomization. Researchers randomize patients into treatment groups to account for variables that cannot be controlled. For example, some patients may be more susceptible to a disease than others due to their dietary habits. Randomizing patients into the treatment or control group helps even out such differences, and it also prevents accidental bias from entering the study.

Replication. The more cases researchers observe, the more accurately they can estimate the effect of the explanatory variable on the response. In a single study, we **replicate** by collecting a sufficiently large sample. Additionally, a group of scientists may replicate an entire study to verify an earlier nding.

Blocking. Researchers sometimes know or suspect that variables, other than the treatment, inuence the response. Under these circumstances, they may rst group individuals based on this variable into **blocks** and then randomize cases within each block to the treatment groups. This strategy is often referred to as **blocking**. For instance, if we are looking at the effect of a drug on heart attacks, we might rst split patients in the study into low-risk and high-risk blocks, then randomly assign half the patients from each block to the control group and the other half to the treatment group, as shown in Figure 1.15. This strategy ensures each treatment group has an equal number of low-risk and high-risk patients.

It is important to incorporate the rst three experimental design principles into any study, and this book describes applicable methods for analyzing data from such experiments. Blocking is a slightly more advanced technique, and statistical methods in this book may be extended to analyze data collected using blocking.

Reducing bias in human experiments

Randomized experiments are the gold standard for data collection, but they do not ensure an unbiased perspective into the cause and effect relationships in all cases. Human studies are perfect examples where bias can unintentionally arise. Here we reconsider a study where a new drug was used to treat heart attack patients.17 In particular, researchers wanted to know if the drug reduced deaths in patients.

[17]*Anturane Reinfarction Trial Research Group. 1980. Sul npyrazone in the prevention of sudden death after myocardial infarction. New England Journal of Medicine 302(5):250-256.*

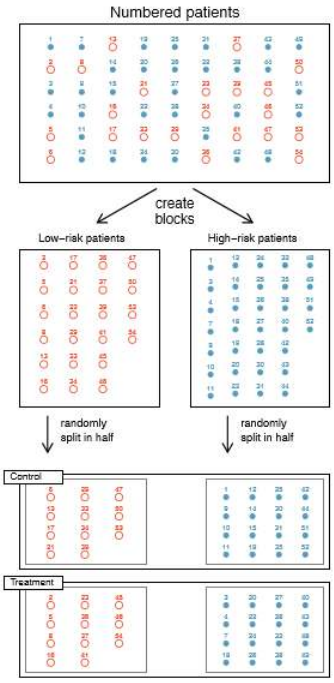


Figure 1.16: Blocking using a variable depicting patient risk. Patients are first divided into low-risk and high-risk blocks, then each block is evenly divided into the treatment groups using randomization. This strategy ensures an equal representation of patients in each treatment group from both the low-risk and high-risk categories.

These researchers designed a randomized experiment because they wanted to draw causal conclusions about the drug's effect. Study volunteers18 were randomly placed into two study groups. One group, the **treatment group**, received the drug. The other group, called the **control group**, did not receive any drug treatment.

Put yourself in the place of a person in the study. If you are in the treatment group, you are given a fancy new drug that you anticipate will help you. On the other hand, a person in the other group doesn't receive the drug and sits idly, hoping her participation doesn't increase her risk of death. These perspectives suggest there are actually two effects: the one of interest is the effectiveness of the drug, and the second is an emotional effect that is difficult to quantify.

Researchers aren't usually interested in the emotional effect, which might bias the study. To circumvent this problem, researchers do not want patients to know which group they are in. When researchers keep the patients uninformed about their treatment, the study is said to be **blind**. But there is one problem: if a patient doesn't receive a treatment, she will know she is in the control group. The solution to this problem is to give fake treatments to patients in the control group. A fake treatment is called a **placebo**, and an effective placebo is the key to making a study truly blind. A classic example of a placebo is a sugar pill that is made to look like the actual treatment pill. Often times, a placebo results in a slight but real improvement in patients. This effect has been dubbed the **placebo effect**.

The patients are not the only ones who should be blinded: doctors and researchers can accidentally bias a study. When a doctor knows a patient has been given the real treatment, she might inadvertently give that patient more attention or care than a patient that she knows is on the placebo. To guard against this bias, which again has been found to have a measurable effect in some instances, most modern studies employ a **double-blind** setup where doctors or researchers who interact with patients are, just like the patients, unaware of who is or is not receiving the treatment.[19]

Exercise 1.14 Look back to the study in Section 1.1 where researchers were testing whether stents were effective at reducing strokes in at-risk patients. Is this an experiment? Was the study blinded? Was it double-blinded?[20]

This page titled 1.2.7: Experiments is shared under a CC BY-SA 3.0 license and was authored, remixed, and/or curated by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel via source content that was edited to the style and standards of the LibreTexts platform.

- **1.6: Experiments** by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel is licensed CC BY-SA 3.0. Original source: https://www.openintro.org/book/os.

1.2.8: How Not to Do Statistics

Many studies are conducted and conclusions are made. However, there are occasions where the study is not conducted in the correct manner or the conclusion is not correctly made based on the data. There are many things that you should question when you read a study. There are many reasons for the study to have bias in it. Bias is where a study may have a certain slant or preference for a certain result. The following are a list of some of the questions or issues you should consider to help decide if there is bias in a study.

One of the first issues you should ask is who funded the study. If the entity that sponsored the study stands to gain either profits or notoriety from the results, then you should question the results. It doesn't mean that the results are wrong, but you should scrutinize them on your own to make sure they are sound. As an example if a study says that genetically modified foods are safe, and the study was funded by a company that sells genetically modified food, then one may question the validity of the study. Since the company funds the study and their profits rely on people buying their food, there may be bias.

An experiment could have **lurking or confounding variables** when you cannot rule out the possibility that the observed effect is due to some other variable rather than the factor being studied. An example of this is when you give fertilizer to some plants and no fertilizer to others, but the no fertilizer plants also are placed in a location that doesn't receive direct sunlight. You won't know if the plants that received the fertilizer grew taller because of the fertilizer or the sunlight. Make sure you design experiments to eliminate the effects of confounding variables by controlling all the factors that you can.

Overgeneralization

Overgeneralization is where you do a study on one group and then try to say that it will happen on all groups. An example is doing cancer treatments on rats. Just because the treatment works on rats does not mean it will work on humans. Another example is that until recently most FDA medication testing had been done on white males of a particular age. There is no way to know how the medication affects other genders, ethnic groups, age groups, and races. The new FDA guidelines stresses using individuals from different groups.

Cause and Effect

Cause and effect is where people decide that one variable causes the other just because the variables are related or correlated. Unless the study was done as an experiment where a variable was controlled, you cannot say that one variable caused the other. Most likely there is another variable that caused both. As an example, there is a relationship between number of drownings at the beach and ice cream sales. This does not mean that ice cream sales increasing causes people to drown. Most likely the cause for both increasing is the heat.

Sampling Error

This is the difference between the sample results and the true population results. This is unavoidable, and results in the fact that samples are different from each other. As an example, if you take a sample of 5 people's height in your class, you will get 5 numbers. If you take another sample of 5 people's heights in your class, you will likely get 5 different numbers.

Nonsampling Error

This is where the sample is collected poorly either through a biased sample or through error in measurements. Care should be taken to avoid this error.

Lastly, there should be care taken in considering the difference between **statistical significance versus practical significance**. This is a major issue in statistics. Something could be statistically significance, which means that a statistical test shows there is evidence to show what you are trying to prove. However, in practice it doesn't mean much or there are other issues to consider. As an example, suppose you find that a new drug for high blood pressure does reduce the blood pressure of patients. When you look at the improvement it actually doesn't amount to a large difference. Even though statistically there is a change, it may not be worth marketing the product because it really isn't that big of a change. Another consideration is that you find the blood pressure medication does improve a person's blood pressure, but it has serious side effects or it costs a great deal for a prescription. In this case, it wouldn't be practical to use it. In both cases, the study is shown to be statistically significant, but practically you don't want

to use the medication. The main thing to remember in a statistical study is that the statistics is only part of the process. You also want to make sure that there is practical significance too.

Surveys

Surveys have their own areas of bias that can occur. A few of the issues with surveys are in the wording of the questions, the ordering of the questions, the manner the survey is conducted, and the response rate of the survey.

The wording of the questions can cause **hidden bias**, which is where the questions are asked in a way that makes a person respond a certain way. An example is that a poll was done where people were asked if they believe that there should be an amendment to the constitution protecting a woman's right to choose. About 60% of all people questioned said yes. Another poll was done where people were asked if they believe that there should be an amendment to the constitution protecting the life of an unborn child. About 60% of all people questioned said yes. These two questions deal with the same issue, though giving opposite results, but how the question was asked affected the outcome.

The ordering of the question can also cause hidden bias. An example of this is if you were asked if there should be a fine for texting while driving, but proceeding that question is the question asking if you text while drive. By asking a person if they actually partake in the activity, that person now personalizes the question and that might affect how they answer the next question of creating the fine.

Non-response

Non-response is where you send out a survey but not everyone returns the survey. You can calculate the response rate by dividing the number of returns by the number of surveys sent. Most response rates are around 30-50%. A response rate less than 30% is very poor and the results of the survey are not valid. To reduce non-response, it is better to conduct the surveys in person, though these are very expensive. Phones are the next best way to conduct surveys, emails can be effective, and physical mailings are the least desirable way to conduct surveys.

Voluntary response

Voluntary response is where people are asked to respond via phone, email or online. The problem with these is that only people who really care about the topic are likely to call or email. These surveys are not scientific and the results from these surveys are not valid. Note: all studies involve volunteers. The difference between a voluntary response survey and a scientific study is that in a scientific study the researchers ask the individuals to be involved, while in a voluntary response survey the individuals become involved on their own choosing.

Example 1.2.8.1: Bias in a Study

Suppose a mathematics department at a community college would like to assess whether computer-based homework improves students' test scores. They use computer-based homework in one classroom with one teacher and use traditional paper and pencil homework in a different classroom with a different teacher. The students using the computer-based homework had higher test scores. What is wrong with this experiment?

Solution

Since there were different teachers, you do not know if the better test scores are because of the teacher or the computer-based homework. A better design would be have the same teacher teach both classes. The control group would utilize traditional paper and pencil homework and the treatment group would utilize the computer-based homework. Both classes would have the same teacher, and the students would be split between the two classes randomly. The only difference between the two groups should be the homework method. Of course, there is still variability between the students, but utilizing the same teacher will reduce any other confounding variables.

Example 1.2.8.2: Cause and Effect

Determine if the one variable did cause the change in the other variable.

a. Cinnamon was giving to a group of people who have diabetes, and then their blood glucose levels were measured a time period later. All other factors for each person were kept the same. Their glucose levels went down. Did the cinnamon cause the reduction?
b. There is a link between spray on tanning products and lung cancer. Does that mean that spray on tanning products cause lung cancer?

Solution

a. Since this was a study where the use of cinnamon was controlled, and all other factors were kept constant from person to person, then any changes in glucose levels can be attributed to the use of cinnamon
b. Since there is only a link, and not a study controlling the use of the tanning spray, then you cannot say that increased use causes lung cancer. You can say that there is a link, and that there could be a cause, but you cannot say for sure that the spray causes the cancer.

Example 1.2.8.3: Generalization

a. A researcher conducts a study on the use of ibuprofen on humans and finds that it is safe. Does that mean that all species can use ibuprofen?
b. Aspirin has been used for years to bring down fevers in humans. Originally it was tested on white males between the ages of 25 and 40 and found to be safe. Is it safe to give to everyone?

Solution

a. No. Just because a drug is safe to use on one species doesn't mean it is safe to use for all species. In fact, ibuprofen is toxic to cats.
b. No. Just because one age group can use it doesn't mean it is safe to use for all age groups. In fact, there has been a link between giving a child under the age of 19 aspirin when they have a fever and Reye's syndrome.

Homework

1. Suppose there is a study where a researcher conducts an experiment to show that deep breathing exercises helps to lower blood pressure. The researcher takes two groups of people and has one group to perform deep breathing exercises and a series of aerobic exercises every day and the other group was asked to refrain from any exercises. The researcher found that the group performing the deep breathing exercises and the aerobic exercises had lower blood pressure. Discuss any issue with this study.
2. Suppose a car dealership offers a low interest rate and a longer payoff period to customers or a high interest rate and a shorter payoff period to customers, and most customers choose the low interest rate and longer payoff period, does that mean that most customers want a lower interest rate? Explain.
3. Over the years it has been said that coffee is bad for you. When looking at the studies that have shown that coffee is linked to poor health, you will see that people who tend to drink coffee don't sleep much, tend to smoke, don't eat healthy, and tend to not exercise. Can you say that the coffee is the reason for the poor health or is there a lurking variable that is the actual cause? Explain.
4. When researchers were trying to figure out what caused polio, they saw a connection between ice cream sales and polio. As ice cream sales increased so did the incident of polio. Does that mean that eating ice cream causes polio? Explain your answer.
5. There is a positive correlation between having a discussion of gun control, which usually occur after a mass shooting, and the sale of guns. Does that mean that the discussion of gun control increases the likelihood that people will buy more guns? Explain.
6. There is a study that shows that people who are obese have a vitamin D deficiency. Does that mean that obesity causes a deficiency in vitamin D? Explain.
7. A study was conducted that shows that polytetrafluoroethylene (PFOA) (Teflon is made from this chemical) has an increase risk of tumors in lab mice. Does that mean that PFOA's have an increased risk of tumors in humans? Explain.
8. Suppose a telephone poll is conducted by contacting U.S. citizens via landlines about their view of gay marriage. Suppose over 50% of those called do not support gay marriage. Does that mean that you can say over 50% of all people in the U.S. do not

support gay marriage? Explain.

9. Suppose that it can be shown to be statistically significant that a smaller percentage of the people are satisfied with your business. The percentage before was 87% and is now 85%. Do you change how you conduct business? Explain?
10. You are testing a new drug for weight loss. You find that the drug does in fact statistically show a weight loss. Do you market the new drug? Why or why not?
11. There was an online poll conducted about whether the mayor of Auckland, New Zealand, should resign due to an affair. The majority of people participating said he should. Should the mayor resign due to the results of this poll? Explain.
12. An online poll showed that the majority of Americans believe that the government covered up events of 9/11. Does that really mean that most Americans believe this? Explain.
13. A survey was conducted at a college asking all employees if they were satisfied with the level of security provided by the security department. Discuss how the results of this question could be biased.
14. An employee survey says, “Employees at this institution are very satisfied with working here. Please rate your satisfaction with the institution.” Discuss how this question could create bias.
15. A survey has a question that says, “Most people are afraid that they will lose their house due to economic collapse. Choose what you think is the biggest issue facing the nation today.
    a. Economic collapse
    b. Foreign policy issues
    c. Environmental concerns.” Discuss how this question could create bias.
16. A survey says, “Please rate the career of Roberto Clemente, one of the best right field baseball players in the world.” Discuss how this question could create bias.

Answer

See solutions

This page titled 1.2.8: How Not to Do Statistics is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Kathryn Kozak via source content that was edited to the style and standards of the LibreTexts platform.

- **1.4: How Not to Do Statistics** by Kathryn Kozak is licensed CC BY-SA 4.0. Original source: https://s3-us-west-2.amazonaws.com/oerfiles/statsusingtech2.pdf.

1.2.9: Exercises

Do Athletes Get Special Treatment?

Prerequisites

Levels of Measurement



Figure **1.2.9.1**: Runners

The Board of Trustees at a university commissioned a top management-consulting firm to address the admission processes for academic and athletic programs. The consulting firm wrote a report discussing the trade-off between maintaining academic and athletic excellence. One of their key findings was:

> The standard for an athlete's admission, as reflected in SAT scores alone, is lower than the standard for non-athletes by as much as 20 percent, with the weight of this difference being carried by the so-called "revenue sports" of football and basketball. Athletes are also admitted through a different process than the one used to admit non-athlete students.

What do you think?

Based on what you have learned in this chapter about measurement scales, does it make sense to compare SAT scores using percentages? Why or why not?

> As you may know, the SAT has an arbitrarily-determined lower limit on test scores of 200. Therefore, SAT is measured on either an ordinal scale or, at most, an interval scale. However, it is clearly not measured on a ratio scale. Therefore, it is not meaningful to report SAT score differences in terms of percentages. For example, consider the effect of subtracting 200 from every student's score so that the lowest possible score is 0. How would that affect the difference as expressed in percentages?

Statistical Errors in Politics

Prerequisites

Inferential Statistics



Figure **1.2.9.2***: Survey*

An article about ignorance of statistics in politics quotes a politician commenting on why the "American Community Survey" should be eliminated:

> "We're spending $70 per person to fill this out. That's just not cost effective, especially since in the end this is not a scientific survey. It's a random survey."

What do you think?

What is wrong with this statement? Despite the error in this statement, what type of sampling could be done so that the sample will be more likely to be representative of the population?

> Randomness is what makes the survey scientific. If the survey were not random, then it would be biased and therefore statistically meaningless, especially since the survey is conducted to make generalizations about the American population. Stratified sampling would likely be more representative of the population.

Reference

Mark C. C., scientopia.org

Contributors and Attributions

- Online Statistics Education: A Multimedia Course of Study (http://onlinestatbook.com/). Project Leader: David M. Lane, Rice University.
- Denise Harvey and David Lane

This page titled 1.2.9: Exercises is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **1.14: Statistical Literacy** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

CHAPTER OVERVIEW

2: Introduction to R

Learning Objectives

Having finished this chapter, you should be able to:

- Interact with an RMarkdown notebook in RStudio
- Describe the difference between a variable and a function
- Describe the different types of variables
- Create a vector or data frame and access its elements
- Install and load an R library
- Load data from a file and view the data frame

This chapter is the first of several distributed throughout the book that will introduce you to increasingly sophisticated things that you can do using the R programming language. The name "R" is a play on the names of the two authors of the software package (Ross Ihaka and Robert Gentleman) as well as an homage to an older statistical software package called "S". R has become one of the most popular programming languages for statistical analysis and "data science". Unlike general-purpose programming languages such as Python or Java, R is purpose-built for statistics. That doesn't mean that you can't do more general things with it, but the place where it really shines is in data analysis and statistics.

2.1: Why Programming Is Hard to Learn
2.2: Using RStudio
2.3: Installing R
2.4: Getting Started with R
2.5: Variables
2.6: Functions
2.7: Letting RStudio Help You with Your Commands
2.8: Vectors
2.9: Math with Vectors
2.10: Data Frames
2.11: Using R Libraries
2.12: Installing and Loading Packages
2.13: Using Comments
2.14: Navigating the File System
2.15: Loading and Saving Data
2.16: Useful Things to Know about Variables
2.17: Factors
2.18: Data frames
2.19: Suggested Readings and Videos

This page titled 2: Introduction to R is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

2.1: Why Programming Is Hard to Learn

Programming a computer is a skill, just like playing a musical instrument or speaking a second language. And just like those skills, it takes a lot of work to get good at it — the only way to acquire a skill is through practice. There is nothing special or magical about people who are experts, other than the quality and quantity of their experience! However, not all practice is equally effective. A large amount of psychological research has shown that practice needs to be *deliberate*, meaning that it focuses on developing the specific skills that one needs to perform the skill, at a level that is always pushing one's ability.

If you have never programmed before, then it's going to seem hard, just as it would seem hard for a native English speaker to start speaking Mandarin. However, just as a beginning guitarist needs to learn to play their scales, we will teach you how to perform the basics of programming, which you can then use to do more powerful things.

One of the most important aspects of computer programming is that you can try things to your heart's content; the worst thing that can happen is that the program will crash. Trying new things and making mistakes is one of the keys to learning.

The hardest part of programming is figuring out why something didn't work, which we call *debugging*. In programming, things are going to go wrong in ways that are often confusing and opaque. Every programmer has a story about spending hours trying to figure out why something didn't work, only to realize that the problem was completely obvious. The more practice you get, the better you will get at figuring out how to fix these errors. But there are a few strategies that can be helpful.

3.1.1 Use the web

In particular, you should take advantage of the fact that there are millions of people programming in R around the world, so nearly any error message you see has already been seen by someone else. Whenever I experience an error that I don't understand, the first thing that I do is to copy and paste the error message into a search engine Often this will provide several pages discussing the problem and the ways that people have solved it.

3.1.2 Rubber duck debugging

The idea behind *rubber duck debugging* is to pretend that you are trying to explain what your code is doing to an inanimate object, like a rubber duck. Often, the process of explaning it aloud is enough to help you find the problem.

This page titled 2.1: Why Programming Is Hard to Learn is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

- **3.1: Why Programming Is Hard to Learn** by Russell A. Poldrack is licensed CC BY-NC 4.0. Original source: https://statsthinking21.github.io/statsthinking21-core-site.

2.2: Using RStudio

When I am using R in my own work, I generally use a free software package called RStudio, which provides a number of nice tools for working with R. In particular, RStudio provides the ability to create “notebooks” that mix together R code and text (formatted using the Markdown text formatting system). In fact, this book is written using exactly that system! You can see the R code used to generate this book here.

This page titled 2.2: Using RStudio is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

- **3.2: Using RStudio** by Russell A. Poldrack is licensed CC BY-NC 4.0. Original source: https://statsthinking21.github.io/statsthinking21-core-site.

2.3: Installing R

Okay, enough with the sales pitch. Let's get started. Just as with any piece of software, R needs to be installed on a "computer", which is a magical box that does cool things and delivers free ponies. Or something along those lines: I may be confusing computers with the iPad marketing campaigns. Anyway, R is freely distributed online, and you can download it from the R homepage, which is:

http://cran.r-project.org/

At the top of the page – under the heading "Download and Install R" – you'll see separate links for Windows users, Mac users, and Linux users. If you follow the relevant link, you'll see that the online instructions are pretty self-explanatory, but I'll walk you through the installation anyway. As of this writing, the current version of R is 3.0.2 (Frisbee Sailing"), but they usually issue updates every six months, so you'll probably have a newer version.[14]

2.3.1 Installing R on a Windows computer

The CRAN homepage changes from time to time, and it's not particularly pretty, or all that well-designed quite frankly. But it's not difficult to find what you're after. In general you'll find a link at the top of the page with the text "Download R for Windows". If you click on that, it will take you to a page that offers you a few options. Again, at the very top of the page you'll be told to click on a link that says to click here if you're installing R for the first time. That's probably what you want. This will take you to a page that has a prominent link at the top called "Download R 3.0.2 for Windows". That's the one you want. Click on that and your browser should start downloading a file called `R-3.0.2-win.exe` , or whatever the equivalent version number is by the time you read this. The file for version 3.0.2 is about 54MB in size, so it may take some time depending on how fast your internet connection is. Once you've downloaded the file, double click to install it. As with any software you download online, Windows will ask you some questions about whether you trust the file and so on. After you click through those, it'll ask you where you want to install it, and what components you want to install. The default values should be fine for most people, so again, just click through. Once all that is done, you should have R installed on your system. You can access it from the Start menu, or from the desktop if you asked it to add a shortcut there. You can now open up R in the usual way if you want to, but what I'm going to suggest is that instead of doing that you should now install RStudio.

2.3.2 Installing R on a Mac

When you click on the Mac OS X link, you should find yourself on a page with the title "R for Mac OS X". The vast majority of Mac users will have a fairly recent version of the operating system: as long as you're running Mac OS X 10.6 (Snow Leopard) or higher, then you'll be fine.[15] There's a fairly prominent link on the page called "R-3.0.2.pkg", which is the one you want. Click on that link and you'll start downloading the installer file, which is (not surprisingly) called `R-3.0.2.pkg` . It's about 61MB in size, so the download can take a while on slower internet connections.

Once you've downloaded `R-3.0.2.pkg` , all you need to do is open it by double clicking on the package file. The installation should go smoothly from there: just follow all the instructions just like you usually do when you install something. Once it's finished, you'll find a file called `R.app` in the Applications folder. You can now open up R in the usual way[16] if you want to, but what I'm going to suggest is that instead of doing that you should now install RStudio.

2.3.3 Installing R on a Linux computer

If you're successfully managing to run a Linux box, regardless of what distribution, then you should find the instructions on the website easy enough. You can compile R from source yourself if you want, or install it through your package management system, which will probably have R in it. Alternatively, the CRAN site has precompiled binaries for Debian, Red Hat, Suse and Ubuntu and has separate instructions for each. Once you've got R installed, you can run it from the command line just by typing `R` . However, if you're feeling envious of Windows and Mac users for their fancy GUIs, you can download RStudio too.

2.3.4 Downloading and installing RStudio

Okay, so regardless of what operating system you're using, the last thing that I told you to do is to download RStudio. To understand why I've suggested this, you need to understand a little bit more about R itself. The term R doesn't really refer to a

specific application on your computer. Rather, it refers to the underlying statistical language. You can use this language through lots of different applications. When you install R initially, it comes with one application that lets you do this: it's the R.exe application on a Windows machine, and the R.app application on a Mac. But that's not the only way to do it. There are lots of different applications that you can use that will let you interact with R. One of those is called RStudio, and it's the one I'm going to suggest that you use. RStudio provides a clean, professional interface to R that I find much nicer to work with than either the Windows or Mac defaults. Like R itself, RStudio is free software: you can find all the details on their webpage. In the meantime, you can download it here:

http://www.RStudio.org/

When you visit the RStudio website, you'll probably be struck by how much cleaner and simpler it is than the CRAN website,[17] and how obvious it is what you need to do: click the big green button that says "Download".

When you click on the download button on the homepage it will ask you to choose whether you want the desktop version or the server version. You want the desktop version. After choosing the desktop version it will take you to a page (http://www.RStudio.org/download/desktop) that shows several possible downloads: there's a different one for each operating system. However, the nice people at RStudio have designed the webpage so that it automatically recommends the download that is most appropriate for your computer. Click on the appropriate link, and the RStudio installer file will start downloading.

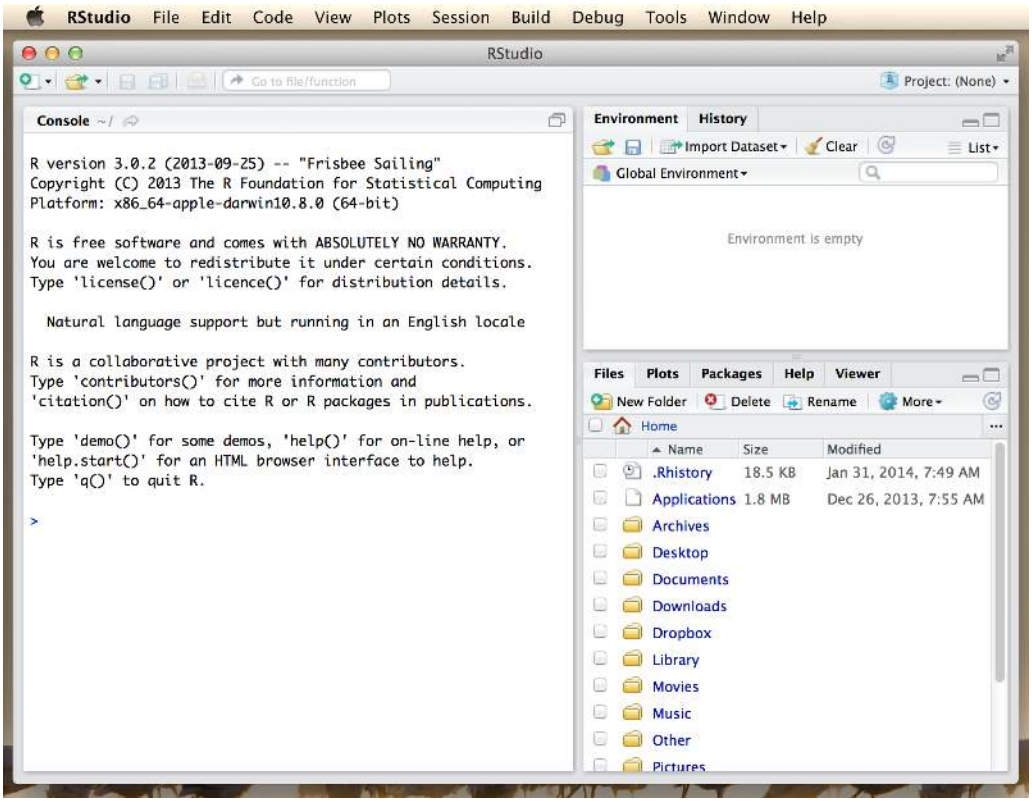


Figure 3.1: An R session in progress running through RStudio. The picture shows RStudio running on a Mac, but the Windows interface is almost identical.

Once it's finished downloading, open the installer file in the usual way to install RStudio. After it's finished installing, you can start R by opening RStudio. You don't need to open R.app or R.exe in order to access R. RStudio will take care of that for you. To illustrate what RStudio looks like, Figure 3.1 shows a screenshot of an R session in progress. In this screenshot, you can see that it's running on a Mac, but it looks almost identical no matter what operating system you have. The Windows version looks more like a Windows application (e.g., the menus are attached to the application window and the colour scheme is slightly different), but it's more or less identical. There are a few minor differences in where things are located in the menus (I'll point them out as we go along) and in the shortcut keys, because RStudio is trying to "feel" like a proper Mac application or a proper Windows application, and this means that it has to change its behaviour a little bit depending on what computer it's running on. Even so, these differences

are very small: I started out using the Mac version of RStudio and then started using the Windows version as well in order to write these notes.

The only "shortcoming" I've found with RStudio is that – as of this writing – it's still a work in progress. The "problem" is that they keep improving it. New features keep turning up the more recent releases, so there's a good chance that by the time you read this book there will be a version out that has some really neat things that weren't in the version that I'm using now.

2.3.5 Starting up R

One way or another, regardless of what operating system you're using and regardless of whether you're using RStudio, or the default GUI, or even the command line, it's time to open R and get started. When you do that, the first thing you'll see (assuming that you're looking at the ***R console***, that is) is a whole lot of text that doesn't make much sense. It should look something like this:

```
R version 3.0.2 (2013-09-25) -- "Frisbee Sailing"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin10.8.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Most of this text is pretty uninteresting, and when doing real data analysis you'll never really pay much attention to it. The important part of it is this…

```
>
```

… which has a flashing cursor next to it. That's the ***command prompt***. When you see this, it means that R is waiting patiently for you to do something!

This page titled 2.3: Installing R is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **3.1: Installing R** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

2.4: Getting Started with R

When we work with R, we often do this using a *command line* in which we type commands and it responds to those commands. In the simplest case, if we just type in a number, it will simply respond with that number. Go into the R console and type the number 3. You should see somethign like this:

```
> 3
[1] 3
```

The `>` symbol is the *command prompt*, which is prompting you to type something in. The next line (`[1] 3`) is R's answer. Let's try something a bit more complicated:

```
> 3 + 4
[1] 7
```

R spits out the answer to whatever you type in, as long as it can figure it out. Now let's try typing in a word:

```
> hello
Error: object 'hello' not found
```

What? Why did this happen? When R encounters a letter or word, it assumes that it is referring to the name of a *variable* — think of `X` from high school algebra. We will return to variables in a little while, but if we want R to print out the word *hello* then we need to contain it in quotation marks, telling R that it is a *character string*.

```
> "hello"
[1] "hello"
```

There are many types of variables in R. You have already seen two examples: integers (like the number 3) and character strings (like the word "hello"). Another important one is *real numbers*, which are the most common kind of numbers that we will deal with in statistics, which span the entire number line including the spaces in between the integers. For example:

```
> 1/3
[1] 0.33
```

In reality the result should be 0.33 followed by an infinite number of threes, but R only shows us two decimal points in this example.

Another kind of variable is known as a *logical* variable, because it is based on the idea from logic that a statement can be either true or false. In R, these are capitalized (`TRUE` and `FALSE`).

To determine whether a statement is true or not, we use *logical operators*. You are already familiar with some of these, like the greater-than (`>`) and less-than (`<`) operators.

```
> 1 < 3
[1] TRUE
> 2 > 4
[1] FALSE
```

Often we want to know whether two numbers are equal or not equal to one another. There are special operators in R to do this: `==` for equals, and `!=` for not-equals:

```
> 3 == 3
[1] TRUE
> 4 != 4
[1] FALSE
```

This page titled 2.4: Getting Started with R is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

- **3.3: Getting Started with R** by Russell A. Poldrack is licensed CC BY-NC 4.0. Original source: https://statsthinking21.github.io/statsthinking21-core-site.

2.5: Variables

A *variable* is a symbol that stands for another value (just like "X" in algebra). We can create a variable by assigning a value to it using the `<-` operator. If we then type the name of the variable R will print out its value.

```
> x <- 4
> x
[1] 4
```

The variable now stands for the value that it contains, so we can perform operations on it and get the same answer as if we used the value itself.

```
> x + 3
[1] 7
> x == 5
[1] FALSE
```

We can change the value of a variable by simply assigning a new value to it.

```
> x <- x + 1
> x
[1] 5
```

A note: You can also use the equals sign `=` instead of the `<-`

This page titled 2.5: Variables is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

- **3.4: Variables** by Russell A. Poldrack is licensed CC BY-NC 4.0. Original source: https://statsthinking21.github.io/statsthinking21-core-site.

2.6: Functions

A *function* is an operator that takes some input and gives an output based on the input. For example, let's say that have a number and we want to determine its absolute value. R has a function called `abs()` that takes in a number and outputs its absolute value:

```
> x <- -3
> abs(x)
[1] 3
```

Most functions take an input like the `abs()` function (which we call an *argument*), but some also have special keywords that can be used to change how the function works. For example, the `rnorm()` function generates random numbers from a normal distribution (which we will learn more about later). Have a look at the help page for this function by typing `help(rnorm)` in the console, which will cause a help page to appear below. The section of the help page for the `rnorm()` function shows the following:

```
rnorm(n, mean = 0, sd = 1)

Arguments

n     number of observations.

mean    vector of means.

sd   vector of standard deviations.
```

You can also obtain some examples of how the function is used by typing `example(rnorm)` in the console.

We can see that the rnorm function has two arguments, *mean* and *sd*, that are shown to be equal to specific values. This means that those values are the *default* settings, so that if you don't do anything, then the function will return random numbers with a mean of 0 and a standard deviation of 1. The other argument, *n*, does not have a default value. Try typing in the function `rnorm()` with no arguments and see what happens — it will return an error telling you that the argument "n" is missing and does not have a default value.

If we wanted to create random numbers with a different mean and standard deviation (say mean == 100 and standard deviation == 15), then we could simply set those values in the function call. Let's say that we would like 5 random numbers from this distribution:

```
> my_random_numbers <- rnorm(5, mean=100, sd=15)
> my_random_numbers
[1] 104 115 101  97 115
```

You will see that I set the variable to the name `my_random_numbers` . In general, it's always good to be as descriptive as possible when creating variables; rather than calling them *x* or *y*, use names that describe the actual contents. This will make it much easier to understand what's going on once things get more complicated.

This page titled 2.6: Functions is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

- **3.5: Functions** by Russell A. Poldrack is licensed CC BY-NC 4.0. Original source: https://statsthinking21.github.io/statsthinking21-core-site.

2.7: Letting RStudio Help You with Your Commands

Time for a bit of a digression. At this stage you know how to type in basic commands, including how to use R functions. And it's probably beginning to dawn on you that there are a *lot* of R functions, all of which have their own arguments. You're probably also worried that you're going to have to remember all of them! Thankfully, it's not that bad. In fact, very few data analysts bother to try to remember all the commands. What they really do is use tricks to make their lives easier. The first (and arguably most important one) is to use the internet. If you don't know how a particular R function works, Google it. Second, you can look up the R help documentation. I'll talk more about these two tricks in Section 4.12. But right now I want to call your attention to a couple of simple tricks that RStudio makes available to you.

2.7.1 Autocomplete using "tab"

The first thing I want to call your attention to is the *autocomplete* ability in RStudio.[32]

Let's stick to our example above and assume that what you want to do is to round a number. This time around, start typing the name of the function that you want, and then hit the "tab" key. RStudio will then display a little window like the one shown in Figure 3.2. In this figure, I've typed the letters `ro` at the command line, and then hit tab. The window has two panels. On the left, there's a list of variables and functions that start with the letters that I've typed shown in black text, and some grey text that tells you where that variable/function is stored. Ignore the grey text for now: it won't make much sense to you until we've talked about packages in Section 4.2. In Figure 3.2 you can see that there's quite a few things that start with the letters `ro` : there's something called `rock` , something called `round` , something called `round.Date` and so on. The one we want is `round` , but if you're typing this yourself you'll notice that when you hit the tab key the window pops up with the top entry (i.e., `rock`) highlighted. You can use the up and down arrow keys to select the one that you want. Or, if none of the options look right to you, you can hit the escape key ("esc") or the left arrow key to make the window go away.

In our case, the thing we want is the `round` option, so we'll select that. When you do this, you'll see that the panel on the right changes. Previously, it had been telling us something about the `rock` data set (i.e., "Measurements on 48 rock samples…") that is distributed as part of R. But when we select `round` , it displays information about the `round()` function, exactly as it is shown in Figure 3.2. This display is really handy. The very first thing it says is `round(x, digits = 0)` : what this is telling you is that the `round()` function has two arguments. The first argument is called `x` , and it doesn't have a default value. The second argument is `digits` , and it has a default value of 0. In a lot of situations, that's all the information you need. But RStudio goes a bit further, and provides some additional information about the function underneath. Sometimes that additional information is very helpful, sometimes it's not: RStudio pulls that text from the R help documentation, and my experience is that the helpfulness of that documentation varies wildly. Anyway, if you've decided that `round()` is the function that you want to use, you can hit the right arrow or the enter key, and RStudio will finish typing the rest of the function name for you.

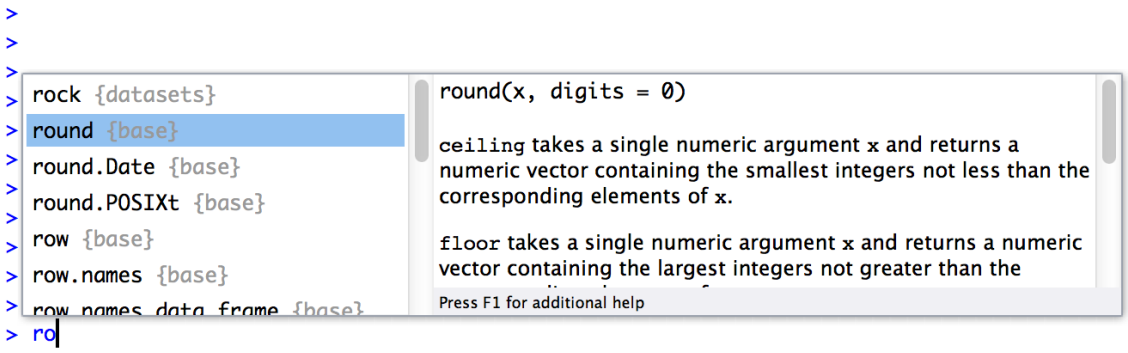


Figure 3.2: Start typing the name of a function or a variable, and hit the "tab" key. RStudio brings up a little dialog box like this one that lets you select the one you want, and even prints out a little information about it.

Start typing the name of a function or a variable, and hit the "tab" key. RStudio brings up a little dialog box like this one that lets you select the one you want, and even prints out a little information about it.

The RStudio autocomplete tool works slightly differently if you've already got the name of the function typed and you're now trying to type the arguments. For instance, suppose I've typed `round(` into the console, and *then* I hit tab. RStudio is smart

enough to recognise that I already know the name of the function that I want, because I've already typed it! Instead, it figures that what I'm interested in is the *arguments* to that function. So that's what pops up in the little window. You can see this in Figure **??**. Again, the window has two panels, and you can interact with this window in exactly the same way that you did with the window shown in 3.2. On the left hand panel, you can see a list of the argument names. On the right hand side, it displays some information about what the selected argument does.

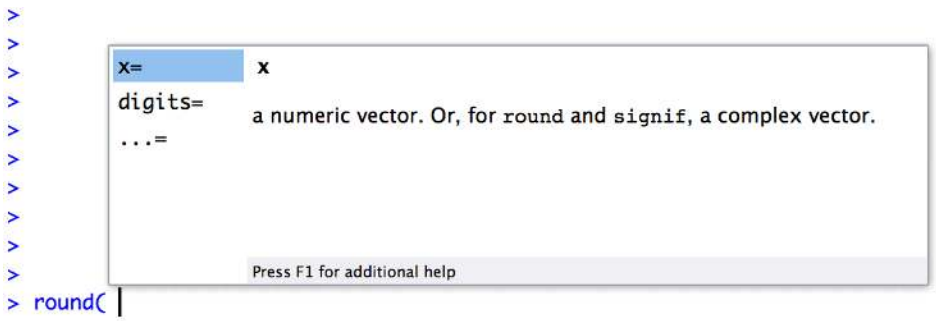


Figure 3.3: If you've typed the name of a function already along with the left parenthesis and then hit the "tab" key, RStudio brings up a different window to the one shown above. This one lists all the arguments to the function on the left, and information about each argument on the right.

If you've typed the name of a function already along with the left parenthesis and then hit the "tab" key, RStudio brings up a different window to the one shown in Figure 3.2. This one lists all the arguments to the function on the left, and information about each argument on the right.

2.7.2 Browsing your command history

One thing that R does automatically is keep track of your "command history". That is, it remembers all the commands that you've previously typed. You can access this history in a few different ways. The simplest way is to use the up and down arrow keys. If you hit the up key, the R console will show you the most recent command that you've typed. Hit it again, and it will show you the command before that. If you want the text on the screen to go away, hit escape[33] Using the up and down keys can be really handy if you've typed a long command that had one typo in it. Rather than having to type it all again from scratch, you can use the up key to bring up the command and fix it.

The second way to get access to your command history is to look at the history panel in RStudio. On the upper right hand side of the RStudio window you'll see a tab labelled "History". Click on that, and you'll see a list of all your recent commands displayed in that panel: it should look something like Figure **??**. If you double click on one of the commands, it will be copied to the R console. (You can achieve the same result by selecting the command you want with the mouse and then clicking the "To Console" button).[34]

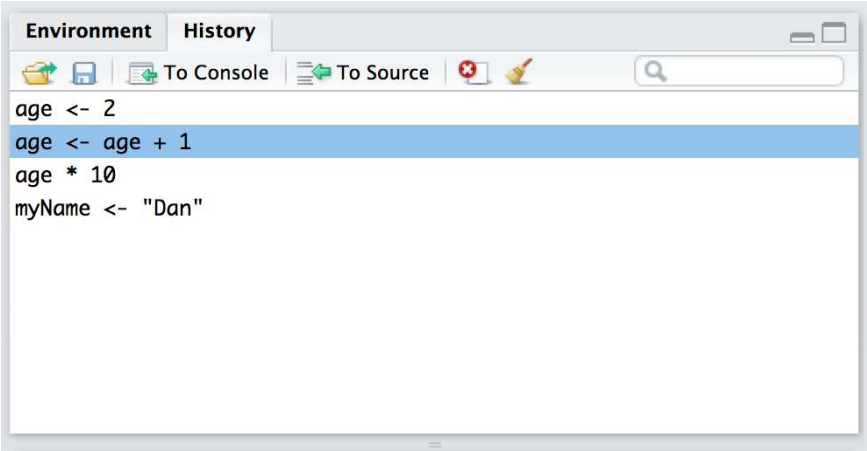


Figure 3.4: The history panel is located in the top right hand side of the RStudio window. Click on the word "History" and it displays this panel.

This page titled 2.7: Letting RStudio Help You with Your Commands is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **3.6: Letting RStudio Help You with Your Commands** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

2.8: Vectors

You may have noticed that the `my_random_numbers` created above wasn't like the variables that we had seen before — it contained a number of values in it. We refer to this kind of variable as a *vector*.

If you want to create your own new vector, you can do that using the `c()` function:

```
> my_vector <- c(4, 5, 6)
> my_vector
[1] 4 5 6
```

You can access the individual elements within a vector by using square brackets along with a number that refers to the location within the vector. These *index* values start at 1, which is different from many other programming languages that start at zero. Let's say we want to see the value in the second place of the vector:

```
> my_vector[2]
[1] 5
```

You can also look at a range of positions, by putting the start and end locations with a colon in between:

```
> my_vector[2:3]
[1] 5 6
```

You can also change the values of specific locations using the same indexing:

```
> my_vector[3] <- 7
> my_vector
[1] 4 5 7
```

This page titled 2.8: Vectors is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

- **3.6: Vectors** by Russell A. Poldrack is licensed CC BY-NC 4.0. Original source: https://statsthinking21.github.io/statsthinking21-core-site.

2.9: Math with Vectors

You can apply mathematical operations to the elements of a vector just as you would with a single number:

```
> my_vector <- c(4, 5, 6)
> my_vector_times_ten <- my_vector*10
> my_vector_times_ten
[1] 40 50 60
```

You can also apply mathematical operations on pairs of vectors. In this case, each matching element is used for the operation.

```
> my_first_vector <- c(1,2,3)
> my_second_vector <- c(10, 20, 20)
> my_first_vector + my_second_vector
[1] 11 22 23
```

We can also apply logical operations across vectors; again, this will return a vector with the operation applied to the pairs of values at each position.

```
> vector_a <- c(1,2,3)
> vector_b <- c(1,2,4)
> vector_a == vector_b
[1]  TRUE  TRUE FALSE
```

Most functions will work with vectors just as they would with a single number. For example, let's say we wanted to obtain the trignometric sine for each of a set of values. We could create a vector and pass it to the `sin()` function, which will return as many sine values as there are input values:

```
> my_angle_values <- c(0, 1, 2)
> my_sin_values <- sin(my_angle_values)
> my_sin_values
[1] 0.00 0.84 0.91
k
```

This page titled 2.9: Math with Vectors is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

- **3.7: Math with Vectors** by Russell A. Poldrack is licensed CC BY-NC 4.0. Original source: https://statsthinking21.github.io/statsthinking21-core-site.

2.10: Data Frames

Often in a dataset we will have a number of different variables that we want to work with. Instead of having a different named variable that stores each one, it is often useful to combine all of the separate variables into a single package, which is referred to as a *data frame*.

If you are familiar with a spreadsheet (say from Microsoft Excel) then you already have a basic understanding of a data frame. Let's say that we have values of price and mileage for three different types of cars. We could start by creating a variable for each one, making sure that the three cars are in the same order for each of the variables:

```
car_model <- c("Ford Fusion", "Hyundai Accent", "Toyota Corolla")
car_price <- c(25000, 16000, 18000)
car_mileage <- c(27, 36, 32)
```

We can then combine these into a single data frame, using the `data.frame()` function. I like to use "_df" in the names of data frames just to make clear that it's a data frame, so we will call this one "cars_df":

```
cars_df <- data.frame(model=car_model, price=car_price, mileage=car_mileage)
```

We can view the data frame by using the `View()` function:

```
View(cars_df)
```

Which will present a view of the data frame much like a spreadsheet, as shown in Figure 2.1:

	model	price	mileage
1	Ford Fusion	25000	27
2	Hyundai Accent	16000	36
3	Toyota Corolla	18000	32

Figure 2.1: A view of the cars data frame generated by the View() function.

Each of the columns in the data frame contains one of the variables, with the name that we gave it when we created the data frame. We can access each of those columns using the `$` operator. For example, if we wanted to access the mileage variable, we would combine the name of the data frame with the name of the variable as follows:

```
> cars_df$mileage
[1] 27 36 32
```

This is just like any other vector, in that we can refer to its individual values using square brackets as we did with regular vectors:

```
> cars_df$mileage[3]
[1] 32
```

In some of the examples in the book, you will see something called a *tibble*; this is basically a souped-up version of a data frame, and can be treated mostly in the same way.

This page titled 2.10: Data Frames is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

- **3.8: Data Frames** by Russell A. Poldrack is licensed CC BY-NC 4.0. Original source: https://statsthinking21.github.io/statsthinking21-core-site.

2.11: Using R Libraries

Many of the useful features in R are not contained in the primary R package, but instead come from *libraries* that have been developed by various members of the R community. For example, the `ggplot2` package provides a number of features for visualizing data, as we will see in a later chapter. Before we can use a package, we need to install it on our system, using the `install.packages()` function:

```
> install.packages("ggplot2")
trying URL 'https://cran.rstudio.com/...
Content type 'application/x-gzip' length 3961383 bytes (3.8 MB)
==================================================
downloaded 3.8 MB


The downloaded binary packages are in
    /var/folders/.../downloaded_packages
```

This will automatically download the package from the Comprehensive R Archive Network (CRAN) and install it on your system. Once it's installed, you can then load the library using the `library()` function:

```
> library(ggplot2)
```

After loading the function, you can now access all of its features. If you want to learn more about its features, you can find them using the help function:

```
> help(ggplot2)
```

This page titled 2.11: Using R Libraries is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

- **3.9: Using R Libraries** by Russell A. Poldrack is licensed CC BY-NC 4.0. Original source: https://statsthinking21.github.io/statsthinking21-core-site.

2.12: Installing and Loading Packages

In this section I discuss R ***packages***, since almost all of the functions you might want to use in R come in packages. A package is basically just a big collection of functions, data sets and other R objects that are all grouped together under a common name. Some packages are already installed when you put R on your computer, but the vast majority of them of R packages are out there on the internet, waiting for you to download, install and use them.

When I first started writing this book, Rstudio didn't really exist as a viable option for using R, and as a consequence I wrote a very lengthy section that explained how to do package management using raw R commands. It's not actually terribly hard to work with packages that way, but it's clunky and unpleasant. Fortunately, we don't have to do things that way anymore. In this section, I'll describe how to work with packages using the Rstudio tools, because they're so much simpler. Along the way, you'll see that whenever you get Rstudio to do something (e.g., install a package), you'll actually see the R commands that get created. I'll explain them as we go, because I think that helps you understand what's going on.

However, before we get started, there's a critical distinction that you need to understand, which is the difference between having a package ***installed*** on your computer, and having a package ***loaded*** in R. As of this writing, there are just over 5000 R packages freely available "out there" on the internet.[42] When you install R on your computer, you don't get all of them: only about 30 or so come bundled with the basic R installation. So right now there are about 30 packages "installed" on your computer, and another 5000 or so that are not installed. So that's what installed means: it means "it's on your computer somewhere". The critical thing to remember is that just because something is on your computer doesn't mean R can use it. In order for R to be able to *use* one of your 30 or so installed packages, that package must also be "loaded". Generally, when you open up R, only a few of these packages (about 7 or 8) are actually loaded. Basically what it boils down to is this:

A package must be installed before it can be loaded.

A package must be loaded before it can be used.

This two step process might seem a little odd at first, but the designers of R had very good reasons to do it this way,[43] and you get the hang of it pretty quickly.

2.12.1 package panel in Rstudio

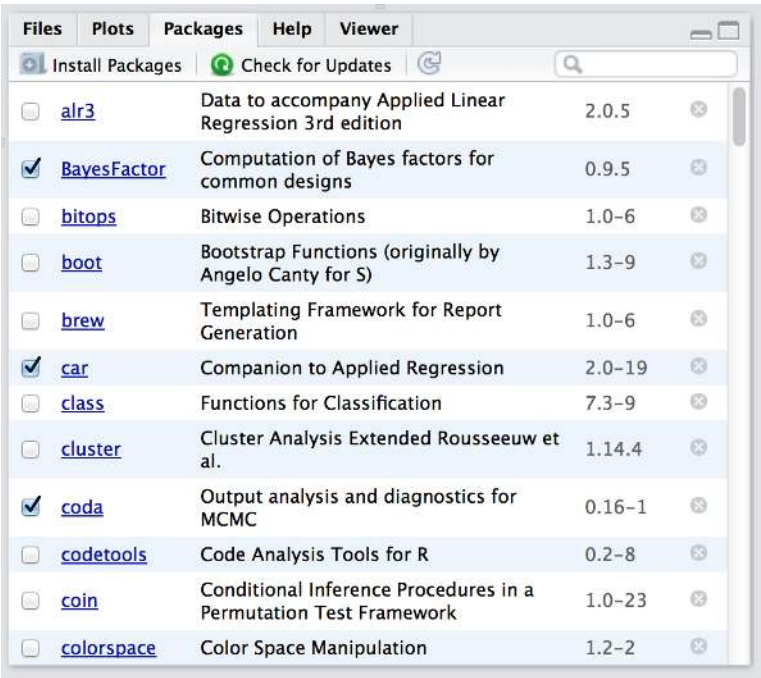


Figure 4.1: The packages panel.

Right, lets get started. The first thing you need to do is look in the lower right hand panel in Rstudio. You'll see a tab labelled "Packages". Click on the tab, and you'll see a list of packages that looks something like Figure 4.1. Every row in the panel

corresponds to a different package, and every column is a useful piece of information about that package.[44] Going from left to right, here's what each column is telling you:

- The check box on the far left column indicates whether or not the package is loaded.
- The one word of text immediately to the right of the check box is the name of the package.
- The short passage of text next to the name is a brief description of the package.
- The number next to the description tells you what version of the package you have installed.
- The little x-mark next to the version number is a button that you can push to uninstall the package from your computer (you almost never need this).

2.12.2 Loading a package

That seems straightforward enough, so let's try loading and unloading packades. For this example, I'll use the `foreign` package. The `foreign` package is a collection of tools that are very handy when R needs to interact with files that are produced by other software packages (e.g., SPSS). It comes bundled with R, so it's one of the ones that you have installed already, but it won't be one of the ones loaded. Inside the `foreign` package is a function called `read.spss()`. It's a handy little function that you can use to import an SPSS data file into R, so let's pretend we want to use it. Currently, the `foreign` package isn't loaded, so if I ask R to tell me if it knows about a function called `read.spss()` it tells me that there's no such thing…

```
exists( "read.spss" )
```

```
## [1] FALSE
```

Now let's load the package. In Rstudio, the process is dead simple: go to the package tab, find the entry for the `foreign` package, and check the box on the left hand side. The moment that you do this, you'll see a command like this appear in the R console:

```
library("foreign", lib.loc="/Library/Frameworks/R.framework/Versions/3.0/Resources/li
```

The `lib.loc` bit will look slightly different on Macs versus on Windows, because that part of the command is just Rstudio telling R where to look to find the installed packages. What I've shown you above is the Mac version. On a Windows machine, you'll probably see something that looks like this:

```
library("foreign", lib.loc="C:/Program Files/R/R-3.0.2/library")
```

But actually it doesn't matter much. The `lib.loc` bit is almost always unnecessary. Unless you've taken to installing packages in idiosyncratic places (which is something that you can do if you really want) R already knows where to look. So in the vast majority of cases, the command to load the `foreign` package is just this:

```
library("foreign")
```

Throughout this book, you'll often see me typing in `library()` commands. You don't actually have to type them in yourself: you can use the Rstudio package panel to do all your package loading for you. The only reason I include the `library()` commands sometimes is as a reminder to you to make sure that you have the relevant package loaded. Oh, and I suppose we should check to see if our attempt to load the package actually worked. Let's see if R now knows about the existence of the `read.spss()` function…

```
exists( "read.spss" )
```

```
## [1] TRUE
```

Yep. All good.

2.12.3 Unloading a package

Sometimes, especially after a long session of working with R, you find yourself wanting to get rid of some of those packages that you've loaded. The Rstudio package panel makes this exactly as easy as loading the package in the first place. Find the entry corresponding to the package you want to unload, and uncheck the box. When you do that for the `foreign` package, you'll see this command appear on screen:

```
detach("package:foreign", unload=TRUE)
```

And the package is unloaded. We can verify this by seeing if the `read.spss()` function still `exists()`:

```
exists( "read.spss" )
```

```
## [1] FALSE
```

Nope. Definitely gone.

2.12.4 extra comments

Sections 4.2.2 and 4.2.3 cover the main things you need to know about loading and unloading packages. However, there's a couple of other details that I want to draw your attention to. A concrete example is the best way to illustrate. One of the other packages that you already have installed on your computer is the `Matrix` package, so let's load that one and see what happens:

```
library( Matrix )

## Loading required package: lattice
```

This is slightly more complex than the output that we got last time, but it's not too complicated. The `Matrix` package makes use of some of the tools in the `lattice` package, and R has kept track of this dependency. So when you try to load the `Matrix` package, R recognises that you're also going to need to have the `lattice` package loaded too. As a consequence, *both* packages get loaded, and R prints out a helpful little note on screen to tell you that it's done so.

R is pretty aggressive about enforcing these dependencies. Suppose, for example, I try to unload the `lattice` package while the `Matrix` package is still loaded. This is easy enough to try: all I have to do is uncheck the box next to "lattice" in the packages panel. But if I try this, here's what happens:

```
detach("package:lattice", unload=TRUE)

## Error: package `lattice' is required by `Matrix' so will not be detached
```

R refuses to do it. This can be quite useful, since it stops you from accidentally removing something that you still need. So, if I want to remove both `Matrix` and `lattice`, I need to do it in the correct order

Something else you should be aware of. Sometimes you'll attempt to load a package, and R will print out a message on screen telling you that something or other has been "masked". This will be confusing to you if I don't explain it now, and it actually ties very closely to the whole reason why R forces you to load packages separately from installing them. Here's an example. Two of the package that I'll refer to a lot in this book are called `car` and `psych`. The `car` package is short for "Companion to Applied Regression" (which is a really great book, I'll add), and it has a lot of tools that I'm quite fond of. The `car` package was written by a guy called John Fox, who has written a lot of great statistical tools for social science applications. The `psych` package was written by William Revelle, and it has a lot of functions that are very useful for psychologists in particular, especially in regards to psychometric techniques. For the most part, `car` and `psych` are quite unrelated to each other. They do different things, so not

surprisingly almost all of the function names are different. But… there's one exception to that. The `car` package and the `psych` package *both* contain a function called `logit()`.[45] This creates a naming conflict. If I load both packages into R, an ambiguity is created. If the user types in `logit(100)`, should R use the `logit()` function in the `car` package, or the one in the `psych` package? The answer is: R uses whichever package you loaded most recently, and it tells you this very explicitly. Here's what happens when I load the `car` package, and then afterwards load the `psych` package:

```
library(car)
```

```
## Loading required package: carData
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following object is masked from 'package:car':
##
##   logit
```

The output here is telling you that the `logit` object (i.e., function) in the `car` package is no longer accessible to you. It's been hidden (or "masked") from you by the one in the `psych` package.[46]

2.12.5 Downloading new packages

One of the main selling points for R is that there are thousands of packages that have been written for it, and these are all available online. So whereabouts online are these packages to be found, and how do we download and install them? There is a big repository of packages called the "Comprehensive R Archive Network" (CRAN), and the easiest way of getting and installing a new package is from one of the many CRAN mirror sites. Conveniently for us, R provides a function called `install.packages()` that you can use to do this. Even *more* conveniently, the Rstudio team runs its own CRAN mirror and Rstudio has a clean interface that lets you install packages without having to learn how to use the `install.packages()` command[47]

Using the Rstudio tools is, again, dead simple. In the top left hand corner of the packages panel (Figure 4.1) you'll see a button called "Install Packages". If you click on that, it will bring up a window like the one shown in Figure 4.2.

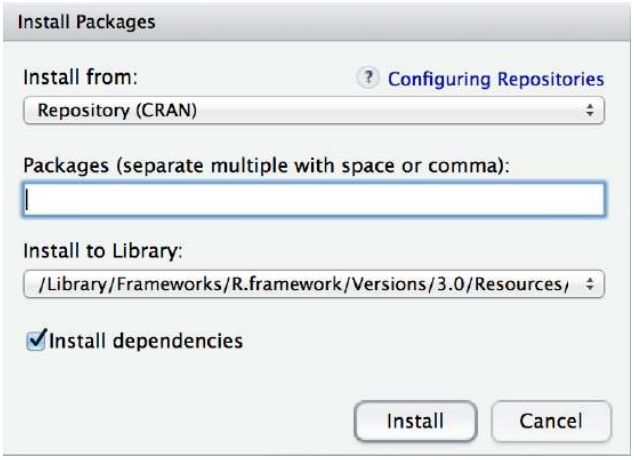


Figure 4.2: The package installation dialog box in Rstudio

There are a few different buttons and boxes you can play with. Ignore most of them. Just go to the line that says "Packages" and start typing the name of the package that you want. As you type, you'll see a dropdown menu appear (Figure 4.3), listing names of

packages that start with the letters that you've typed so far.

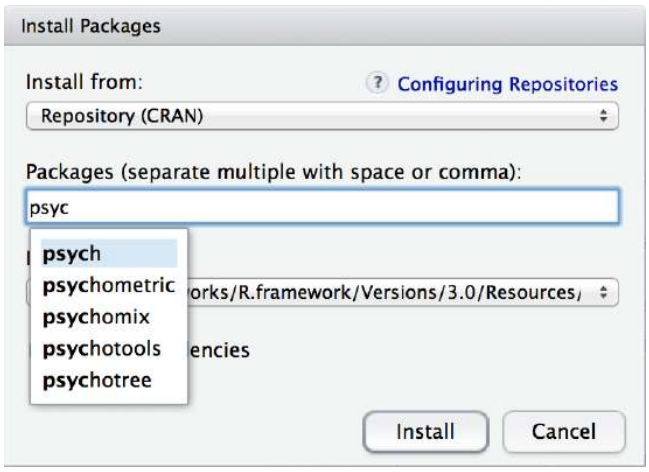


Figure 4.3: When you start typing, you'll see a dropdown menu suggest a list of possible packages that you might want to install

You can select from this list, or just keep typing. Either way, once you've got the package name that you want, click on the install button at the bottom of the window. When you do, you'll see the following command appear in the R console:

```
install.packages("psych")
```

This is the R command that does all the work. R then goes off to the internet, has a conversation with CRAN, downloads some stuff, and installs it on your computer. You probably don't care about all the details of R's little adventure on the web, but the `install.packages()` function is rather chatty, so it reports a bunch of gibberish that you really aren't all that interested in:

```
trying URL 'http://cran.rstudio.com/bin/macosx/contrib/3.0/psych_1.4.1.tgz'
Content type 'application/x-gzip' length 2737873 bytes (2.6 Mb)
opened URL
==================================================
downloaded 2.6 Mb


The downloaded binary packages are in
  /var/folders/cl/thhsyrz53g73q0w1kb5z3l_80000gn/T//RtmpmQ9VT3/downloaded_packages
```

Despite the long and tedious response, all thar really means is "I've installed the psych package". I find it best to humour the talkative little automaton. I don't actually read any of this garbage, I just politely say "thanks" and go back to whatever I was doing.

2.12.6 Updating R and R packages

Every now and then the authors of packages release updated versions. The updated versions often add new functionality, fix bugs, and so on. It's generally a good idea to update your packages periodically. There's an `update.packages()` function that you can use to do this, but it's probably easier to stick with the Rstudio tool. In the packages panel, click on the "Update Packages" button. This will bring up a window that looks like the one shown in Figure 4.4. In this window, each row refers to a package that needs to be updated. You can to tell R which updates you want to install by checking the boxes on the left. If you're feeling lazy and just want to update everything, click the "Select All" button, and then click the "Install Updates" button. R then prints out a *lot* of garbage on the screen, individually downloading and installing all the new packages. This might take a while to complete depending on how good your internet connection is. Go make a cup of coffee. Come back, and all will be well.

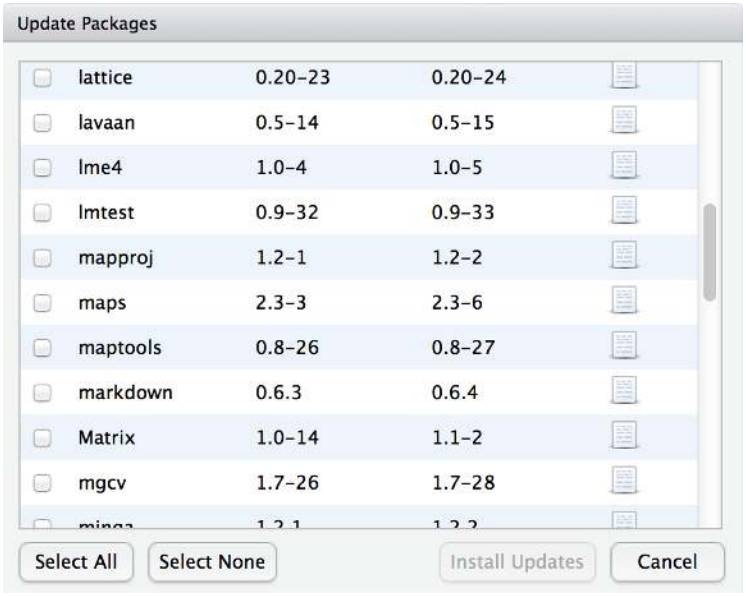


Figure 4.4: The Rstudio dialog box for updating packages

About every six months or so, a new version of R is released. You can't update R from within Rstudio (not to my knowledge, at least): to get the new version you can go to the CRAN website and download the most recent version of R, and install it in the same way you did when you originally installed R on your computer. This used to be a slightly frustrating event, because whenever you downloaded the new version of R, you would lose all the packages that you'd downloaded and installed, and would have to repeat the process of re-installing them. This was pretty annoying, and there were some neat tricks you could use to get around this. However, newer versions of R don't have this problem so I no longer bother explaining the workarounds for that issue.

2.12.7 What packages does this book use?

There are several packages that I make use of in this book. The most prominent ones are:

- lot of interesting high-powered tools: it's just a small collection of handy little things that I think can be useful to novice users. As you get more comfortable with R this package should start to feel pretty useless to you.
- `psych`. This package, written by William Revelle, includes a lot of tools that are of particular use to psychologists. In particular, there's several functions that are particularly convenient for producing analyses or summaries that are very common in psych, but less common in other disciplines.
- `car`. This is the *Companion to Applied Regression* package, which accompanies the excellent book of the same name by (Fox and Weisberg 2011). It provides a lot of very powerful tools, only some of which we'll touch in this book.

Besides these three, there are a number of packages that I use in a more limited fashion: `gplots`, `sciplot`, `foreign`, `effects`, `R.matlab`, `gdata`, `lmtest`, and probably one or two others that I've missed. There are also a number of packages that I refer to but don't actually use in this book, such as `reshape`, `compute.es`, `HistData` and `multcomp` among others. Finally, there are a number of packages that provide more advanced tools that I hope to talk about in future versions of the book, such as `sem`, `ez`, `nlme` and `lme4`. In any case, whenever I'm using a function that isn't in the core packages, I'll make sure to note this in the text.

This page titled 2.12: Installing and Loading Packages is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **4.2: Installing and Loading Packages** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

2.13: Using Comments

Before discussing any of the more complicated stuff, I want to introduce the ***comment*** character, `#`. It has a simple meaning: it tells R to ignore everything else you've written on this line. You won't have much need of the `#` character immediately, but it's very useful later on when writing scripts (see Chapter 8). However, while you don't need to use it, I want to be able to include comments in my R extracts. For instance, if you read this:[41]

```
seeker <- 3.1415            # create the first variable
lover <- 2.7183             # create the second variable
keeper <- seeker * lover    # now multiply them to create a third one
print( keeper )             # print out the value of 'keeper'
```

```
## [1] 8.539539
```

it's a lot easier to understand what I'm doing than if I just write this:

```
seeker <- 3.1415
lover <- 2.7183
keeper <- seeker * lover
print( keeper )
```

```
## [1] 8.539539
```

You might have already noticed that the code extracts in Chapter 3 included the `#` character, but from now on, you'll start seeing `#` characters appearing in the extracts, with some human-readable explanatory remarks next to them. These are still perfectly legitimate commands, since R knows that it should ignore the `#` character and everything after it. But hopefully they'll help make things a little easier to understand.

This page titled 2.13: Using Comments is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **4.1: Using Comments** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

2.14: Navigating the File System

In this section I talk a little about how R interacts with the file system on your computer. It's not a terribly interesting topic, but it's useful. As background to this discussion, I'll talk a bit about how file system locations work in Section 4.4.1. Once upon a time *everyone* who used computers could safely be assumed to understand how the file system worked, because it was impossible to successfully use a computer if you didn't! However, modern operating systems are much more "user friendly", and as a consequence of this they go to great lengths to hide the file system from users. So these days it's not at all uncommon for people to have used computers most of their life and not be familiar with the way that computers organise files. If you already know this stuff, skip straight to Section 4.4.2. Otherwise, read on. I'll try to give a brief introduction that will be useful for those of you who have never been forced to learn how to navigate around a computer using a DOS or UNIX shell.

2.14.1 file system itself

In this section I describe the basic idea behind file locations and file paths. Regardless of whether you're using Window, Mac OS or Linux, every file on the computer is assigned a (fairly) human readable address, and every address has the same basic structure: it describes a *path* that starts from a *root* location , through as series of *folders* (or if you're an old-school computer user, *directories*), and finally ends up at the file.

On a Windows computer the root is the physical drive[50] on which the file is stored, and for most home computers the name of the hard drive that stores all your files is C: and therefore most file names on Windows begin with C:. After that comes the folders, and on Windows the folder names are separated by a `\` symbol. So, the complete path to this book on my Windows computer might be something like this:

```
C:\Users\danRbook\LSR.pdf
```

and what that *means* is that the book is called LSR.pdf, and it's in a folder called `book` which itself is in a folder called dan which itself is … well, you get the idea. On Linux, Unix and Mac OS systems, the addresses look a little different, but they're more or less identical in spirit. Instead of using the backslash, folders are separated using a forward slash, and unlike Windows, they don't treat the physical drive as being the root of the file system. So, the path to this book on my Mac might be something like this:

```
/Users/dan/Rbook/LSR.pdf
```

So that's what we mean by the "path" to a file. The next concept to grasp is the idea of a ***working directory*** and how to change it. For those of you who have used command line interfaces previously, this should be obvious already. But if not, here's what I mean. The working directory is just "whatever folder I'm currently looking at". Suppose that I'm currently looking for files in Explorer (if you're using Windows) or using Finder (on a Mac). The folder I currently have open is my user directory (i.e., `C:\Users\dan` or `/Users/dan`). That's my current working directory.

The fact that we can imagine that the program is "in" a particular directory means that we can talk about moving *from* our current location *to* a new one. What that means is that we might want to specify a new location in relation to our current location. To do so, we need to introduce two new conventions. Regardless of what operating system you're using, we use . to refer to the current working directory, and .. to refer to the directory above it. This allows us to specify a path to a new location in relation to our current location, as the following examples illustrate. Let's assume that I'm using my Windows computer, and my working directory is `C:\Users\danRbook`). The table below shows several addresses in relation to my current one:

Table 4.1: Basic arithmetic operations in R. These five operators are used very frequently throughout the text, so it's important to be familiar with them at the outset.

absolute path (i.e., from root)	relative path (i.e. from C:)
C:\Users\dan	..
C:\Users	..\.. \
C:\Users\danRbook\source	.\source
C:\Users\dan\nerdstuff	..\nerdstuff

There's one last thing I want to call attention to: the `~` directory. I normally wouldn't bother, but R makes reference to this concept sometimes. It's quite common on computers that have multiple users to define `~` to be the user's home directory. On my Mac, for instance, the home directory `~` for the "dan" user is `\Users\dan\`. And so, not surprisingly, it is possible to define other directories in terms of their relationship to the home directory. For example, an alternative way to describe the location of the `LSR.pdf` file on my Mac would be

```
~Rbook\LSR.pdf
```

That's about all you really need to know about file paths. And since this section already feels too long, it's time to look at how to navigate the file system in R.

2.14.2 Navigating the file system using the R console

In this section I'll talk about how to navigate this file system from within R itself. It's not particularly user friendly, and so you'll probably be happy to know that Rstudio provides you with an easier method, and I will describe it in Section 4.4.4. So in practice, you won't *really* need to use the commands that I babble on about in this section, but I do think it helps to see them in operation at least once before forgetting about them forever.

Okay, let's get started. When you want to load or save a file in R it's important to know what the working directory is. You can find out by using the `getwd()` command. For the moment, let's assume that I'm using Mac OS or Linux, since there's some subtleties to Windows. Here's what happens:

```
getwd()
## [1] "/Users/dan"
```

We can change the working directory quite easily using `setwd()`. The `setwd()` function has only the one argument, `dir`, is a character string specifying a path to a directory, or a path relative to the working directory. Since I'm currently located at `/Users/dan`, the following two are equivalent:

```
setwd("/Users/dan/Rbook/data")
setwd("./Rbook/data")
```

Now that we're here, we can type `list.files()` command to get a listing of all the files in that directory. Since this is the directory in which I store all of the data files that we'll use in this book, here's what we get as the result:

```
list.files()
## [1] "afl24.Rdata"             "aflsmall.Rdata"          "aflsmall2.Rdata"
## [4] "agpp.Rdata"              "all.zip"                 "annoying.Rdata"
## [7] "anscombesquartet.Rdata"  "awesome.Rdata"           "awesome2.Rdata"
## [10] "booksales.csv"           "booksales.Rdata"         "booksales2.csv"
## [13] "cakes.Rdata"             "cards.Rdata"             "chapek9.Rdata"
## [16] "chico.Rdata"             "clinicaltrial_old.Rdata" "clinicaltrial.Rdata"
## [19] "coffee.Rdata"            "drugs.wmc.rt.Rdata"      "dwr_all.Rdata"
## [22] "effort.Rdata"            "happy.Rdata"             "harpo.Rdata"
## [25] "harpo2.Rdata"            "likert.Rdata"            "nightgarden.Rdata"
## [28] "nightgarden2.Rdata"      "parenthood.Rdata"        "parenthood2.Rdata"
## [31] "randomness.Rdata"        "repeated.Rdata"          "rtfm.Rdata"
## [34] "salem.Rdata"             "zeppo.Rdata"
```

Not terribly exciting, I'll admit, but it's useful to know about. In any case, there's only one more thing I want to make a note of, which is that R also makes use of the home directory. You can find out what it is by using the `path.expand()` function, like this:

```
path.expand("~")
## [1] "/Users/dan"
```

You can change the user directory if you want, but we're not going to make use of it very much so there's no reason to. The only reason I'm even bothering to mention it at all is that when you use Rstudio to open a file, you'll see output on screen that defines the path to the file relative to the #~# directory. I'd prefer you not to be confused when you see it.[51]

2.14.3 the Windows paths use the wrong slash?

Let's suppose I'm on Windows. As before, I can find out what my current working directory is like this:

```
getwd()
## [1] "C:/Users/dan/
```

This seems about right, but you might be wondering why R is displaying a Windows path using the wrong type of slash. The answer is slightly complicated, and has to do with the fact that R treats the `\` character as "special" (see Section 7.8.7). If you're deeply wedded to the idea of specifying a path using the Windows style slashes, then what you need to do is to type `/` whenever you mean `\`. In other words, if you want to specify the working directory on a Windows computer, you need to use one of the following commands:

```
setwd( "C:/Users/dan" )
setwd( "C:\\Users\\dan" )
```

It's kind of annoying to have to do it this way, but as you'll see later on in Section 7.8.7 it's a necessary evil. Fortunately, as we'll see in the next section, Rstudio provides a much simpler way of changing directories…

2.14.4 Navigating the file system using the Rstudio file panel

Although I think it's important to understand how all this command line stuff works, in many (maybe even most) situations there's an easier way. For our purposes, the easiest way to navigate the file system is to make use of Rstudio's built in tools. The "file" panel – the lower right hand area in Figure 4.7 – is actually a pretty decent file browser. Not only can you just point and click on the names to move around the file system, you can also use it to set the working directory, and even load files.

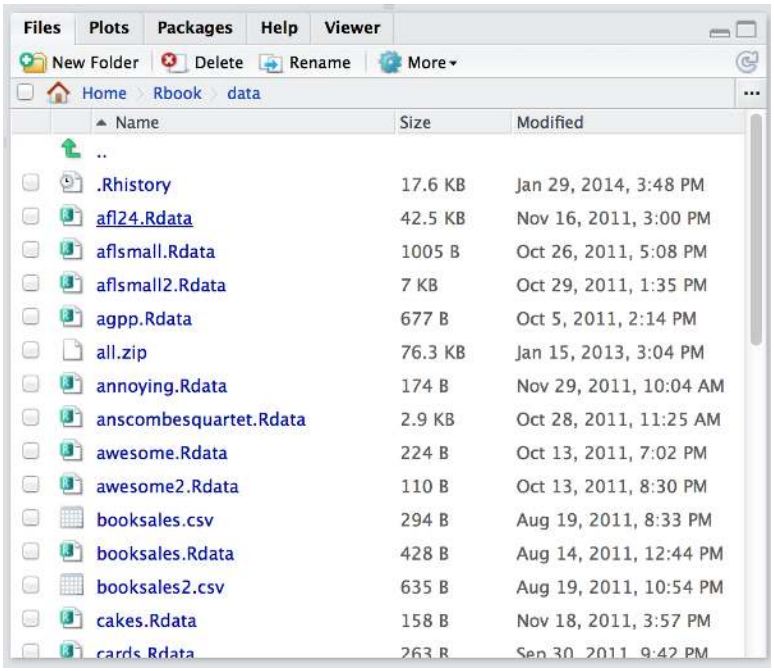


Figure 4.7: The "file panel" is the area shown in the lower right hand corner. It provides a very easy way to browse and navigate your computer using R. See main text for details.

Here's what you need to do to change the working directory using the file panel. Let's say I'm looking at the actual screen shown in Figure 4.7. At the top of the file panel you see some text that says "Home > Rbook > data". What that means is that it's *displaying* the files that are stored in the

```
/Users/dan/Rbook/data
```

directory on my computer. It does *not* mean that this is the R working directory. If you want to change the R working directory, using the file panel, you need to click on the button that reads "More". This will bring up a little menu, and one of the options will be "Set as Working Directory". If you select that option, then R really will change the working directory. You can tell that it has done so because this command appears in the console:

```
setwd("~/Rbook/data")
```

In other words, Rstudio sends a command to the R console, exactly as if you'd typed it yourself. The file panel can be used to do other things too. If you want to move "up" to the parent folder (e.g., from `/Users/dan/Rbook/data` to `/Users/dan/Rbook` click on the ".." link in the file panel. To move to a subfolder, click on the name of the folder that you want to open. You can open some types of file by clicking on them. You can delete files from your computer using the "delete" button, rename them with the "rename" button, and so on.

As you can tell, the file panel is a very handy little tool for navigating the file system. But it can do more than just navigate. As we'll see later, it can be used to open files. And if you look at the buttons and menu options that it presents, you can even use it to rename, delete, copy or move files, and create new folders. However, since most of that functionality isn't critical to the basic goals of this book, I'll let you discover those on your own.

This page titled 2.14: Navigating the File System is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **4.4: Navigating the File System** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

2.15: Loading and Saving Data

There are several different types of files that are likely to be relevant to us when doing data analysis. There are three in particular that are especially important from the perspective of this book:

- *Workspace files* are those with a .Rdata file extension. This is the standard kind of file that R uses to store data and variables. They're called "workspace files" because you can use them to save your whole workspace.
- *Comma separated value (CSV) files* are those with a .csv file extension. These are just regular old text files, and they can be opened with almost any software. It's quite typical for people to store data in CSV files, precisely because they're so simple.
- *Script files* are those with a .R file extension. These aren't data files at all; rather, they're used to save a collection of commands that you want R to execute later. They're just text files, but we won't make use of them until Chapter 8.

There are also several other types of file that R makes use of,[52] but they're not really all that central to our interests. There are also several other kinds of data file that you might want to import into R. For instance, you might want to open Microsoft Excel spreadsheets (.xlsx files), or data files that have been saved in the native file formats for other statistics software, such as SPSS, SAS, Minitab, Stata or Systat. Finally, you might have to handle databases. R tries hard to play nicely with other software, so it has tools that let you open and work with any of these and many others. I'll discuss some of these other possibilities elsewhere in this book (Section 7.9), but for now I want to focus primarily on the two kinds of data file that you're most likely to need: .Rdata files and .csv files. In this section I'll talk about how to load a workspace file, how to import data from a CSV file, and how to save your workspace to a workspace file. Throughout this section I'll first describe the (sometimes awkward) R commands that do all the work, and then I'll show you the (much easier) way to do it using Rstudio.

2.15.1 Loading workspace files using R

When I used the `list.files()` command to list the contents of the `/Users/dan/Rbook/data` directory (in Section 4.4.2), the output referred to a file called booksales.Rdata. Let's say I want to load the data from this file into my workspace. The way I do this is with the `load()` function. There are two arguments to this function, but the only one we're interested in is

- `file`. This should be a character string that specifies a path to the file that needs to be loaded. You can use an absolute path or a relative path to do so.

Using the absolute file path, the command would look like this:

```
load( file = "/Users/dan/Rbook/data/booksales.Rdata" )
```

but this is pretty lengthy. Given that the working directory (remember, we changed the directory at the end of Section 4.4.4) is `/Users/dan/Rbook/data`, I could use a relative file path, like so:

```
load( file = "../data/booksales.Rdata" )
```

However, my preference is usually to change the working directory first, and *then* load the file. What that would look like is this:

```
setwd( "../data" )         # move to the data directory
load( "booksales.Rdata" )  # load the data
```

If I were then to type `who()` I'd see that there are several new variables in my workspace now. Throughout this book, whenever you see me loading a file, I will assume that the file is actually stored in the working directory, or that you've changed the working directory so that R is pointing at the directory that contains the file. Obviously, *you* don't need type that command yourself: you can use the Rstudio file panel to do the work.

2.15.2 Loading workspace files using Rstudio

Okay, so how do we open an .Rdata file using the Rstudio file panel? It's terribly simple. First, use the file panel to find the folder that contains the file you want to load. If you look at Figure 4.7, you can see that there are several .Rdata files listed. Let's say I

want to load the `booksales.Rdata` file. All I have to do is click on the file name. Rstudio brings up a little dialog box asking me to confirm that I do want to load this file. I click yes. The following command then turns up in the console,

```
load("~/Rbook/data/booksales.Rdata")
```

and the new variables will appear in the workspace (you'll see them in the Environment panel in Rstudio, or if you type `who()`). So easy it barely warrants having its own section.

One quite commonly used data format is the humble "comma separated value" file, also called a CSV file, and usually bearing the file extension .csv. CSV files are just plain old-fashioned text files, and what they store is basically just a table of data. This is illustrated in Figure 4.8, which shows a file called booksales.csv that I've created. As you can see, each row corresponds to a variable, and each row represents the book sales data for one month. The first row doesn't contain actual data though: it has the names of the variables.

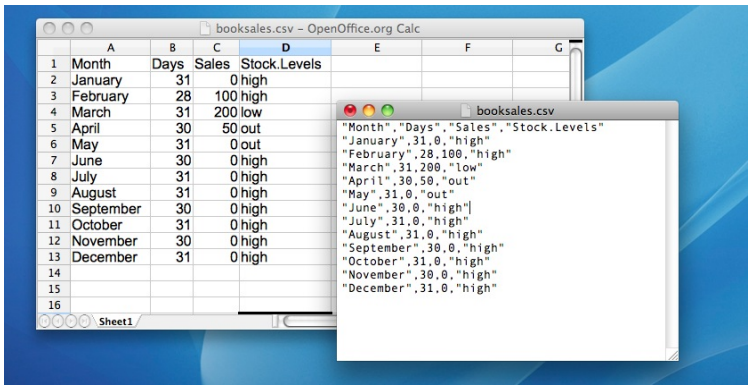


Figure 4.8: The booksales.csv data file. On the left, I've opened the file in using a spreadsheet program (OpenOffice), which shows that the file is basically a table. On the right, the same file is open in a standard text editor (the TextEdit program on a Mac), which shows how the file is formatted. The entries in the table are wrapped in quote marks and separated by commas.

If Rstudio were not available to you, the easiest way to open this file would be to use the `read.csv()` function.[53] This function is pretty flexible, and I'll talk a lot more about it's capabilities in Section 7.9 for more details, but for now there's only two arguments to the function that I'll mention:

- `file`. This should be a character string that specifies a path to the file that needs to be loaded. You can use an absolute path or a relative path to do so.
- `header`. This is a logical value indicating whether or not the first row of the file contains variable names. The default value is `TRUE`.

Therefore, to import the CSV file, the command I need is:

```
books <- read.csv( file = "booksales.csv" )
```

There are two very important points to notice here. Firstly, notice that I *didn't* try to use the `load()` function, because that function is only meant to be used for .Rdata files. If you try to use `load()` on other types of data, you get an error. Secondly, notice that when I imported the CSV file I assigned the result to a variable, which I imaginatively called `books`.[54] file. There's a reason for this. The idea behind an `.Rdata` file is that it stores a whole workspace. So, if you had the ability to look inside the file yourself you'd see that the data file keeps track of all the variables and their names. So when you `load()` the file, R restores all those original names. CSV files are treated differently: as far as R is concerned, the CSV only stores *one* variable, but that variable is big table. So when you import that table into the workspace, R expects *you* to give it a name.] Let's have a look at what we've got:

```
print( books )
```

```
##        Month Days Sales Stock.Levels
## 1    January   31     0         high
## 2   February   28   100         high
## 3      March   31   200          low
## 4      April   30    50          out
## 5        May   31     0          out
## 6       June   30     0         high
## 7       July   31     0         high
## 8     August   31     0         high
## 9  September   30     0         high
## 10   October   31     0         high
## 11  November   30     0         high
## 12  December   31     0         high
```

Clearly, it's worked, but the format of this output is a bit unfamiliar. We haven't seen anything like this before. What you're looking at is a *data frame*, which is a very important kind of variable in R, and one I'll discuss in Section 4.8. For now, let's just be happy that we imported the data and that it looks about right.

2.15.3 Importing data from CSV files using Rstudio

Yet again, it's easier in Rstudio. In the environment panel in Rstudio you should see a button called "Import Dataset". Click on that, and it will give you a couple of options: select the "From Text File…" option, and it will open up a very familiar dialog box asking you to select a file: if you're on a Mac, it'll look like the usual Finder window that you use to choose a file; on Windows it looks like an Explorer window. An example of what it looks like on a Mac is shown in Figure 4.9. I'm assuming that you're familiar with your own computer, so you should have no problem finding the CSV file that you want to import! Find the one you want, then click on the "Open" button. When you do this, you'll see a window that looks like the one in Figure 4.10.

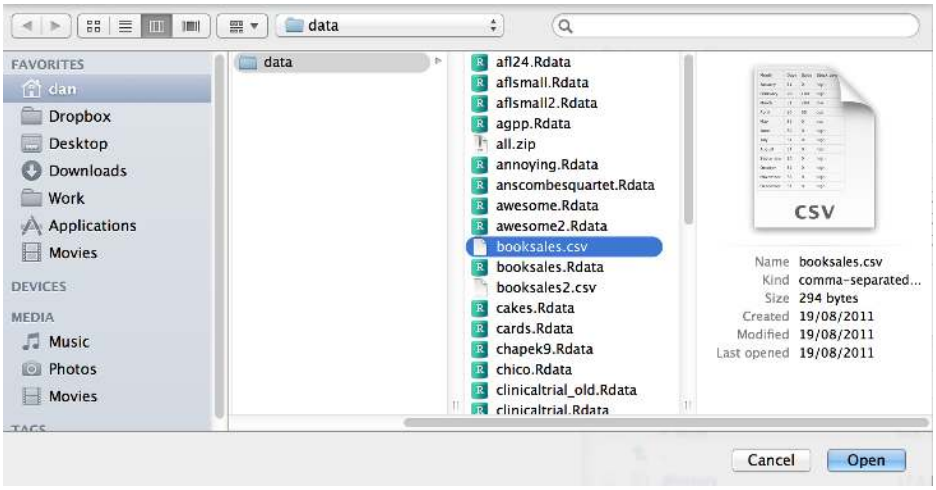


Figure 4.9: A dialog box on a Mac asking you to select the CSV file R should try to import. Mac users will recognise this immediately: it's the usual way in which a Mac asks you to find a file. Windows users won't see this: they'll see the usual explorer window that Windows always gives you when it wants you to select a file.

The import data set window is relatively straightforward to understand.

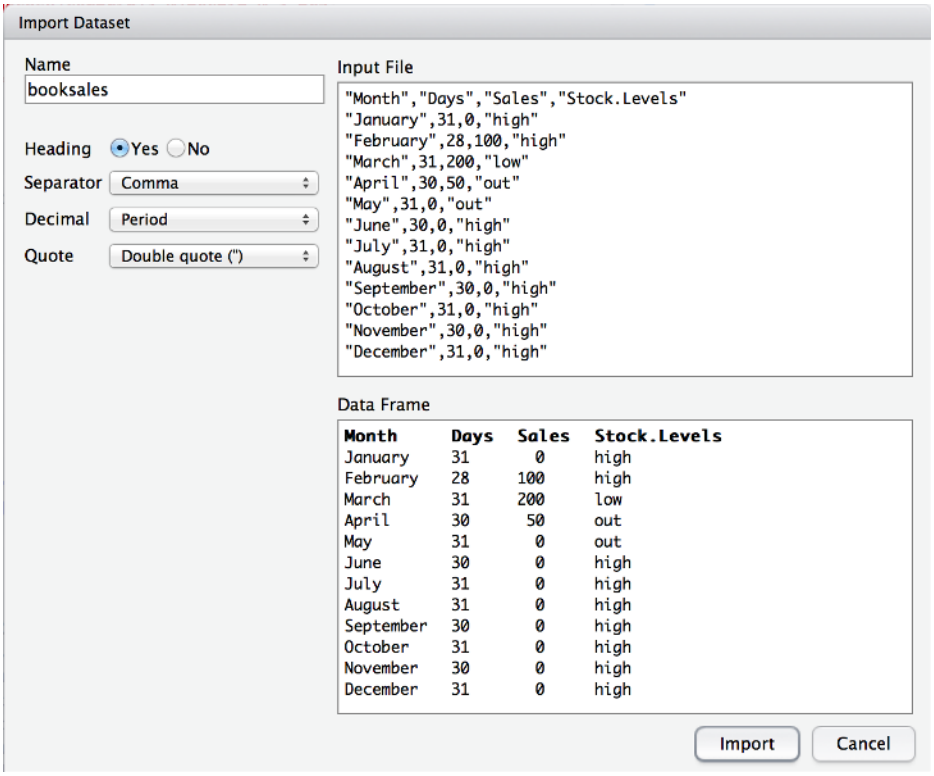


Figure 4.10: The Rstudio window for importing a CSV file into R

In the top left corner, you need to type the name of the variable you R to create. By default, that will be the same as the file name: our file is called `booksales.csv`, so Rstudio suggests the name `booksales`. If you're happy with that, leave it alone. If not, type something else. Immediately below this are a few things that you can tweak to make sure that the data gets imported correctly:

- Heading. Does the first row of the file contain raw data, or does it contain headings for each variable? The `booksales.csv` file has a header at the top, so I selected "yes".
- Separator. What character is used to separate different entries? In most CSV files this will be a comma (it is "comma separated" after all). But you can change this if your file is different.
- Decimal. What character is used to specify the decimal point? In English speaking countries, this is almost always a period (i.e., `.`). That's not universally true: many European countries use a comma. So you can change that if you need to.
- Quote. What character is used to denote a block of text? That's usually going to be a double quote mark. It is for the `booksales.csv` file, so that's what I selected.

The nice thing about the Rstudio window is that it shows you the raw data file at the top of the window, and it shows you a preview of the data at the bottom. If the data at the bottom doesn't look right, try changing some of the settings on the left hand side. Once you're happy, click "Import". When you do, two commands appear in the R console:

```
booksales <- read.csv("~/Rbook/data/booksales.csv")
View(booksales)
```

The first of these commands is the one that loads the data. The second one will display a pretty table showing the data in Rstudio.

2.15.4 Saving a workspace file using `save`

Not surprisingly, saving data is very similar to loading data. Although Rstudio provides a simple way to save files (see below), it's worth understanding the actual commands involved. There are two commands you can use to do this, `save()` and `save.image()`. If you're happy to save *all* of the variables in your workspace into the data file, then you should use `save.image()`. And if you're happy for R to save the file into the current working directory, all you have to do is this:

```
save.image( file = "myfile.Rdata" )
```

Since `file` is the first argument, you can shorten this to `save.image("myfile.Rdata")` ; and if you want to save to a different directory, then (as always) you need to be more explicit about specifying the path to the file, just as we discussed in Section 4.4. Suppose, however, I have several variables in my workspace, and I only want to save some of them. For instance, I might have this as my workspace:

```
who()
##   -- Name --    -- Class --    -- Size --
##   data          data.frame     3 x 2
##   handy         character      1
##   junk          numeric        1
```

I want to save `data` and `handy` , but not `junk` . But I don't want to delete `junk` right now, because I want to use it for something else later on. This is where the `save()` function is useful, since it lets me indicate exactly which variables I want to save. Here is one way I can use the `save` function to solve my problem:

```
save(data, handy, file = "myfile.Rdata")
```

Importantly, you *must* specify the name of the `file` argument. The reason is that if you don't do so, R will think that `"myfile.Rdata"` is actually a *variable* that you want to save, and you'll get an error message. Finally, I should mention a second way to specify which variables the `save()` function should save, which is to use the `list` argument. You do so like this:

```
save.me <- c("data", "handy")   # the variables to be saved
save( file = "booksales2.Rdata", list = save.me )   # the command to save them
```

2.15.5 Saving a workspace file using Rstudio

Rstudio allows you to save the workspace pretty easily. In the environment panel (Figures 4.5 and 4.6) you can see the "save" button. There's no text, but it's the same icon that gets used on every computer everywhere: it's the one that looks like a floppy disk. You know, those things that haven't been used in about 20 years. Alternatively, go to the "Session" menu and click on the "Save Workspace As…" option.[55] This will bring up the standard "save" dialog box for your operating system (e.g., on a Mac it'll look a little bit like the loading dialog box in Figure 4.9). Type in the name of the file that you want to save it to, and all the variables in your workspace will be saved to disk. You'll see an R command like this one

```
save.image("~/Desktop/Untitled.RData")
```

Pretty straightforward, really.

2.15.6 Other things you might want to save

Until now, we've talked mostly about loading and saving *data*. Other things you might want to save include:

- *The output*. Sometimes you might also want to keep a copy of all your interactions with R, including everything that you typed in and everything that R did in response. There are some functions that you can use to get R to write its output to a file rather than to print onscreen (e.g., `sink()`), but to be honest, if you do want to save the R output, the easiest thing to do is to use the mouse to select the relevant text in the R console, go to the "Edit" menu in Rstudio and select "Copy". The output has now been copied to the clipboard. Now open up your favourite text editor or word processing software, and paste it. And you're done. However, this will only save the contents of the console, not the plots you've drawn (assuming you've drawn some). We'll talk about saving images later on.

- *A script*. While it is possible – and sometimes handy – to save the R output as a method for keeping a copy of your statistical analyses, another option that people use a lot (especially when you move beyond simple "toy" analyses) is to write *scripts*. A script is a text file in which you write out all the commands that you want R to run. You can write your script using whatever software you like. In real world data analysis writing scripts is a key skill – and as you become familiar with R you'll probably find that most of what you do involves scripting rather than typing commands at the R prompt. However, you won't need to do much scripting initially, so we'll leave that until Chapter 8.

This page titled 2.15: Loading and Saving Data is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **4.5: Loading and Saving Data** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

2.16: Useful Things to Know about Variables

In Chapter 3 I talked a lot about variables, how they're assigned and some of the things you can do with them, but there's a lot of additional complexities. That's not a surprise of course. However, some of those issues are worth drawing your attention to now. So that's the goal of this section; to cover a few extra topics. As a consequence, this section is basically a bunch of things that I want to briefly mention, but don't really fit in anywhere else. In short, I'll talk about several different issues in this section, which are only loosely connected to one another.

2.16.1 Special values

The first thing I want to mention are some of the "special" values that you might see R produce. Most likely you'll see them in situations where you were expecting a number, but there are quite a few other ways you can encounter them. These values are `Inf`, `NaN`, `NA` and `NULL`. These values can crop up in various different places, and so it's important to understand what they mean.

- *Infinity* (`Inf`). The easiest of the special values to explain is `Inf`, since it corresponds to a value that is infinitely large. You can also have `-Inf`. The easiest way to get `Inf` is to divide a positive number by 0:

```
1 / 0
```

```
## [1] Inf
```

In most real world data analysis situations, if you're ending up with infinite numbers in your data, then something has gone awry. Hopefully you'll never have to see them.

- *Not a Number* (`NaN`). The special value of `NaN` is short for "not a number", and it's basically a reserved keyword that means "there isn't a mathematically defined number for this". If you can remember your high school maths, remember that it is conventional to say that 0/0 doesn't have a proper answer: mathematicians would say that 0/0 is *undefined*. R says that it's not a number:

```
 0 / 0
```

```
## [1] NaN
```

Nevertheless, it's still treated as a "numeric" value. To oversimplify, `NaN` corresponds to cases where you asked a proper numerical question that genuinely has *no meaningful answer*.

- *Not available* (`NA`). `NA` indicates that the value that is "supposed" to be stored here is missing. To understand what this means, it helps to recognise that the `NA` value is something that you're most likely to see when analysing data from real world experiments. Sometimes you get equipment failures, or you lose some of the data, or whatever. The point is that some of the information that you were "expecting" to get from your study is just plain missing. Note the difference between `NA` and `NaN`. For `NaN`, we really do know what's supposed to be stored; it's just that it happens to correspond to something like 0/0 that doesn't make any sense at all. In contrast, `NA` indicates that we actually don't know what was supposed to be there. The information is *missing*.
- *No value* (`NULL`). The `NULL` value takes this "absence" concept even further. It basically asserts that the variable genuinely has no value whatsoever. This is quite different to both `NaN` and `NA`. For `NaN` we actually know what the value is, because it's something insane like 0/0. For `NA`, we believe that there is supposed to be a value "out there", but a dog ate our homework and so we don't quite know what it is. But for `NULL` we strongly believe that there is *no value at all*.

2.16.2 Assigning names to vector elements

One thing that is sometimes a little unsatisfying about the way that R prints out a vector is that the elements come out unlabelled. Here's what I mean. Suppose I've got data reporting the quarterly profits for some company. If I just create a no-frills vector, I have

to rely on memory to know which element corresponds to which event. That is:

```
profit <- c( 3.1, 0.1, -1.4, 1.1 )
profit
```

```
## [1]  3.1  0.1 -1.4  1.1
```

You can probably guess that the first element corresponds to the first quarter, the second element to the second quarter, and so on, but that's only because I've told you the back story and because this happens to be a very simple example. In general, it can be quite difficult. This is where it can be helpful to assign `names` to each of the elements. Here's how you do it:

```
names(profit) <- c("Q1","Q2","Q3","Q4")
profit
```

```
##   Q1   Q2   Q3   Q4
##  3.1  0.1 -1.4  1.1
```

This is a slightly odd looking command, admittedly, but it's not too difficult to follow. All we're doing is assigning a vector of labels (character strings) to `names(profit)`. You can always delete the names again by using the command `names(profit) <- NULL`. It's also worth noting that you don't have to do this as a two stage process. You can get the same result with this command:

```
profit <- c( "Q1" = 3.1, "Q2" = 0.1, "Q3" = -1.4, "Q4" = 1.1 )
profit
```

```
##   Q1   Q2   Q3   Q4
##  3.1  0.1 -1.4  1.1
```

The important things to notice are that (a) this does make things much easier to read, but (b) the names at the top aren't the "real" data. The *value* of `profit[1]` is still `3.1`; all I've done is added a *name* to `profit[1]` as well. Nevertheless, names aren't purely cosmetic, since R allows you to pull out particular elements of the vector by referring to their names:

```
profit["Q1"]
```

```
##  Q1
## 3.1
```

And if I ever need to pull out the names themselves, then I just type `names(profit)`.

2.16.3 Variable classes

As we've seen, R allows you to store different kinds of data. In particular, the variables we've defined so far have either been character data (text), numeric data, or logical data.[56] It's important that we remember what kind of information each variable stores (and even more important that R remembers) since different kinds of variables allow you to do different things to them. For instance, if your variables have numerical information in them, then it's okay to multiply them together:

```
x <- 5   # x is numeric
y <- 4   # y is numeric
x * y
```

```
## [1] 20
```

But if they contain character data, multiplication makes no sense whatsoever, and R will complain if you try to do it:

```
x <- "apples"   # x is character
y <- "oranges"  # y is character
x * y
```

```
## Error in x * y: non-numeric argument to binary operator
```

Even R is smart enough to know you can't multiply `"apples"` by `"oranges"`. It knows this because the quote marks are indicators that the variable is supposed to be treated as text, not as a number.

This is quite useful, but notice that it means that R makes a big distinction between `5` and `"5"`. Without quote marks, R treats `5` as the number five, and will allow you to do calculations with it. With the quote marks, R treats `"5"` as the textual character five, and doesn't recognise it as a number any more than it recognises `"p"` or `"five"` as numbers. As a consequence, there's a big difference between typing `x <- 5` and typing `x <- "5"`. In the former, we're storing the number `5`; in the latter, we're storing the character `"5"`. Thus, if we try to do multiplication with the character versions, R gets stroppy:

```
x <- "5"   # x is character
y <- "4"   # y is character
x * y
```

```
## Error in x * y: non-numeric argument to binary operator
```

Okay, let's suppose that I've forgotten what kind of data I stored in the variable `x` (which happens depressingly often). R provides a function that will let us find out. Or, more precisely, it provides *three* functions: `class()`, `mode()` and `typeof()`. Why the heck does it provide three functions, you might be wondering? Basically, because R actually keeps track of three different kinds of information about a variable:

1. The ***class*** of a variable is a "high level" classification, and it captures psychologically (or statistically) meaningful distinctions. For instance `"2011-09-12"` and `"my birthday"` are both text strings, but there's an important difference between the two: one of them is a date. So it would be nice if we could get R to recognise that `"2011-09-12"` is a date, and allow us to do things like add or subtract from it. The class of a variable is what R uses to keep track of things like that. Because the class of a variable is critical for determining what R can or can't do with it, the `class()` function is very handy.
2. The ***mode*** of a variable refers to the format of the information that the variable stores. It tells you whether R has stored text data or numeric data, for instance, which is kind of useful, but it only makes these "simple" distinctions. It can be useful to know about, but it's not the main thing we care about. So I'm not going to use the `mode()` function very much.[57]
3. The ***type*** of a variable is a very low level classification. We won't use it in this book, but (for those of you that care about these details) this is where you can see the distinction between integer data, double precision numeric, etc. Almost none of you actually will care about this, so I'm not even going to bother demonstrating the `typeof()` function.

For purposes, it's the `class()` of the variable that we care most about. Later on, I'll talk a bit about how you can convince R to "coerce" a variable to change from one class to another (Section 7.10). That's a useful skill for real world data analysis, but it's not something that we need right now. In the meantime, the following examples illustrate the use of the `class()` function:

```
x <- "hello world"     # x is text
class(x)
```

```
## [1] "character"
```

```
x <- TRUE     # x is logical
class(x)
```

```
## [1] "logical"
```

```
x <- 100     # x is a number
class(x)
```

```
## [1] "numeric"
```

Exciting, no?

This page titled 2.16: Useful Things to Know about Variables is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **4.6: Useful Things to Know about Variables** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

2.17: Factors

Okay, it's time to start introducing some of the data types that are somewhat more specific to statistics. If you remember back to Chapter 2, when we assign numbers to possible outcomes, these numbers can mean quite different things depending on what kind of variable we are attempting to measure. In particular, we commonly make the distinction between *nominal*, *ordinal*, *interval* and *ratio* scale data. How do we capture this distinction in R? Currently, we only seem to have a single numeric data type. That's probably not going to be enough, is it?

A little thought suggests that the numeric variable class in R is perfectly suited for capturing ratio scale data. For instance, if I were to measure response time (RT) for five different events, I could store the data in R like this:

```
RT <- c(342, 401, 590, 391, 554)
```

where the data here are measured in milliseconds, as is conventional in the psychological literature. It's perfectly sensible to talk about "twice the response time", 2×RT, or the "response time plus 1 second", RT+1000, and so both of the following are perfectly reasonable things for R to do:

```
2 * RT
```

```
## [1]  684  802 1180  782 1108
```

```
RT + 1000
```

```
## [1] 1342 1401 1590 1391 1554
```

And to a lesser extent, the "numeric" class is okay for interval scale data, as long as we remember that multiplication and division aren't terribly interesting for these sorts of variables. That is, if my IQ score is 110 and yours is 120, it's perfectly okay to say that you're 10 IQ points smarter than me[58], but it's not okay to say that I'm only 92% as smart as you are, because intelligence doesn't have a natural zero.[59] We might even be willing to tolerate the use of numeric variables to represent ordinal scale variables, such as those that you typically get when you ask people to rank order items (e.g., like we do in Australian elections), though as we will see R actually has a built in tool for representing ordinal data (see Section 7.11.2) However, when it comes to nominal scale data, it becomes completely unacceptable, because almost all of the "usual" rules for what you're allowed to do with numbers don't apply to nominal scale data. It is for this reason that R has ***factors***.

2.17.1 Introducing factors

Suppose, I was doing a study in which people could belong to one of three different treatment conditions. Each group of people were asked to complete the same task, but each group received different instructions. Not surprisingly, I might want to have a variable that keeps track of what group people were in. So I could type in something like this

```
group <- c(1,1,1,2,2,2,3,3,3)
```

so that `group[i]` contains the group membership of the `i` -th person in my study. Clearly, this is numeric data, but equally obviously this is a nominal scale variable. There's no sense in which "group 1" plus "group 2" equals "group 3", but nevertheless if I try to do that, R won't stop me because it doesn't know any better:

```
group + 2
```

```
## [1] 3 3 3 4 4 4 5 5 5
```

Apparently R seems to think that it's allowed to invent "group 4" and "group 5", even though they didn't actually exist. Unfortunately, R is too stupid to know any better: it thinks that `3` is an ordinary number in this context, so it sees no problem in calculating `3 + 2`. But since *we're* not that stupid, we'd like to stop R from doing this. We can do so by instructing R to treat `group` as a factor. This is easy to do using the `as.factor()` function.[60]

```
group <- as.factor(group)
group
```

```
## [1] 1 1 1 2 2 2 3 3 3
## Levels: 1 2 3
```

It looks more or less the same as before (though it's not immediately obvious what all that `Levels` rubbish is about), but if we ask R to tell us what the class of the `group` variable is now, it's clear that it has done what we asked:

```
class(group)
```

```
## [1] "factor"
```

Neat. Better yet, now that I've converted `group` to a factor, look what happens when I try to add 2 to it:

```
group + 2
```

```
## Warning in Ops.factor(group, 2): '+' not meaningful for factors
```

```
## [1] NA NA NA NA NA NA NA NA NA
```

This time even R is smart enough to know that I'm being an idiot, so it tells me off and then produces a vector of missing values. (i.e., `NA`: see Section 4.6.1).

2.17.2 Labelling the factor levels

I have a confession to make. My memory is not infinite in capacity; and it seems to be getting worse as I get older. So it kind of annoys me when I get data sets where there's a nominal scale variable called `gender`, with two levels corresponding to males and females. But when I go to print out the variable I get something like this:

```
gender
```

```
## [1] 1 1 1 1 1 2 2 2 2
## Levels: 1 2
```

Okaaaay. That's not helpful at all, and it makes me very sad. Which number corresponds to the males and which one corresponds to the females? Wouldn't it be nice if R could actually keep track of this? It's way too hard to remember which number corresponds to which gender. And besides, the problem that this causes is much more serious than a single sad nerd… because R has no way of knowing that the `1`s in the `group` variable are a very different kind of thing to the `1`s in the `gender` variable. So if I try to ask which elements of the `group` variable are equal to the corresponding elements in `gender`, R thinks this is totally kosher, and gives me this:

```
group == gender
```

```
## Error in Ops.factor(group, gender): level sets of factors are different
```

Well, that's … especially stupid.[61] The problem here is that R is very literal minded. Even though you've declared both `group` and `gender` to be factors, it still assumes that a `1` is a `1` no matter which variable it appears in.

To fix both of these problems (my memory problem, and R's infuriating literal interpretations), what we need to do is assign meaningful labels to the different *levels* of each factor. We can do that like this:

```
levels(group) <- c("group 1", "group 2", "group 3")
print(group)
```

```
## [1] group 1 group 1 group 1 group 2 group 2 group 2 group 3 group 3 group 3
## Levels: group 1 group 2 group 3
```

```
levels(gender) <- c("male", "female")
print(gender)
```

```
## [1] male   male   male   male   male   female female female female
## Levels: male female
```

That's much easier on the eye, and better yet, R is smart enough to know that `"female"` is not equal to `"group 2"`, so now when I try to ask which group memberships are "equal to" the gender of the corresponding person,

```
group == gender
```

```
## Error in Ops.factor(group, gender): level sets of factors are different
```

R correctly tells me that I'm an idiot.

2.17.3 Moving on…

Factors are very useful things, and we'll use them a lot in this book: they're *the* main way to represent a nominal scale variable. And there are lots of nominal scale variables out there. I'll talk more about factors in 7.11.2, but for now you know enough to be able to get started.

This page titled 2.17: Factors is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **4.7: Factors** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

2.18: Data frames

It's now time to go back and deal with the somewhat confusing thing that happened in Section **??** when we tried to open up a CSV file. Apparently we succeeded in loading the data, but it came to us in a very odd looking format. At the time, I told you that this was a ***data frame***. Now I'd better explain what that means.

2.18.1 Introducing data frames

In order to understand why R has created this funny thing called a data frame, it helps to try to see what problem it solves. So let's go back to the little scenario that I used when introducing factors in Section 4.7. In that section I recorded the `group` and `gender` for all 9 participants in my study. Let's also suppose I recorded their ages and their `score` on "Dan's Terribly Exciting Psychological Test":

```
age <- c(17, 19, 21, 37, 18, 19, 47, 18, 19)
score <- c(12, 10, 11, 15, 16, 14, 25, 21, 29)
```

Assuming no other variables are in the workspace, if I type `who()` I get this:

```
who()
```

```
##    -- Name --   -- Class --   -- Size --
##    age          numeric      9
##    gender       factor       9
##    group        factor       9
##    score        numeric      9
```

So there are four variables in the workspace, `age`, `gender`, `group` and `score`. And it just so happens that all four of them are the same size (i.e., they're all vectors with 9 elements). Aaaand it just so happens that `age[1]` corresponds to the age of the first person, and `gender[1]` is the gender of that very same person, etc. In other words, you and I both know that all four of these variables correspond to the *same* data set, and all four of them are organised in exactly the same way.

However, R *doesn't* know this! As far as it's concerned, there's no reason why the `age` variable has to be the same length as the `gender` variable; and there's no particular reason to think that `age[1]` has any special relationship to `gender[1]` any more than it has a special relationship to `gender[4]`. In other words, when we store everything in separate variables like this, R doesn't know anything about the relationships between things. It doesn't even really know that these variables actually refer to a proper data set. The data frame fixes this: if we store our variables inside a data frame, we're telling R to treat these variables as a single, fairly coherent data set.

To see how they do this, let's create one. So how do we create a data frame? One way we've already seen: if we import our data from a CSV file, R will store it as a data frame. A second way is to create it directly from some existing variables using the `data.frame()` function. All you have to do is type a list of variables that you want to include in the data frame. The output of a `data.frame()` command is, well, a data frame. So, if I want to store all four variables from my experiment in a data frame called `expt` I can do so like this:

```
expt <- data.frame ( age, gender, group, score )
expt
```

```
##   age gender   group score
## 1  17   male group 1    12
## 2  19   male group 1    10
## 3  21   male group 1    11
## 4  37   male group 2    15
## 5  18   male group 2    16
## 6  19 female group 2    14
## 7  47 female group 3    25
## 8  18 female group 3    21
## 9  19 female group 3    29
```

Note that `expt` is a completely self-contained variable. Once you've created it, it no longer depends on the original variables from which it was constructed. That is, if we make changes to the original `age` variable, it will *not* lead to any changes to the age data stored in `expt` .

At this point, our workspace contains only the one variable, a data frame called `expt` . But as we can see when we told R to print the variable out, this data frame contains 4 variables, each of which has 9 observations. So how do we get this information out again? After all, there's no point in storing information if you don't use it, and there's no way to use information if you can't access it. So let's talk a bit about how to pull information out of a data frame.

The first thing we might want to do is pull out one of our stored variables, let's say `score` . One thing you might try to do is ignore the fact that `score` is locked up inside the `expt` data frame. For instance, you might try to print it out like this:

```
score
```

```
## Error in eval(expr, envir, enclos): object 'score' not found
```

This doesn't work, because R doesn't go "peeking" inside the data frame unless you explicitly tell it to do so. There's actually a very good reason for this, which I'll explain in a moment, but for now let's just assume R knows what it's doing. How do we tell R to look inside the data frame? As is always the case with R there are several ways. The simplest way is to use the `$` operator to extract the variable you're interested in, like this:

```
expt$score
```

```
## [1] 12 10 11 15 16 14 25 21 29
```

2.18.2 Getting information about a data frame

One problem that sometimes comes up in practice is that you forget what you called all your variables. Normally you might try to type `objects()` or `who()` , but neither of those commands will tell you what the names are for those variables inside a data frame! One way is to ask R to tell you what the *names* of all the variables stored in the data frame are, which you can do using the `names()` function:

```
names(expt)
```

```
## [1] "age"    "gender" "group"  "score"
```

An alternative method is to use the `who()` function, as long as you tell it to look at the variables inside data frames. If you set `expand = TRUE` then it will not only list the variables in the workspace, but it will "expand" any data frames that you've got in the workspace, so that you can see what they look like. That is:

```
who(expand = TRUE)
```

```
##    -- Name --   -- Class --   -- Size --
##    expt         data.frame    9 x 4
##     $age        numeric       9
##     $gender     factor        9
##     $group      factor        9
##     $score      numeric       9
```

or, since `expand` is the first argument in the `who()` function you can just type `who(TRUE)` . I'll do that a lot in this book.

2.18.3 Looking for more on data frames?

There's a lot more that can be said about data frames: they're fairly complicated beasts, and the longer you use R the more important it is to make sure you really understand them. We'll talk a lot more about them in Chapter 7.

This page titled 2.18: Data frames is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **4.8: Data frames** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

2.19: Suggested Readings and Videos

There are many online resources for learning R. Here are a few:

- Datacamp: Offers free online courses for many aspects of R programming
- A Student's Guide to R
- R for cats: A humorous introduction to R programming
- aRrgh: a newcomer's (angry) guide to R
- Quick-R
- RStudio Cheat Sheets: Quick references for many different aspects of R programming
- tidverse Style Guide: Make your code beautiful and reader-friendly!
- R for Data Science: This free online book focuses on working with data in R.
- Advanced R: This free online book by Hadley Wickham will help you get to the next level once your R skills start to develop.
- R intro for Python users: Used Python before? Check this out for a guide on how to transition to R.

This page titled 2.19: Suggested Readings and Videos is shared under a CC BY-NC 2.0 license and was authored, remixed, and/or curated by Russell A. Poldrack via source content that was edited to the style and standards of the LibreTexts platform.

- **3.11: Suggested Readings and Videos** by Russell A. Poldrack is licensed CC BY-NC 4.0. Original source: https://statsthinking21.github.io/statsthinking21-core-site.

CHAPTER OVERVIEW

3: Summarizing Data Visually

3.1: Qualitative Data

3.2: Quantitative Data

3.3: Other Graphical Representations of Data

3.4: Statistical Literacy

3: Summarizing Data Visually is shared under a not declared license and was authored, remixed, and/or curated by LibreTexts.

3.1: Qualitative Data

Remember, qualitative data are words describing a characteristic of the individual. There are several different graphs that are used for qualitative data. These graphs include bar graphs, Pareto charts, and pie charts.

Pie charts and bar graphs are the most common ways of displaying qualitative data. A spreadsheet program like Excel can make both of them. The first step for either graph is to make a **frequency or relative frequency table**. A frequency table is a summary of the data with counts of how often a data value (or category) occurs.

Example 3.1.1

Suppose you have the following data for which type of car students at a college drive?

Ford, Chevy, Honda, Toyota, Toyota, Nissan, Kia, Nissan, Chevy, Toyota, Honda, Chevy, Toyota, Nissan, Ford, Toyota, Nissan, Mercedes, Chevy, Ford, Nissan, Toyota, Nissan, Ford, Chevy, Toyota, Nissan, Honda, Porsche, Hyundai, Chevy, Chevy, Honda, Toyota, Chevy, Ford, Nissan, Toyota, Chevy, Honda, Chevy, Saturn, Toyota, Chevy, Chevy, Nissan, Honda, Toyota, Toyota, Nissan

Solution

A listing of data is too hard to look at and analyze, so you need to summarize it. First you need to decide the categories. In this case it is relatively easy; just use the car type. However, there are several cars that only have one car in the list. In that case it is easier to make a category called other for the ones with low values. Now just count how many of each type of cars there are. For example, there are 5 Fords, 12 Chevys, and 6 Hondas. This can be put in a frequency distribution:

Table **3.1.1**: Frequency Table for Type of Car Data

Cateogry	Frequency
Ford	5
Chevy	12
Honda	6
Toyota	12
Nissan	10
Other	5
Total	50

The total of the frequency column should be the number of observations in the data.

Since raw numbers are not as useful to tell other people it is better to create a third column that gives the relative frequency of each category. This is just the frequency divided by the total. As an example for Ford category:

$$\text{relative frequency} = \frac{5}{50} = 0.10$$

This can be written as a decimal, fraction, or percent. You now have a relative frequency distribution:

Table **3.1.2**: Relative Frequency Table for Type of Car Data

Category	Frequency	Relative Frequency
Ford	5	0.10
Chevy	12	0.24
Honda	6	0.12
Toyota	12	0.24
Nissan	10	0.20
Other	5	0.10
Total	50	1.00

The relative frequency column should add up to 1.00. It might be off a little due to rounding errors.

Now that you have the frequency and relative frequency table, it would be good to display this data using a graph. There are several different types of graphs that can be used: bar chart, pie chart, and Pareto charts.

Bar graphs or charts consist of the frequencies on one axis and the categories on the other axis. Then you draw rectangles for each category with a height (if frequency is on the vertical axis) or length (if frequency is on the horizontal axis) that is equal to the frequency. All of the rectangles should be the same width, and there should be equally width gaps between each bar.

Example 3.1.2 drawing a bar graph

Draw a bar graph of the data in Example **3.1.1**.

Solution

Table **3.1.2**: Relative Frequency Table for Type of Car Data

Category	Frequency	Relative Frequency
Ford	5	0.10
Chevy	12	0.24
Honda	6	0.12
Toyota	12	0.24
Nissan	10	0.20
Other	5	0.10
Total	50	1.00

Put the frequency on the vertical axis and the category on the horizontal axis.

Then just draw a box above each category whose height is the frequency.

All graphs are drawn using *R*. The command in *R* to create a bar graph is:

> variable<-c(type in percentages or frequencies for each class with commas in between values)
>
> barplot(variable,names.arg=c("type in name of 1st category", "type in name of 2nd category",…,"type in name of last category"),
>
> ylim=c(0,number over max), xlab="type in label for x-axis", ylab="type in label for y-axis",ylim=c(0,number above maximum y value), main="type in title", col="type in a color") – creates a bar graph of the data in a color if you want.

For this example the command would be:

car<-c(5, 12, 6, 12, 10, 5)

barplot(car, names.arg=c("Ford", "Chevy", "Honda", "Toyota", "Nissan", "Other"), xlab="Type of Car", ylab="Frequency", ylim=c(0,12), main="Type of Car Driven by College Students", col="blue")

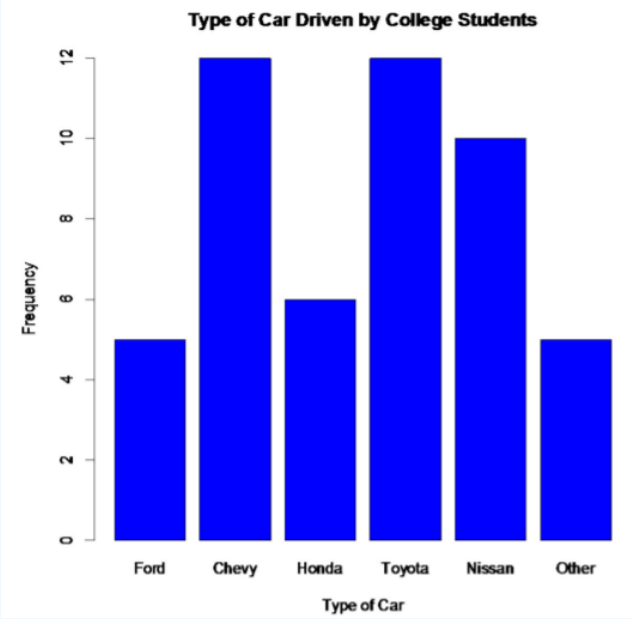


Figure for Type of Car Data

Notice from the graph, you can see that Toyota and Chevy are the more popular car, with Nissan not far behind. Ford seems to be the type of car that you can tell was the least liked, though the cars in the other category would be liked less than a Ford.

Some key features of a bar graph:

- Equal spacing on each axis.
- Bars are the same width.
- There should be labels on each axis and a title for the graph.
- There should be a scaling on the frequency axis and the categories should be listed on the category axis.
- The bars don't touch.

You can also draw a bar graph using relative frequency on the vertical axis. This is useful when you want to compare two samples with different sample sizes. The relative frequency graph and the frequency graph should look the same, except for the scaling on the frequency axis.

Using R, the command would be:

car<-c(0.1, 0.24, 0.12, 0.24, 0.2, 0.1)

barplot(car, names.arg=c("Ford", "Chevy", "Honda", "Toyota", "Nissan", "Other"), xlab="Type of Car", ylab="Relative Frequency", main="Type of Car Driven by College Students", col="blue", ylim=c(0,.25))

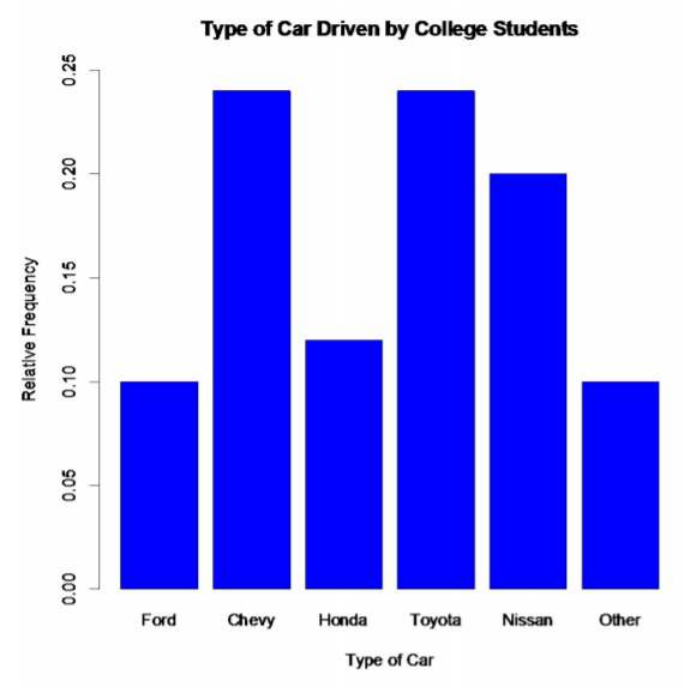


Figure for Type of Car Data

Another type of graph for qualitative data is a pie chart. A pie chart is where you have a circle and you divide pieces of the circle into pie shapes that are proportional to the size of the relative frequency. There are 360 degrees in a full circle. Relative frequency is just the percentage as a decimal. All you have to do to find the angle by multiplying the relative frequency by 360 degrees. Remember that 180 degrees is half a circle and 90 degrees is a quarter of a circle

Example 3.1.3 drawing a pie chart

Draw a pie chart of the data in Example 3.1.1.

First you need the relative frequencies.

Table 3.1.2: Relative Frequency Table for Type of Car Data

Category	Frequency	Relative Frequency
Ford	5	0.10
Chevy	12	0.24
Honda	6	0.12
Toyota	12	0.24
Nissan	10	0.20
Other	5	0.10
Total	50	1.00

Solution

Then you multiply each relative frequency by 360° to obtain the angle measure for each category.

Table 3.1.3: Pie Chart Angles for Type of Car Data

Category	Relative Frequency	Angle (in degrees (°))
Ford	0.10	36.0
Chevy	0.24	86.4
Honda	0.12	43.2
Toyota	0.24	86.4
Nissan	0.20	72.0
Other	0.10	36.0
Total	1.00	360.0

Now draw the pie chart using a compass, protractor, and straight edge. Technology is preferred. If you use technology, there is no need for the relative frequencies or the angles.

You can use R to graph the pie chart. In R, the commands would be:

pie(variable,labels=c("type in name of 1st category", "type in name of 2nd category",…,"type in name of last category"),main="type in title", col=rainbow(number of categories)) – creates a pie chart with a title and rainbow of colors for each category.

For this example, the commands would be:

car<-c(5, 12, 6, 12, 10, 5)

pie(car, labels=c("Ford, 10%", "Chevy, 24%", "Honda, 12%", "Toyota, 24%", "Nissan, 20%", "Other, 10%"), main="Type of Car Driven by College Students", col=rainbow(6))

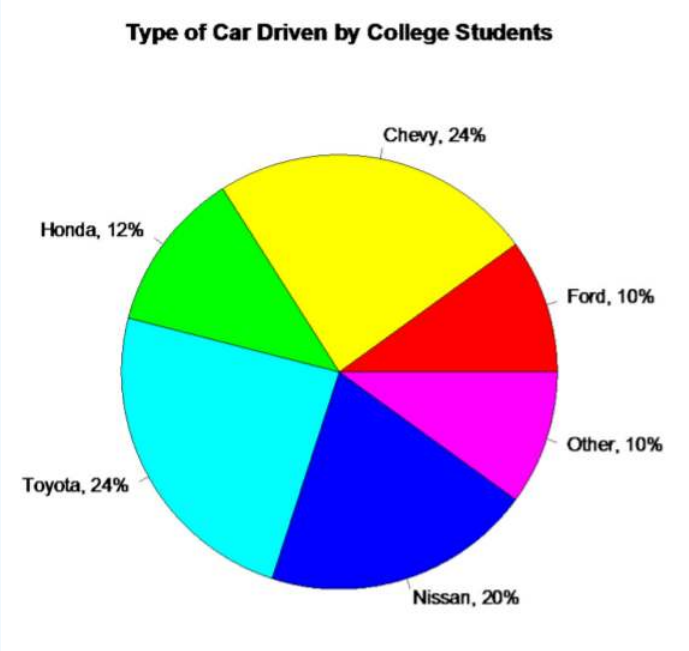


Figure **3.1.3**: Pie Chart for Type of Car Data

As you can see from the graph, Toyota and Chevy are more popular, while the cars in the other category are liked the least. Of the cars that you can determine from the graph, Ford is liked less than the others.

Pie charts are useful for comparing sizes of categories. Bar charts show similar information. It really doesn't matter which one you use. It really is a personal preference and also what information you are trying to address. However, pie charts are best when you

only have a few categories and the data can be expressed as a percentage. The data doesn't have to be percentages to draw the pie chart, but if a data value can fit into multiple categories, you cannot use a pie chart. As an example, if you are asking people about what their favorite national park is, and you say to pick the top three choices, then the total number of answers can add up to more than 100% of the people involved. So you cannot use a pie chart to display the favorite national park.

A third type of qualitative data graph is a **Pareto chart**, which is just a bar chart with the bars sorted with the highest frequencies on the left. Here is the Pareto chart for the data in Example **3.1.1**.

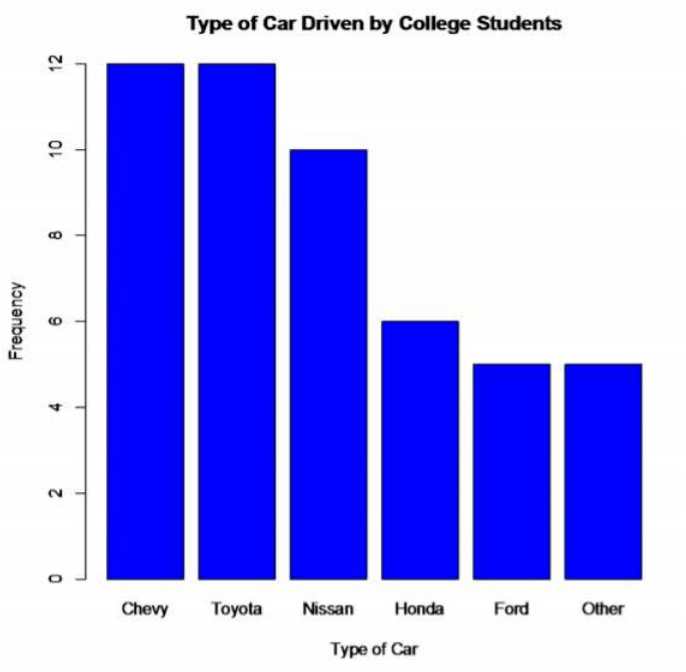


Figure **3.1.4**: Pareto Chart for Type of Car Data

The advantage of Pareto charts is that you can visually see the more popular answer to the least popular. This is especially useful in business applications, where you want to know what services your customers like the most, what processes result in more injuries, which issues employees find more important, and other type of questions like these.

There are many other types of graphs that can be used on qualitative data. There are spreadsheet software packages that will create most of them, and it is better to look at them to see what can be done. It depends on your data as to which may be useful. The next example illustrates one of these types known as a multiple bar graph.

Example 3.1.4 multiple bar graph

In the Wii Fit game, you can do four different types of exercises: yoga, strength, aerobic, and balance. The Wii system keeps track of how many minutes you spend on each of the exercises everyday. The following graph is the data for Dylan over one week time period. Discuss any indication you can infer from the graph.

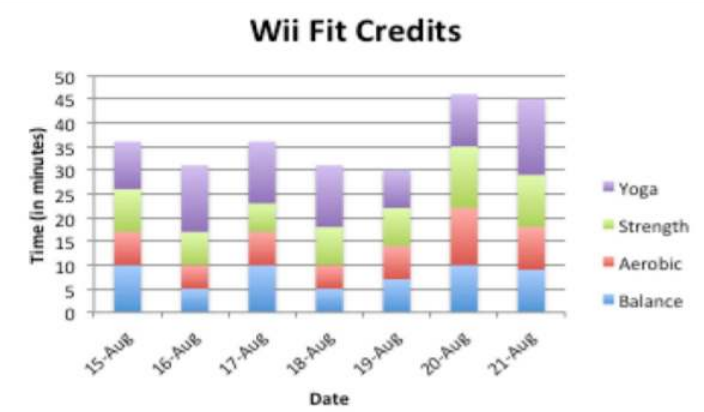


Figure 3.1.5: Multiple Bar Chart for Wii Fit Data

Solution

It appears that Dylan spends more time on balance exercises than on any other exercises on any given day. He seems to spend less time on strength exercises on a given day. There are several days when the amount of exercise in the different categories is almost equal.

The usefulness of a multiple bar graph is the ability to compare several different categories over another variable, in Example 3.1.4 the variable would be time. This allows a person to interpret the data with a little more ease.

Homework

1. Eyeglassomatic manufactures eyeglasses for different retailers. The number of lenses for different activities is in Example 3.1.4.

Activity	Grind	Multicoat	Assemble	Make frames	Receive finished	Unknown
Number of lenses	18872	12105	4333	25880	26991	1508

Table 3.1.4: *Data for Eyeglassomatic*

Grind means that they ground the lenses and put them in frames, multicoat means that they put tinting or scratch resistance coatings on lenses and then put them in frames, assemble means that they receive frames and lenses from other sources and put them together, make frames means that they make the frames and put lenses in from other sources, receive finished means that they received glasses from other source, and unknown means they do not know where the lenses came from. Make a bar chart and a pie chart of this data. State any findings you can see from the graphs.

2. To analyze how Arizona workers ages 16 or older travel to work the percentage of workers using carpool, private vehicle (alone), and public transportation was collected. Create a bar chart and pie chart of the data in Example 3.1.5. State any findings you can see from the graphs.

Table 3.1.5: Data of Travel Mode for Arizona Workers

Transportation type	Percentage
Carpool	11.6%
Private Vehicle (Alone)	75.8%
Public Transportation	2.0%
Other	10.6%

3. The number of deaths in the US due to carbon monoxide (CO) poisoning from generators from the years 1999 to 2011 are in table #2.1.6 (Hinatov, 2012). Create a bar chart and pie chart of this data. State any findings you see from the graphs.

Table **3.1.6**: Data of Number of Deaths Due to CO Poisoning

Region	Number of Deaths from CO While Using a Generator
Urban Core	401
Sub-Urban	97
Large Rural	86
Small Rural/Isolated	111

4. In Connecticut households use gas, fuel oil, or electricity as a heating source. Example **3.1.7** shows the percentage of households that use one of these as their principle heating sources ("Electricity usage," 2013), ("Fuel oil usage," 2013), ("Gas usage," 2013). Create a bar chart and pie chart of this data. State any findings you see from the graphs.

Table **3.1.7**: Data of Household Heating Sources

Heating Source	Percentage
Electricity	15.3%
Fuel Oil	46.3%
Gas	35.6%
Other	2.85

5. Eyeglassomatic manufactures eyeglasses for different retailers. They test to see how many defective lenses they made during the time period of January 1 to March 31. Example **3.1.8** gives the defect and the number of defects. Create a Pareto chart of the data and then describe what this tells you about what causes the most defects.

Table **3.1.8**: Data of Defect Type

Defect type	Number of defects
Scratch	5865
Right shaped - small	4613
Flaked	1992
Wrong axis	1838
Chamfer wrong	1596
Crazing, cracks	1546
Wrong shape	1485
Wrong PD	1398
Spots and bubbles	1371
Wrong height	1130
Right shape - big	1105
Lost in lab	976
Spots/bubble - intern	976

6. People in Bangladesh were asked to state what type of birth control method they use. The percentages are given in Example **3.1.9** ("Contraceptive use," 2013). Create a Pareto chart of the data and then state any findings you can from the graph.

Table **3.1.9**: Data of Birth Control Type

Method	Percentage
Condom	4.50%
Pill	28.50%
Periodic Abstinence	4.90%
Injection	7.00%
Female Sterilization	5.00%
IUD	0.90%
Male Sterilization	0.70%
Withdrawal	2.90%
Other Modern Methods	0.70%
Other Traditional Methods	0.60%

7. The percentages of people who use certain contraceptives in Central American countries are displayed in *Graph 2.1.6* ("Contraceptive use," 2013). State any findings you can from the graph.

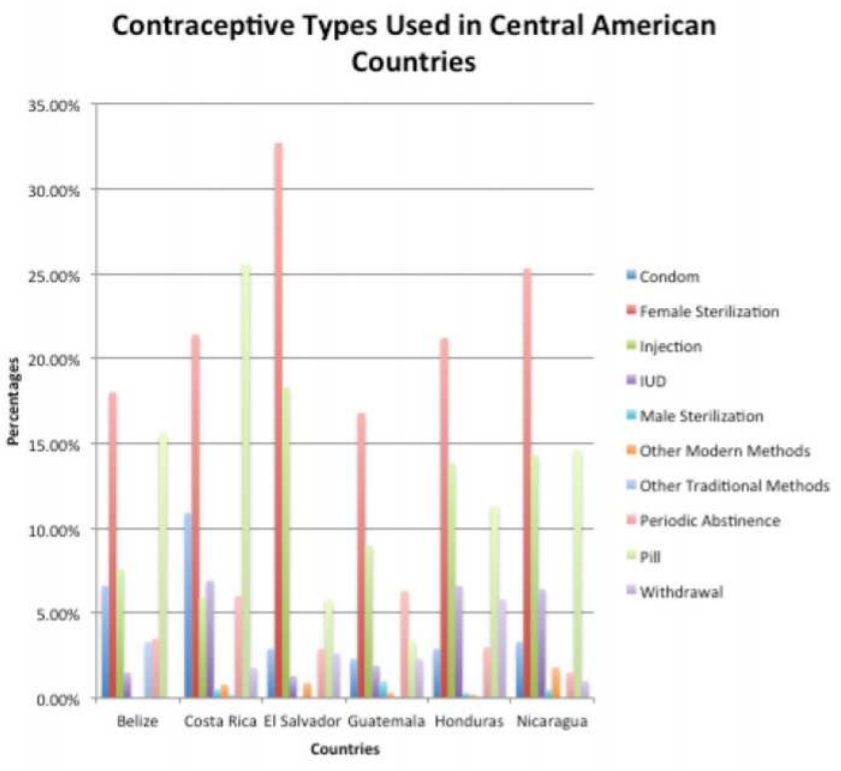


Figure **3.1.6***: Multiple Bar Chart for Contraceptive Types*

Answer

See solutions

This page titled 3.1: Qualitative Data is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Kathryn Kozak via source content that was edited to the style and standards of the LibreTexts platform.

- **2.1: Qualitative Data** by Kathryn Kozak is licensed CC BY-SA 4.0. Original source: https://s3-us-west-2.amazonaws.com/oerfiles/statsusingtech2.pdf.

3.2: Quantitative Data

The graph for quantitative data looks similar to a bar graph, except there are some major differences. First, in a bar graph the categories can be put in any order on the horizontal axis. There is no set order for these data values. You can't say how the data is distributed based on the shape, since the shape can change just by putting the categories in different orders. With quantitative data, the data are in specific orders, since you are dealing with numbers. With quantitative data, you can talk about a distribution, since the shape only changes a little bit depending on how many categories you set up. This is called a **frequency distribution**.

This leads to the second difference from bar graphs. In a bar graph, the categories that you made in the frequency table were determined by you. In quantitative data, the categories are numerical categories, and the numbers are determined by how many categories (or what are called classes) you choose. If two people have the same number of categories, then they will have the same frequency distribution. Whereas in qualitative data, there can be many different categories depending on the point of view of the author.

The third difference is that the categories touch with quantitative data, and there will be no gaps in the graph. The reason that bar graphs have gaps is to show that the categories do not continue on, like they do in quantitative data. Since the graph for quantitative data is different from qualitative data, it is given a new name. The name of the graph is a **histogram**. To create a histogram, you must first create the frequency distribution. The idea of a frequency distribution is to take the interval that the data spans and divide it up into equal subintervals called classes.

Summary of the Steps Involved in Making a Frequency Distribution

1. Find the range = largest value – smallest value
2. Pick the number of classes to use. Usually the number of classes is between five and twenty. Five classes are used if there are a small number of data points and twenty classes if there are a large number of data points (over 1000 data points). (Note: categories will now be called classes from now on.)
3. Class width = $\frac{\text{range}}{\#\text{ classes}}$ Always round up to the next integer (even if the answer is already a whole number go to the next integer). If you don't do this, your last class will not contain your largest data value, and you would have to add another class just for it. If you round up, then your largest data value will fall in the last class, and there are no issues.
4. Create the classes. Each class has limits that determine which values fall in each class. To find the class limits, set the smallest value as the lower class limit for the first class. Then add the class width to the lower class limit to get the next lower class limit. Repeat until you get all the classes. The upper class limit for a class is one less than the lower limit for the next class.
5. In order for the classes to actually touch, then one class needs to start where the previous one ends. This is known as the class boundary. To find the class boundaries, subtract 0.5 from the lower class limit and add 0.5 to the upper class limit.
6. Sometimes it is useful to find the class midpoint. The process is
   Midpoint $= \frac{\text{lower limit} + \text{upper limit}}{2}$
7. To figure out the number of data points that fall in each class, go through each data value and see which class boundaries it is between. Utilizing tally marks may be helpful in counting the data values. The frequency for a class is the number of data values that fall in the class.

Note

The above description is for data values that are whole numbers. If you data value has decimal places, then your class width should be rounded up to the nearest value with the same number of decimal places as the original data. In addition, your class boundaries should have one more decimal place than the original data. As an example, if your data have one decimal place, then the class width would have one decimal place, and the class boundaries are formed by adding and subtracting 0.05 from each class limit.

Example 3.2.1 creating a frequency table

Example **3.2.1** contains the amount of rent paid every month for 24 students from a statistics course. Make a relative frequency distribution using 7 classes.

Table **3.2.1**: Data of Monthly Rent

1500	1350	350	1200	850	900
1500	1150	1500	900	1400	1100
1250	600	610	960	890	1325
900	800	2550	495	1200	690

Solution

1. Find the range:
   largest value - smallest value $= 2550 - 350 = 2200$
2. Pick the number of classes:
   The directions to say to use 7 classes.
3. Find the class width:
   width $= \frac{\text{range}}{7} = \frac{2200}{7} \approx 314.286$
   Round up to 315
   $Always round up to the next integer even if the width is already an integer.$
4. Find the class limits:
   Start at the smallest value. This is the lower class limit for the first class. Add the width to get the lower limit of the next class. Keep adding the width to get all the lower limits.
   $350 + 315 = 665, 665 + 315 = 980, 980 + 315 = 1295 \rightleftharpoons,$
   The upper limit is one less than the next lower limit: so for the first class the upper class limit would be $665 - 1 = 664$.
   When you have all 7 classes, make sure the last number, in this case the 2550, is at least as large as the largest value in the data. If not, you made a mistake somewhere.
5. Find the class boundaries:
   Subtract 0.5 from the lower class limit to get the class boundaries. Add 0.5 to the upper class limit for the last class's boundary.
   $350 - 0.5 = 349.5, \quad 665 - 0.5 = 664.5, \quad 980 - 0.5 = 979.5, \quad 1295 - 0.5 = 1294.5 \rightleftharpoons$
   Every value in the data should fall into exactly one of the classes. No data values should fall right on the boundary of two classes.
6. Find the class midpoints:
   midpoint $= \frac{\text{lower limit} + \text{upper limit}}{2}$
   $\frac{350 + 664}{2} = 507, \frac{665 + 979}{2} = 822, \rightleftharpoons$
7. Tally and find the frequency of the data:
   Go through the data and put a tally mark in the appropriate class for each piece of data by looking to see which class boundaries the data value is between. Fill in the frequency by changing each of the tallies into a number.

Table **3.2.2**: *Frequency Distribution for Monthly Rent*

Class Limits	Class Boundaries	Class Midpoint	Tally	Frequency
350-664	349.5-664.5	507	||||	4
665-979	664.5-979.5	822	𝍸 |||	8
980-1294	979.5-1294.5	1137	𝍸	5
1295-1609	1294.5-1609.5	1452	𝍸 |	6
1610-1924	1609.5-1924.5	1767		0
1925-2239	1924.5-2239.5	2082		0
2240-2554	2239.5-2554.5	2397	|	1

Make sure the total of the frequencies is the same as the number of data points.

R command for a frequency distribution:

To create a frequency distribution:

summary(variable) – so you can find out the minimum and maximum.

breaks = seq(min, number above max, by = class width)

breaks – so you can see the breaks that R made.

variable.cut=cut(variable, breaks, right=FALSE) – this will cut up the data into the classes.

variable.freq=table(variable.cut) – this will create the frequency table.

variable.freq – this will display the frequency table.

For the data in Example **3.2.1**, the R command would be:

rent<-c(1500, 1350, 350, 1200, 850, 900, 1500, 1150, 1500, 900, 1400, 1100, 1250, 600, 610, 960, 890, 1325, 900, 800, 2550, 495, 1200, 690) summary(rent)

Output:

Min	1st Qu.	Median	Mean	3rd Qu.	Max
350	837.5	1030.0	1082.0	1331.0	2550.0

breaks=seq(350, 3000, by = 315)
breaks

Output:
[1] 350 665 980 1295 1610 1925 2240 2555 2870
These are your lower limits of the frequency distribution. You can now write your own table.

rent.cut=cut(rent, breaks, right=FALSE)
rent.freq=table(rent.cut)

Output:
rent.cut

$$[350, 665) \quad [665, 980) \quad [980, 1.3e+03) \quad [1.3e+03, 1.61e+03) \quad [1.61e+03, 1.92e+03) \quad [1.92e+03, 2.22e+03) \quad [2.22e+03, 2.56e+03) \quad [2.56e+03, 2.87e+03)$$
$$4 \qquad 8 \qquad 5 \qquad 6 \qquad 0 \qquad 0 \qquad 1 \qquad 0$$

It is difficult to determine the basic shape of the distribution by looking at the frequency distribution. It would be easier to look at a graph. The graph of a frequency distribution for quantitative data is called a **frequency histogram** or just histogram for short.

Definition 3.2.1: Histogram

A Histogram is a graph of the frequencies on the vertical axis and the class boundaries on the horizontal axis. Rectangles where the height is the frequency and the width is the class width are drawn for each class.

Example \(\PageIndex{2}\): Drawing a Histogram

Draw a histogram for the distribution from Example **3.2.1**.

Solution

The class boundaries are plotted on the horizontal axis and the frequencies are plotted on the vertical axis. You can plot the midpoints of the classes instead of the class boundaries. *Graph 2.2.1* was created using the midpoints because it was easier to do with the software that created the graph. On R, the command is

hist(variable, col="type in what color you want", breaks, main="type the title you want", xlab="type the label you want for the horizontal axis",
ylim=c(0, number above maximum frequency) – produces histogram with specified color and using the breaks you made for the frequency distribution.

For this example, the command in R would be (assuming you created a frequency distribution in R as described previously):

hist(rent, col="blue", breaks, right=FALSE, main="Monthly Rent Paid by Students", ylim=c(0,8) xlab="Monthly Rent ($)")

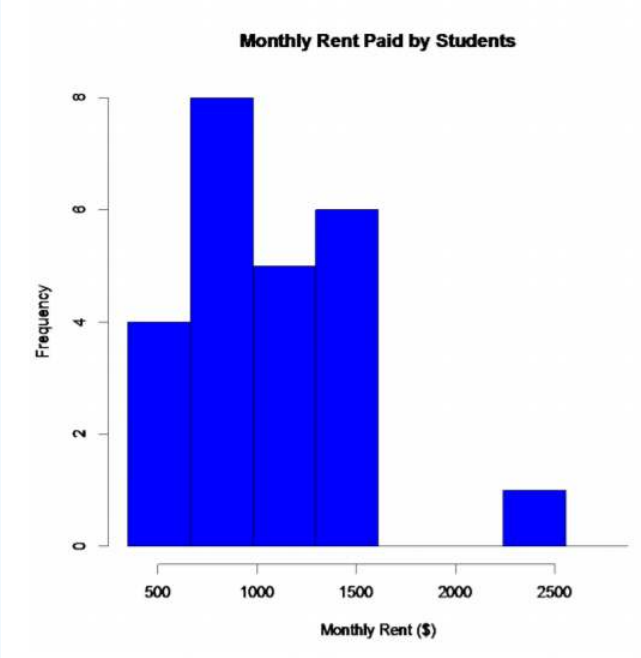


Figure **3.2.1**: Histogram for Monthly Rent

If no frequency distribution was created before the histogram, then the command would be:

> hist(variable, col="type in what color you want", number of classes, main="type the title you want", xlab="type the label you want for the horizontal axis") – produces histogram with specified color and number of classes (though the number of classes is an estimate and R will create the number of classes near this value).

For this example, the R command without a frequency distribution created first would be:

> hist(rent, col="blue", 7, main="Monthly Rent Paid by Students", xlab="Monthly Rent ($)")

Notice the graph has the axes labeled, the tick marks are labeled on each axis, and there is a title.

Reviewing the graph you can see that most of the students pay around $750 per month for rent, with about $1500 being the other common value. You can see from the graph, that most students pay between $600 and $1600 per month for rent. Of

course, these values are just estimates from the graph. There is a large gap between the $1500 class and the highest data value. This seems to say that one student is paying a great deal more than everyone else. This value could be considered an outlier. An **outlier** is a data value that is far from the rest of the values. It may be an unusual value or a mistake. It is a data value that should be investigated. In this case, the student lives in a very expensive part of town, thus the value is not a mistake, and is just very unusual. There are other aspects that can be discussed, but first some other concepts need to be introduced.

Frequencies are helpful, but understanding the relative size each class is to the total is also useful. To find this you can divide the frequency by the total to create a relative frequency. If you have the relative frequencies for all of the classes, then you have a relative frequency distribution.

Definition 3.2.2

Relative Frequency Distribution

A variation on a frequency distribution is a relative frequency distribution. Instead of giving the frequencies for each class, the relative frequencies are calculated.

$$\text{Relative frequency} = \frac{\text{frequency}}{\text{\# of data points}}$$

This gives you percentages of data that fall in each class.

Example 3.2.3 creating a relative frequency table

Find the relative frequency for the grade data.

Solution

From Example 3.2.1, the frequency distribution is reproduced in Example 3.2.2.

Table 3.2.2: Frequency Distribution for Monthly Rent

Class Limits	Class Boundaries	Class Midpoint	Frequency
350-664	349.5-664.5	507	4
665-979	664.5-979.5	822	8
980-1294	979.5-1294.5	1127	5
1295-1609	1294.5-1609.5	1452	6
1610-1924	1609.5-1924.5	1767	0
1925-2239	1924.5-2239.5	2082	0
2240-2554	2239.5-2554.5	2397	1

Divide each frequency by the number of data points.

$$\frac{4}{24} = 0.17, \frac{8}{24} = 0.33, \frac{5}{24} = 0.21, \rightleftharpoons$$

Table 3.2.3: Relative Frequency Distribution for Monthly Rent

Class Limits	Class Boundaries	Class Midpoint	Frequency	Relative Frequency
350-664	349.5-664.5	507	4	0.17
665-979	664.5-979.5	822	8	0.33
980-1294	979.5-1294.5	1127	5	0.21
1295-1609	1294.5-1609.5	1452	6	0.25
1610-1924	1609.5-1924.5	1767	0	0
1925-2239	1924.5-2239.5	2082	0	0
2240-2554	2239.5-2554.5	2397	1	0.04
Total			24	1

The relative frequencies should add up to 1 or 100%. (This might be off a little due to rounding errors.)

The graph of the relative frequency is known as a relative frequency histogram. It looks identical to the frequency histogram, but the vertical axis is relative frequency instead of just frequencies.

Example 3.2.4 drawing a relative frequency histogram

Draw a relative frequency histogram for the grade distribution from Example **3.2.1**.

Solution

The class boundaries are plotted on the horizontal axis and the relative frequencies are plotted on the vertical axis. (This is not easy to do in R, so use another technology to graph a relative frequency histogram.)



Figure **3.2.2**: Relative Frequency Histogram for Monthly Rent

Notice the shape is the same as the frequency distribution.

Another useful piece of information is how many data points fall below a particular class boundary. As an example, a teacher may want to know how many students received below an 80%, a doctor may want to know how many adults have cholesterol below 160, or a manager may want to know how many stores gross less than $2000 per day. This is known as a **cumulative frequency**. If you want to know what percent of the data falls below a certain class boundary, then this would be a **cumulative relative frequency**. For cumulative frequencies you are finding how many data values fall below the upper class limit.

To create a **cumulative frequency distribution**, count the number of data points that are below the upper class boundary, starting with the first class and working up to the top class. The last upper class boundary should have all of the data points below it. Also include the number of data points below the lowest class boundary, which is zero.

Example 3.2.5 creating a cumulative frequency distribution

Create a cumulative frequency distribution for the data in Example 3.2.1.

Solution

The frequency distribution for the data is in Example 3.2.2.

Table 3.2.2: Frequency Distribution for Monthly Rent

Class Limits	Class Boundaries	Class Midpoint	Frequency
350-664	349.5-664.5	507	4
665-979	664.5-979.5	822	8
980-1294	979.5-1294.5	1127	5
1295-1609	1294.5-1609.5	1452	6
1610-1924	1609.5-1924.5	1767	0
1925-2239	1924.5-2239.5	2082	0
2240-2554	2239.5-2554.5	2397	1

Now ask yourself how many data points fall below each class boundary. Below 349.5, there are 0 data points. Below 664.5 there are 4 data points, below 979.5, there are 4 + 8 = 12 data points, below 1294.5 there are 4 + 8 + 5 = 17 data points, and continue this process until you reach the upper class boundary. This is summarized in Example 3.2.4.

To produce cumulative frequencies in R, you need to have performed the commands for the frequency distribution. Once you have complete that, then use variable.cumfreq=cumsum(variable.freq) – creates the cumulative frequencies for the variable
cumfreq0=c(0,variable.cumfreq) – creates a cumulative frequency table for the variable.
cumfreq0 – displays the cumulative frequency table.

For this example the command would be:
rent.cumfreq=cumsum(rent.freq)
cumfreq0=c(0,rent.cumfreq)
cumfreq0

Output:

[350, 665, 980, 1.3e+03, 1.61e+03, 1.92e+03, 2.24e+03, 2.56e+03, 2.87e+03)
0 4 12 17 23 23 23 24 24

Now type this into a table. See Example 3.2.4.

Table 3.2.4: Cumulative Distribution for Monthly Rent

Class Limits	Class Boundaries	Class Midpoint	Frequency	Cumulative Frequency
350-664	349.5-664.5	507	4	4
665-979	664.5-979.5	822	8	12
980-1294	979.5-1294.5	1127	5	17
1295-1609	1294.5-1609.5	1452	6	23
1610-1924	1609.5-1924.5	1767	0	23
1925-2239	1924.5-2239.5	2082	0	23
2240-2554	2239.5-2554.5	2397	1	24

Again, it is hard to look at the data the way it is. A graph would be useful. The graph for cumulative frequency is called an **ogive** (o-jive). To create an ogive, first create a scale on both the horizontal and vertical axes that will fit the data. Then plot the points of the class upper class boundary versus the cumulative frequency. Make sure you include the point with the lowest class boundary and the 0 cumulative frequency. Then just connect the dots.

Example 3.2.6 drawing an ogive

Draw an ogive for the data in Example **3.2.1**.

Solution

In R, the commands would be:
plot(breaks,cumfreq0, main="title you want to use", xlab="label you want to use", ylab="label you want to use", ylim=c(0, number above maximum cumulative frequency) – plots the ogive
lines(breaks,cumfreq0) – connects the dots on the ogive

For this example, the commands would be:
Plot(breaks,cumfreq0, main="Cumulative Frequency for Monthly Rent", xlab="Monthly Rent ($)", ylab="Cumulative Frequency", ylim=c(0,25))
lines(breaks,cumfreq0)

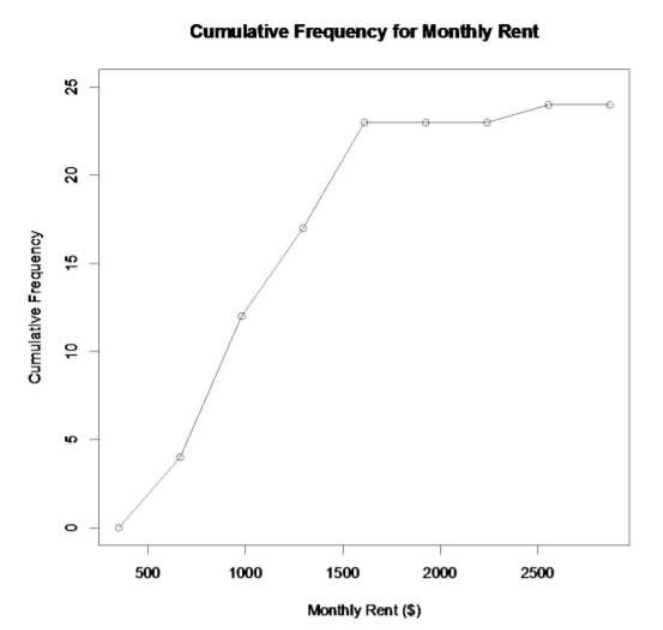


Figure **3.2.3**: Ogive for Monthly Rent

The usefulness of a ogive is to allow the reader to find out how many students pay less than a certain value, and also what amount of monthly rent is paid by a certain number of students. As an example, suppose you want to know how many students pay less than $1500 a month in rent, then you can go up from the $1500 until you hit the graph and then you go over to the cumulative frequency axes to see what value corresponds to this value. It appears that around 20 students pay less than $1500. (See *Graph 2.2.4.*)

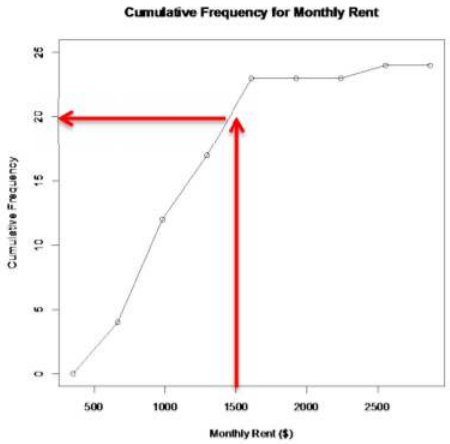


Figure **3.2.4**: Ogive for Monthly Rent with Example

Also, if you want to know the amount that 15 students pay less than, then you start at 15 on the vertical axis and then go over to the graph and down to the horizontal axis where the line intersects the graph. You can see that 15 students pay less than about $1200 a month. (See *Graph 2.2.5*.)

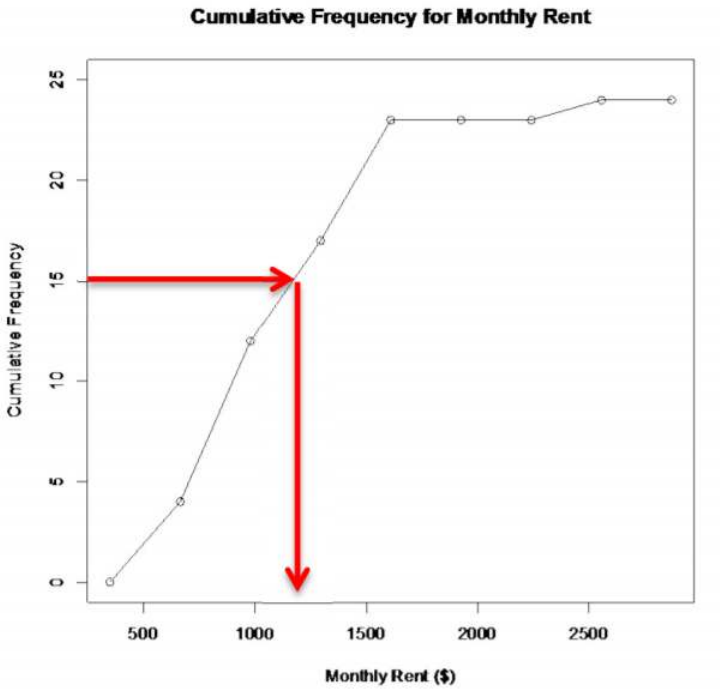


Figure **3.2.5**: Ogive for Monthly Rent with Example

If you graph the cumulative relative frequency then you can find out what percentage is below a certain number instead of just the number of people below a certain value.

Shapes of the distribution:

When you look at a distribution, look at the basic shape. There are some basic shapes that are seen in histograms. Realize though that some distributions have no shape. The common shapes are symmetric, skewed, and uniform. Another interest is how many peaks a graph may have. This is known as modal.

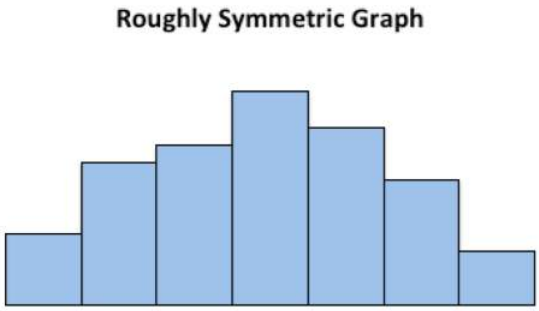
Symmetric means that you can fold the graph in half down the middle and the two sides will line up. You can think of the two sides as being mirror images of each other. Skewed means one “tail” of the graph is longer than the other. The graph is skewed in the direction of the longer tail (backwards from what you would expect). A uniform graph has all the bars the same height.

Modal refers to the number of peaks. Unimodal has one peak and bimodal has two peaks. Usually if a graph has more than two peaks, the modal information is not longer of interest.

Other important features to consider are gaps between bars, a repetitive pattern, how spread out is the data, and where the center of the graph is.

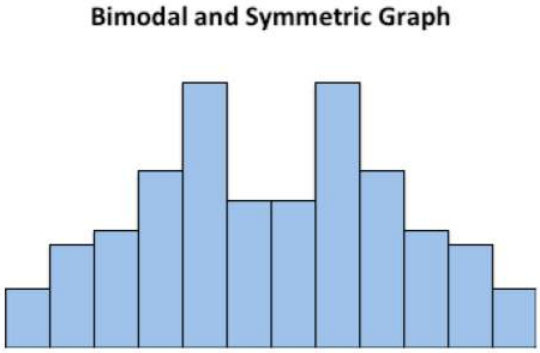
Examples of Graphs:

This graph is roughly symmetric and unimodal:



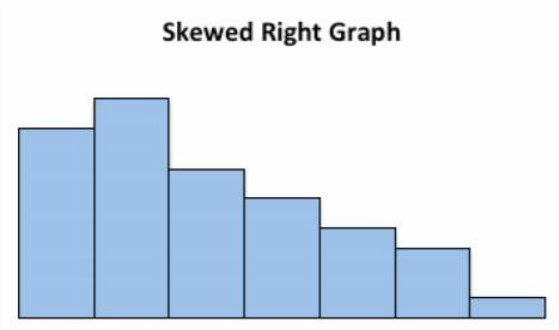
Figure

This graph is symmetric and bimodal:



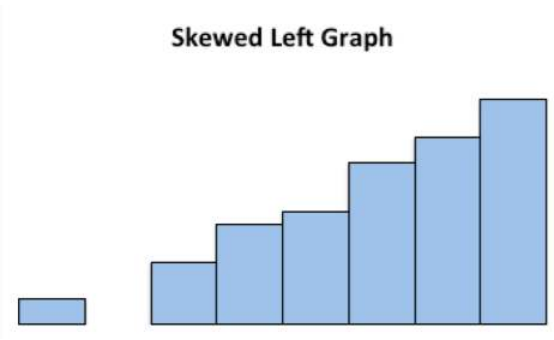
Figure

This graph is skewed to the right:



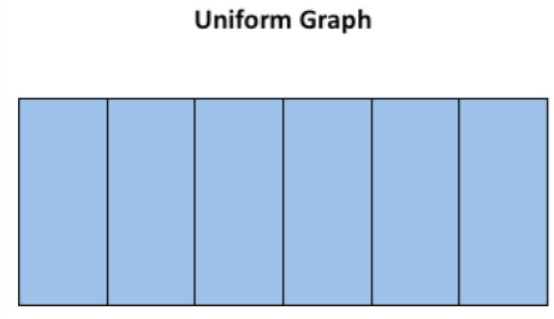
Figure

This graph is skewed to the left and has a gap:



Figure

This graph is uniform since all the bars are the same height:



Figure

Example 3.2.7 creating a frequency distribution, histogram, and ogive

The following data represents the percent change in tuition levels at public, fouryear colleges (inflation adjusted) from 2008 to 2013 (Weissmann, 2013). Create a frequency distribution, histogram, and ogive for the data.

Table 3.2.5: Data of Tuition Levels at Public, Four-Year Colleges

19.5%	40.8%	57.0%	15.1%	17.4%	5.2%	13.0%
15.6%	51.5%	15.6%	14.5%	22.4%	19.5%	31.3%
21.7%	27.0%	13.1%	26.8%	24.3%	38.0%	21.1%
9.3%	46.7%	14.5%	78.4%	67.3%	21.1%	22.4%
5.3%	17.3%	17.5%	36.6%	72.0%	63.2%	15.1%
2.2%	17.5%	36.7%	2.8%	16.2%	20.5%	17.8%
30.1%	63.6%	17.8%	23.2%	25.3%	21.4%	28.5%
9.4%						

Solution

1. Find the range:
   largest value - smallest value = $78.4\% - -2.2\% = 76.2\%$
2. Pick the number of classes:
   Since there are 50 data points, then around 6 to 8 classes should be used. Let's use 8.
3. Find the class width:
   $\text{width} = \frac{\text{range}}{8} = \frac{76.2\%}{8} \approx 9.525\%$
   Since the data has one decimal place, then the class width should round to one decimal place. Make sure you round up.
   $\text{width} = 9.6\%$

4. Find the class limits:
   $2.2\% + 9.6\% = 11.8\%, 11.8\% + 9.6\% = 21.4\%, 21.4\% + 9.6\% = 31.0\%, \leftrightharpoons$
5. Find the class boundaries:
   Since the data has one decimal place, the class boundaries should have two decimal places, so subtract 0.05 from the lower class limit to get the class boundaries. Add 0.05 to the upper class limit for the last class's boundary.
   $2.2 - 0.05 = 2.15\%, 11.8 - 0.05 = 11.75\%, 21.4 - 0.05 = 21.35\% \leftrightharpoons$
   Every value in the data should fall into exactly one of the classes. No data values should fall right on the boundary of two classes.
6. Find the class midpoints:
   $\text{midpoint} = \frac{\text{lower limt } + \text{ upper limit}}{2}$
   $\frac{2.2 + 11.7}{2} = 6.95\%, \frac{11.8 + 21.3}{2} = 16.55\%, \leftrightharpoons$
7. Tally and find the frequency of the data:

Table **3.2.6**: Frequency Distribution for Tuition Levels at Public, Four-Year Colleges

Class Limits	Class Boundaries	Class Midpoint	Tally	Frequency	Relative Frequency	Cumulative Frequency
2.2-11.7	2.15-11.75	6.95	𝍸 |	6	0.12	6
11.8-21.3	11.75-21.35	16.55	𝍸 𝍸 𝍸 𝍸	20	0.40	26
21.4-30.9	21.35-30.95	26.15	𝍸 𝍸 |	11	0.22	37
31.0-45.0	30.95-40.55	35.75	||||	4	0.08	41
40.6-50.1	40.55-50.15	45.35	||	2	0.04	43
50.2-59.7	50.15-59.75	54.95	||	2	0.04	45
59.8-69.3	59.75-69.35	64.55	|||	3	0.06	48
69.4-78.9	69.35-78.95	74.15	||	2	0.04	50

Make sure the total of the frequencies is the same as the number of data points.

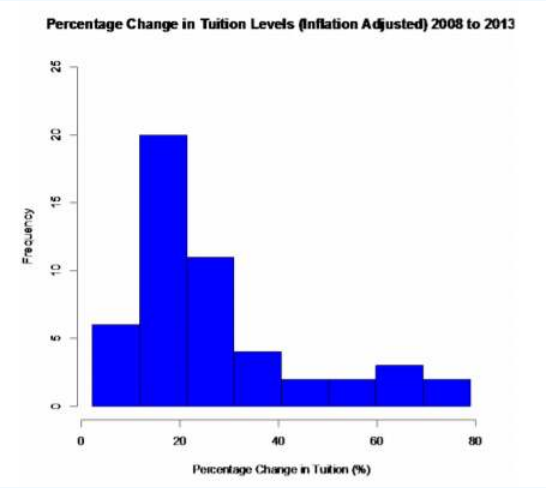


Figure **3.2.11**: Histogram for Tuition Levels at Public, Four-Year Colleges

This graph is skewed right, with no gaps. This says that most percent increases in tuition were around 16.55%, with very few states having a percent increase greater than 45.35%.

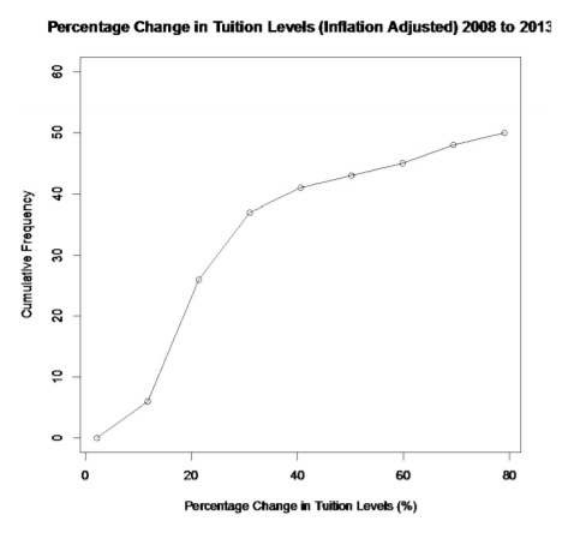


Figure **3.2.12**: *Ogive for Tuition Levels at Public, Four-Year Colleges*

Looking at the ogive, you can see that 30 states had a percent change in tuition levels of about 25% or less.

There are occasions where the class limits in the frequency distribution are predetermined. Example **3.2.8** demonstrates this situation.

Example 3.2.8 creating a frequency distribution and histogram

The following are the percentage grades of 25 students from a statistics course. Make a frequency distribution and histogram.

Table **3.2.7**: Data of Test Grades

62	87	81	69	87	62	45	95	76	76
62	71	65	67	72	80	40	77	87	58
84	73	93	64	89					

Solution

Since this data is percent grades, it makes more sense to make the classes in multiples of 10, since grades are usually 90 to 100%, 80 to 90%, and so forth. It is easier to not use the class boundaries, but instead use the class limits and think of the upper class limit being up to but not including the next classes lower limit. As an example the class 80 – 90 means a grade of 80% up to but not including a 90%. A student with an 89.9% would be in the 80-90 class.

Table **3.2.8**: Frequency Distribution for Test Grades

Class Limit	Class Midpoint	Tally	Freqeuncy
40-50	45	||	2
50-60	55			1
60-70	65	𝍸 ||	7
70-80	75	𝍸		6
80-90	85	𝍸 ||	7
90-100	95	||	2

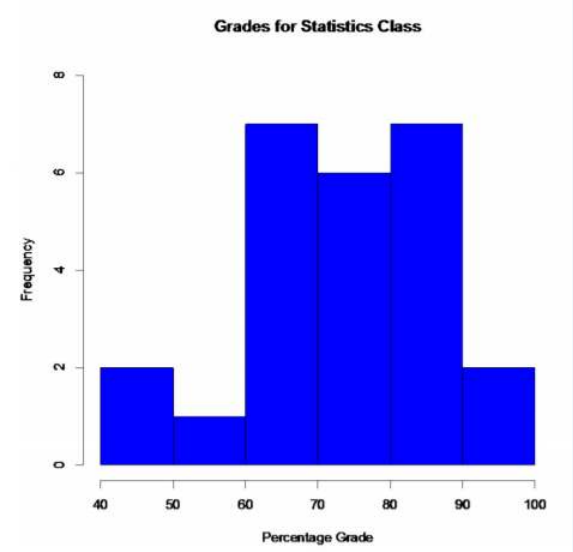


Figure **3.2.13**: Histogram for Test Grades

It appears that most of the students had between 60 to 90%. This graph looks somewhat symmetric and also bimodal. The same number of students earned between 60 to 70% and 80 to 90%.

There are other types of graphs for quantitative data. They will be explored in the next section.

Homework

1. The median incomes of males in each state of the United States, including the District of Columbia and Puerto Rico, are given in Example **3.2.9** ("Median income of," 2013). Create a frequency distribution, relative frequency distribution, and cumulative frequency distribution using 7 classes.

Table **3.2.9**: Data of Median Income for Males

$42,951	$52,379	$42,544	$37,488	$49,281	$50,987
$60,705	$50,411	$66,760	$40,951	$43,902	$45,494
$41,528	$50,746	$45,183	$43,624	$43,993	$41,612
$46,313	$43,944	$56,708	$60,264	$50,053	$50,580
$40,202	$43,146	$41,635	$42,182	$41,803	$53,033
$60,568	$41,037	$50,388	$41,950	$44,660	$46,176
$41,420	$45,976	$47,956	$22,529	$48,842	$41,464
$40,285	$41,309	$43,160	$47,573	$44,057	$52,805
$53,046	$42,125	$46,214	$51,630		

2. The median incomes of females in each state of the United States, including the District of Columbia and Puerto Rico, are given in Example **3.2.10** ("Median income of," 2013). Create a frequency distribution, relative frequency distribution, and cumulative frequency distribution using 7 classes.

Table **3.2.10**: Data of Median Income for Females

$31,862	$40,550	$36,048	$30,752	$41,817	$40,236
$47,476	$40,500	$60,332	$33,823	$35,438	$37,242
$31,238	$39,150	$34,023	$33,745	$33,269	$32,684
$31,844	$34,599	$48,748	$46,185	$36,931	$40,416
$29,548	$33,865	$31,067	$33,424	$35,484	$41,021
$47,155	$32,316	$42,113	$33,459	$32,462	$35,746
$31,274	$36,027	$37,089	$22,117	$41,412	$31,330
$31,329	$33,184	$35,301	$32,843	$38,177	$40,969
$40,993	$29,688	$35,890	$34,381		

3. The density of people per square kilometer for African countries is in Example **3.2.11** ("Density of people," 2013). Create a frequency distribution, relative frequency distribution, and cumulative frequency distribution using 8 classes.

Table **3.2.11**: Data of Density of People per Square Kilometer

15	16	81	3	62	367	42	123
8	9	337	12	29	70	39	83
26	51	79	6	157	105	42	45
72	72	37	4	36	134	12	3
630	563	72	29	3	13	176	341
415	187	65	194	75	16	41	18
69	49	103	65	143	2	18	31

4. The Affordable Care Act created a market place for individuals to purchase health care plans. In 2014, the premiums for a 27 year old for the bronze level health insurance are given in Example **3.2.12** ("Health insurance marketplace," 2013). Create a frequency distribution, relative frequency distribution, and cumulative frequency distribution using 5 classes.

Table **3.2.12**: Data of Health Insurance Premiums

$114	$119	$121	$125	$132	$139
$139	$141	$143	$145	$151	$153
$156	$159	$162	$163	$165	$166
$170	$170	$176	$177	$181	$185
$185	$186	$186	$189	$190	$192
$196	$203	$204	$219	$254	$286

5. Create a histogram and relative frequency histogram for the data in Example **3.2.9**. Describe the shape and any findings you can from the graph.

6. Create a histogram and relative frequency histogram for the data in Example **3.2.10**. Describe the shape and any findings you can from the graph.

7. Create a histogram and relative frequency histogram for the data in Example **3.2.11**. Describe the shape and any findings you can from the graph.

8. Create a histogram and relative frequency histogram for the data in Example **3.2.12**. Describe the shape and any findings you can from the graph.

9. Create an ogive for the data in Example **3.2.9**. Describe any findings you can from the graph.

10. Create an ogive for the data in Example **3.2.10**. Describe any findings you can from the graph.

11. Create an ogive for the data in Example **3.2.11**. Describe any findings you can from the graph.

12. Create an ogive for the data in Example **3.2.12**. Describe any findings you can from the graph.

13. Students in a statistics class took their first test. The following are the scores they earned. Create a frequency distribution and histogram for the data using class limits that make sense for grade data. Describe the shape of the distribution.

Table **3.2.13**: Data of Test 1 Grades

80	79	89	74	73	67	79
93	70	70	76	88	83	73
81	79	80	85	79	80	79
58	93	94	74			

14. Students in a statistics class took their first test. The following are the scores they earned. Create a frequency distribution and histogram for the data using class limits that make sense for grade data. Describe the shape of the distribution. Compare to the graph in question 13.

Table **3.2.14**: Data of Test 1 Grades

67	67	76	47	85	70
87	76	80	72	84	98
84	64	65	82	81	81
88	74	87	83		

Answer

See solutions

This page titled 3.2: Quantitative Data is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Kathryn Kozak via source content that was edited to the style and standards of the LibreTexts platform.

- **2.2: Quantitative Data** by Kathryn Kozak is licensed CC BY-SA 4.0. Original source: https://s3-us-west-2.amazonaws.com/oerfiles/statsusingtech2.pdf.

3.3: Other Graphical Representations of Data

There are many other types of graphs. Some of the more common ones are the frequency polygon, the dot plot, the stem plot, scatter plot, and a time-series plot. There are also many different graphs that have emerged lately for qualitative data. Many are found in publications and websites. The following is a description of the stem plot, the scatter plot, and the time-series plot.

Stem Plots

Stem plots are a quick and easy way to look at small samples of numerical data. You can look for any patterns or any strange data values. It is easy to compare two samples using stem plots.

The first step is to divide each number into 2 parts, the stem (such as the leftmost digit) and the leaf (such as the rightmost digit). There are no set rules, you just have to look at the data and see what makes sense.

Example 3.3.1 stem plot for grade distribution

The following are the percentage grades of 25 students from a statistics course. Draw a stem plot of the data.

Table **3.3.1**: Data of Test Grades

62	87	81	69	87	62	45	95	76	76
62	71	65	67	72	80	40	77	87	58
84	73	93	64	89					

Solution

Divide each number so that the tens digit is the stem and the ones digit is the leaf. 62 becomes 6|2.

Make a vertical chart with the stems on the left of a vertical bar. Be sure to fill in any missing stems. In other words, the stems should have equal spacing (for example, count by ones or count by tens). The *Graph 2.3.1* shows the stems for this example.

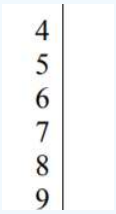


Figure **3.3.1**: Stem Plot for Test Grades Step 1

Now go through the list of data and add the leaves. Put each leaf next to its corresponding stem. Don't worry about order yet just get all the leaves down.

When the data value 62 is placed on the plot it looks like the plot in *Graph 2.3.2*.

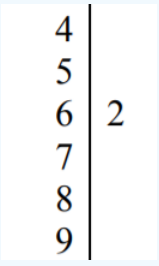


Figure **3.3.2**: Stem Plot for Test Grades Step 2

When the data value 87 is placed on the plot it looks like the plot in *Graph 2.3.3*.

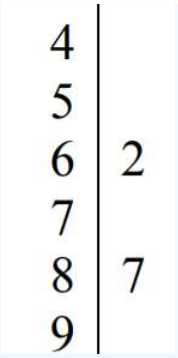


Figure **3.3.3**: Stem Plot for Test Grades Step 3

Filling in the rest of the leaves to obtain the plot in *Graph 2.3.4*.

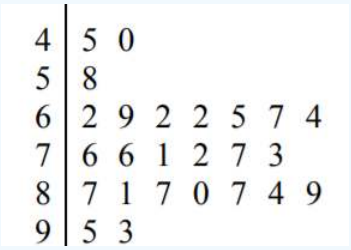


Figure **3.3.4**: Stem Plot for Test Grades Step 4

Now you have to add labels and make the graph look pretty. You need to add a label and sort the leaves into increasing order. You also need to tell people what the stems and leaves mean by inserting a legend. **Be careful to line the leaves up in columns**. You need to be able to compare the lengths of the rows when you interpret the graph. The final stem plot for the test grade data is in *Graph 2.3.5*.

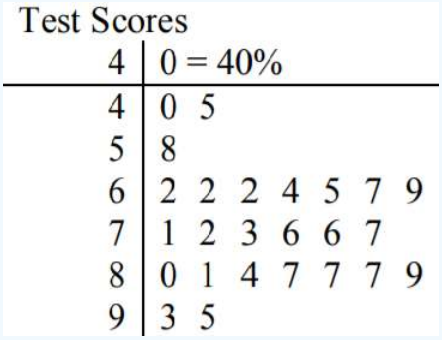


Figure **3.3.5**: Stem Plot for Test Grades

Now you can interpret the stem-and-leaf display. The data is bimodal and somewhat symmetric. There are no gaps in the data. The center of the distribution is around 70.

You can create a stem and leaf plot on R. the command is:

stem(variable) – creates a stem and leaf plot, if you do not get a stem plot that shows all of the stems then use scale = a number. Adjust the number until you see all of the stems. So you would have stem(variable, scale = a number)

For Example **3.3.1**, the command would be

grades<-c(62, 87, 81, 69, 87, 62, 45, 95, 76, 76, 62, 71, 65, 67, 72, 80, 40, 77, 87, 58, 84, 73, 93, 64, 89)
stem(grades, scale = 2)

Output:

The decimal point is 1 digit(s) to the right of the |

4 | 05
5 | 8
6 | 2224579
7 | 123667
8 | 0147779
9 | 35

Now just put a title on the stem plot.

Scatter Plot

Sometimes you have two different variables and you want to see if they are related in any way. A scatter plot helps you to see what the relationship would look like. A scatter plot is just a plotting of the ordered pairs.

Example 3.3.2 scatter plot

Is there any relationship between elevation and high temperature on a given day? The following data are the high temperatures at various cities on a single day and the elevation of the city.

Table **3.3.2**: Data of Temperature versus Elevation

Elevation (in feet)	7000	4000	6000	3000	7000	4500	5000
Temperature (°F)	50	60	48	70	55	55	60

Solution

Preliminary: State the random variables

Let x = altitude

y = high temperature

Now plot the x values on the horizontal axis, and the y values on the vertical axis. Then set up a scale that fits the data on each axes. Once that is done, then just plot the x and y values as an ordered pair. In R, the command is:

independent variable<-c(type in data with commas in between values)
dependent variable<-c(type in data with commas in between values)
plot(independent variable, dependent variable, main="type in a title you want", xlab="type in a label for the horizontal axis", ylab="type in a label for the vertical axis", ylim=c(0, number above maximum y value)

For this example, that would be:
elevation<-c(7000, 4000, 6000, 3000, 7000, 4500, 5000)
temperature<-c(50, 60, 48, 70, 55, 55, 60)
plot(elevation, temperature, main="Temperature versus Elevation", xlab="Elevation (in feet)", ylab="Temperature (in degrees F)", ylim=c(0, 80))

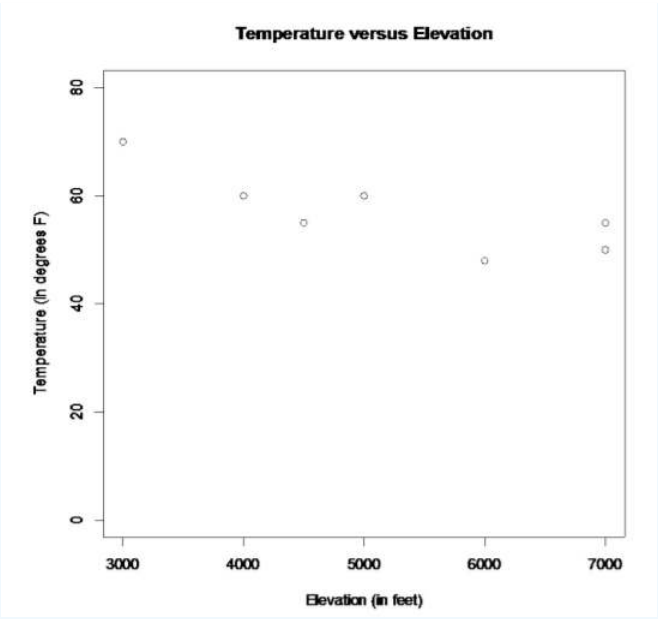


Figure **3.3.6**: Scatter Plot of Temperature versus Elevation

Looking at the graph, it appears that there is a linear relationship between temperature and elevation. It also appears to be a negative relationship, thus as elevation increases, the temperature decreases.

Time-Series

A time-series plot is a graph showing the data measurements in chronological order, the data being quantitative data. For example, a time-series plot is used to show profits over the last 5 years. To create a time-series plot, the time always goes on the horizontal axis, and the other variable goes on the vertical axis. Then plot the ordered pairs and connect the dots. The purpose of a time-series graph is to look for trends over time. Caution, you must realize that the trend may not continue. Just because you see an increase, doesn't mean the increase will continue forever. As an example, prior to 2007, many people noticed that housing prices were increasing. The belief at the time was that housing prices would continue to increase. However, the housing bubble burst in 2007, and many houses lost value, and haven't recovered.

Example **3.3.3** Time-series plot

The following table tracks the weight of a dieter, where the time in months is measuring how long since the person started the diet

Table **3.3.3**: Data of Weights versus Time

Time (months)	0	1	2	3	4	5
Weight (pounds)	200	195	192	193	190	187

Make a time-series plot of this data

Solution

In R, the command would be:

variable1<-c(type in data with commas in between values, this should be the time variable)
variable2<-c(type in data with commas in between values)
plot(variable1, variable2, ylim=c(0,number over max), main="type in a title you want", xlab="type in a label for the horizontal axis", ylab="type in a label for the vertical axis")
lines(variable1, variable2) – connects the dots

For this example:
time<-c(0, 1, 2, 3, 4, 5)
weight<-c(200, 195, 192, 193, 190, 187)
plot(time, weight, ylim=c(0,250), main="Weight over Time", xlab="Time (Months) ", ylab="Weight (pounds)")
ines(time, weight)

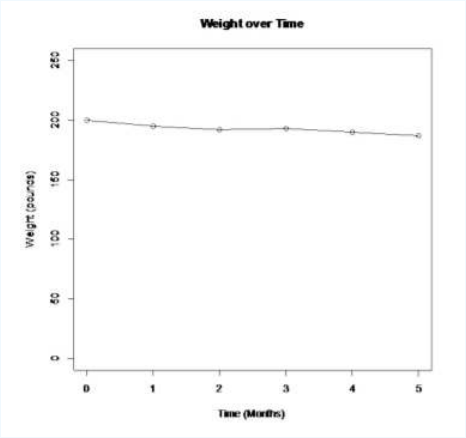
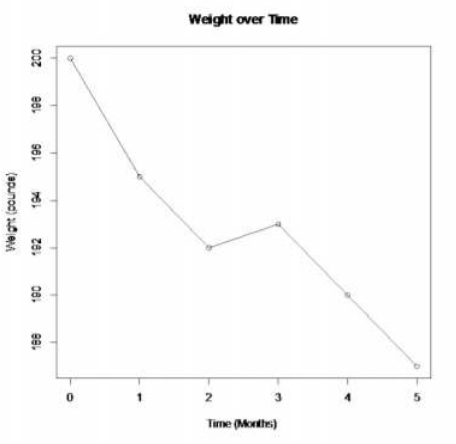


Figure of Weight versus Time

Notice, that over the 5 months, the weight appears to be decreasing. Though it doesn't look like there is a large decrease.

Be careful when making a graph. If you don't start the vertical axis at 0, then the change can look much more dramatic than it really is. As an example, *Graph 2.3.8* shows the *Graph 2.3.7* with a different scaling on the vertical axis. Notice the decrease in weight looks much larger than it really is.



Figure

Homework

1. Students in a statistics class took their first test. The data in Example **3.3.4** are the scores they earned. Create a stem plot.

Table **3.3.4**: Data of Test 1 Grades

80	79	89	74	73	67	79
93	70	70	76	88	83	73
81	79	80	85	79	80	79
58	93	94	74			

2. Students in a statistics class took their first test. The data in Example **3.3.5** are the scores they earned. Create a stem plot. Compare to the graph in question 1.

Table **3.3.5**: Data of Test 1 Grades

67	67	76	47	85	70
87	76	80	72	84	98
84	64	65	82	81	81
88	74	87	83		

3. When an anthropologist finds skeletal remains, they need to figure out the height of the person. The height of a person (in cm) and the length of one of their metacarpal bone (in cm) were collected and are in Example **3.3.6** ("Prediction of height," 2013). Create a scatter plot and state if there is a relationship between the height of a person and the length of their metacarpal.

Table **3.3.6**: Data of Metacarpal versus Height

Length of Metacarpal	Height of Person
45	171
51	178
39	157
41	163
48	172
49	183
46	173
43	175
47	173

4. Example **3.3.7** contains the value of the house and the amount of rental income in a year that the house brings in ("Capital and rental," 2013). Create a scatter plot and state if there is a relationship between the value of the house and the annual rental income.

Table **3.3.7**: Data of House Value versus Rental

Value	Rental	Value	Rental	Value	Rental	Value	Rental
81000	6656	77000	4576	75000	7280	67500	6864
95000	7904	94000	8736	90000	6240	85000	7072
121000	12064	115000	7904	110000	7072	104000	7904
135000	8320	130000	9776	126000	6240	125000	7904
145000	8320	140000	9568	140000	9152	135000	7488
165000	13312	165000	8528	155000	7488	148000	8320
178000	11856	174000	10400	170000	9568	170000	12688
200000	12272	200000	10608	194000	11232	190000	8320
214000	8528	280000	10400	200000	10400	200000	8320
240000	10192	240000	12064	240000	11648	225000	12480
289000	11648	270000	12896	262000	10192	244500	11232
325000	12480	310000	12480	303000	12272	300000	12480

5. The World Bank collects information on the life expectancy of a person in each country ("Life expectancy at," 2013) and the fertility rate per woman in the country ("Fertility rate," 2013). The data for 24 randomly selected countries for the year 2011 are in Example **3.3.8**. Create a scatter plot of the data and state if there appears to be a relationship between life expectancy and the number of births per woman.

Table **3.3.8**: Data of Life Expectancy versus Fertility Rate

Life Expectancy	Fertility Rate	Life Expectancy	Fertility rate
77.2	1.7	72.3	3.9
55.4	5.8	76.0	1.5
69.9	2.2	66.0	4.2
76.4	2.1	5.9	5.2
75.0	1.8	54.4	6.8
78.2	2.0	62.9	4.7
73.0	2.6	78.3	2.1
70.8	2.8	72.1	2.9
82.6	1.4	80.7	1.4
68.9	2.6	74.2	2.5
81.0	1.5	73.3	1.5
54.2	6.9	67.1	2.4

6. The World Bank collected data on the percentage of gross domestic product (GDP) that a country spends on health expenditures ("Health expenditure," 2013) and the percentage of woman receiving prenatal care ("Pregnant woman receiving," 2013). The data for the countries where this information is available for the year 2011 is in Example **3.3.9**. Create a scatter plot of the data and state if there appears to be a relationship between percentage spent on health expenditure and the percentage of woman receiving prenatal care.

Table **3.3.9**: Data of Prenatal Care versus Health Expenditure

Prenatal Care (%)	Health Expenditure (% of GDP)
47.9	9.6
54.6	3.7
93.7	5.2
84.7	5.2
100.0	10.0
42.5	4.7
96.4	4.8
77.1	6.0
58.3	5.4
95.4	4.8
78.0	4.1
93.3	6.0
93.3	9.5
93.7	6.8
89.8	6.1

7. The Australian Institute of Criminology gathered data on the number of deaths (per 100,000 people) due to firearms during the period 1983 to 1997 ("Deaths from firearms," 2013). The data is in Example **3.3.10**. Create a time-series plot of the data and state any findings you can from the graph.

Table **3.3.10**: Data of Year versus Number of Deaths due to Firearms

Year	1983	1984	1985	1986	1987	1988	1989	1990
Rate	4.31	4.42	4.52	4.35	4.39	4.21	3.40	3.61
Year	1991	1992	1993	1994	1995	1996	1997	
Rate	3.67	3.61	2.98	2.95	2.72	2.95	2.3	

8. The economic crisis of 2008 affected many countries, though some more than others. Some people in Australia have claimed that Australia wasn't hurt that badly from the crisis. The bank assets (in billions of Australia dollars (AUD)) of the Reserve Bank of Australia (RBA) for the time period of March 2007 through March 2013 are contained in Example **3.3.11** ("B1 assets of," 2013). Create a time-series plot and interpret any findings.

Table **3.3.11**: Data of Date versus RBA Assets

Date	Assets in Billions of AUD
Mar-2006	96.9
Jun-2006	107.4
Sep-2006	107.2
Dec-2006	116.2
Mar-2007	123.7
Jun-2007	134.0
Sep-2007	123.0
Dec-2007	93.2
Mar-2008	93.7
Jun-2008	105.6
Sep-2008	101.5
Dec-2008	158.8
Mar-2009	118.7
Jun-2009	111.9
Sep-2009	87.0
Dec-2009	86.1
Mar-2010	83.4
Jun-2010	85.7
Sep-2010	74.8
Dec-2010	76.0
Mar-2011	75.7
Jun-2011	75.9
Sep-2011	75.2
Dec-2011	87.9
Mar-2012	91.0
Jun-2012	90.1
Sep-2012	83.9
Dec-2012	95.8
Mar-2013	90.5

9. The consumer price index (CPI) is a measure used by the U.S. government to describe the cost of living. Example **3.3.12** gives the cost of living for the U.S. from the years 1947 through 2011, with the year 1977 being used as the year that all others are compared (DeNavas-Walt, Proctor & Smith, 2012). Create a time-series plot and interpret.

Table **3.3.12**: Data of Time versus CPI

Year	CPI-U-RS1 index (December 1977=100)	Year	CPI-U-RS1 index (December 1977=100)
1947	37.5	1980	127.1
1948	40.5	1981	139.2
1949	40.0	1982	147.6
1950	40.5	1983	153.9
1951	43.7	1984	160.2
1952	44.5	1985	165.7
1953	44.8	1986	168.7
1954	45.2	1987	174.4
1955	45.0	1988	180.8
1956	45.7	1989	188.6
1957	47.2	1990	198.0
1958	48.5	1991	205.1
1959	48.9	1992	210.3
1960	49.7	1993	215.5
1961	50.2	1994	220.1
1962	50.7	1995	225.4
1963	51.4	1996	231.4
1964	52.1	1997	236.4
1965	52.9	1998	239.7
1966	54.4	1999	244.7
1967	56.1	2000	252.9
1968	58.3	2001	260.0
1969	60.9	2002	264.2
1970	63.9	2003	270.1
1971	66.7	2004	277.4
1972	68.7	2005	286.7
1973	73.0	2006	296.1
1974	80.3	2007	304.5
1975	86.9	2008	316.2
1976	91.9	2009	315.0
1977	97.7	2010	320.2
1978	104.4	2011	330.3

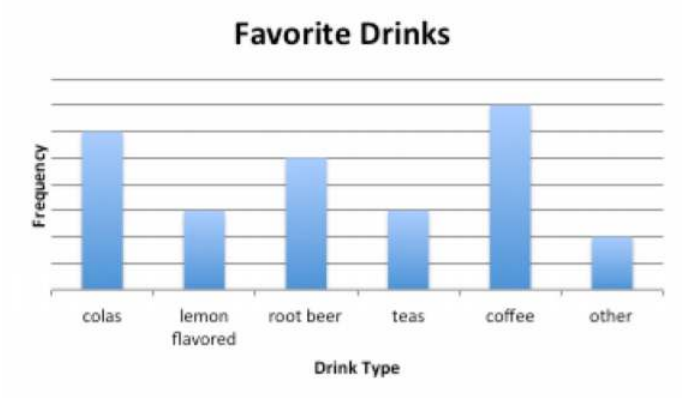
Year	CPI-U-RS1 index (December 1977=100)	Year	CPI-U-RS1 index (December 1977=100)
1979	114.4		

10. The median incomes for all households in the U.S. for the years 1967 to 2011 are given in Example **3.3.13** (DeNavas-Walt, Proctor & Smith, 2012). Create a time-series plot and interpret.

Table **3.3.13**: Data of Time versus Median Income

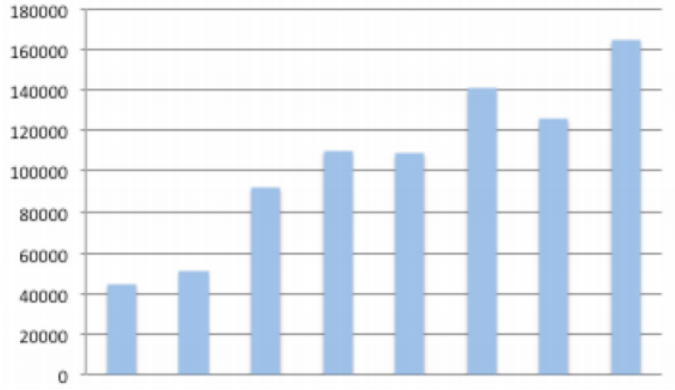
Year	Median Income	Year	Median Income
1967	42,056	1990	49,950
1968	43,868	1991	48,516
1969	45,499	1992	48,117
1970	45,146	1993	47,884
1971	44,707	1994	48,418
1972	46,622	1995	49,935
1973	47,563	1996	50,661
1974	46,057	1997	51,704
1975	44,851	1998	53,582
1976	45,595	1999	54,932
1977	45,884	2000	54,841
1978	47,659	2001	53,646
1979	47,527	2002	53,019
1980	46,024	2003	52,973
1981	45,260	2004	52,788
1982	45,139	2005	53,371
1983	44,823	2006	53,768
1984	46,215	2007	54,489
1985	47,079	2008	52,546
1986	48,746	2009	52,195
1987	49,358	2010	50,831
1988	49,737	2011	50,054
1989	50,624		

11. State everything that makes *Graph 2.3.9* a misleading or poor graph.



***Graph 2.3.9**: Example of a Poor Graph*

12. State everything that makes *Graph 2.3.10* a misleading or poor graph (Benen, 2011).



***Graph 2.3.10**: Example of a Poor Graph*

13. State everything that makes *Graph 2.3.11* a misleading or poor graph ("United States unemployment," 2013).



***Graph 2.3.11**: Example of a Poor Graph*

14. State everything that makes *Graph 2.3.12* a misleading or poor graph.



***Graph 2.3.12**: Example of a Poor Graph*

Answer

See solutions

Data Sources:

B1 assets of financial institutions. (2013, June 27). Retrieved from www.rba.gov.au/statistics/tables/xls/b01hist.xls

Benen, S. (2011, September 02). [Web log message]. Retrieved from http://www.washingtonmonthly.com/pol...edit031960.php

Capital and rental values of Auckland properties. (2013, September 26). Retrieved from http://www.statsci.org/data/oz/rentcap.html

Contraceptive use. (2013, October 9). Retrieved from http://www.prb.org/DataFinder/Topic/...gs.aspx?ind=35

Deaths from firearms. (2013, September 26). Retrieved from http://www.statsci.org/data/oz/firearms.html

DeNavas-Walt, C., Proctor, B., & Smith, J. U.S. Department of Commerce, U.S. Census Bureau. (2012). *Income, poverty, and health insurance coverage in the United States: 2011* (P60-243). Retrieved from website: www.census.gov/prod/2012pubs/p60-243.pdf

Density of people in Africa. (2013, October 9). Retrieved from http://www.prb.org/DataFinder/Topic/...249,250,251,252,253,254,34227,255,257,258,259,260,261,262,263,264,265,266,267,268,269,270,271,272,274,275,276,277,278,279,280,281,282,283,284,285,286,287,288,289,290,291,292,294,295,296,297,298,299,300,301,302,304,305,306,307,308

Department of Health and Human Services, ASPE. (2013). *Health insurance marketplace premiums for 2014*. Retrieved from website: aspe.hhs.gov/health/reports/2...b_premiumsland scape.pdf

Electricity usage. (2013, October 9). Retrieved from http://www.prb.org/DataFinder/Topic/...s.aspx?ind=162

Fertility rate. (2013, October 14). Retrieved from http://data.worldbank.org/indicator/SP.DYN.TFRT.IN

Fuel oil usage. (2013, October 9). Retrieved from http://www.prb.org/DataFinder/Topic/...s.aspx?ind=164

Gas usage. (2013, October 9). Retrieved from http://www.prb.org/DataFinder/Topic/...s.aspx?ind=165

Health expenditure. (2013, October 14). Retrieved from http://data.worldbank.org/indicator/SH.XPD.TOTL.ZS Hinatov, M. U.S. Consumer Product Safety Commission, Directorate of Epidemiology. (2012). *Incidents, deaths, and in-depth investigations associated with non-fire carbon monoxide from engine-driven generators and other engine-driven tools, 1999-2011*. Retrieved from website: www.cpsc.gov/PageFiles/129857/cogenerators.pdf

Life expectancy at birth. (2013, October 14). Retrieved from http://data.worldbank.org/indicator/SP.DYN.LE00.IN

Median income of males. (2013, October 9). Retrieved from http://www.prb.org/DataFinder/Topic/...s.aspx?ind=137

Median income of males. (2013, October 9). Retrieved from http://www.prb.org/DataFinder/Topic/...s.aspx?ind=136

Prediction of height from metacarpal bone length. (2013, September 26). Retrieved from http://www.statsci.org/data/general/stature.html

Pregnant woman receiving prenatal care. (2013, October 14). Retrieved from http://data.worldbank.org/indicator/SH.STA.ANVC.ZS

United States unemployment. (2013, October 14). Retrieved from http://www.tradingeconomics.com/unit...mployment-rate

Weissmann, J. (2013, March 20). A truly devastating graph on state higher education spending. *The Atlantic*. Retrieved from http://www.theatlantic.com/business/...ending/274199/

This page titled 3.3: Other Graphical Representations of Data is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Kathryn Kozak via source content that was edited to the style and standards of the LibreTexts platform.

- **2.3: Other Graphical Representations of Data** by Kathryn Kozak is licensed CC BY-SA 4.0. Original source: https://s3-us-west-2.amazonaws.com/oerfiles/statsusingtech2.pdf.

3.4: Statistical Literacy

Are Commercial Vehicles in Texas Unsafe?

Prerequisites

Graphing Distributions

A news report on the safety of commercial vehicles in Texas stated that one out of five commercial vehicles have been pulled off the road in 2012 because they were unsafe. In addition, 12,301 commercial drivers have been banned from the road for safety violations.

The author presents the bar chart below to provide information about the percentage of fatal crashes involving commercial vehicles in Texas since 2006. The author also quotes DPS director Steven McCraw:

> Commercial vehicles are responsible for approximately 15 percent of the fatalities in Texas crashes. Those who choose to drive unsafe commercial vehicles or drive a commercial vehicle unsafely pose a serious threat to the motoring public.

Example 3.4.1

Based on what you have learned in this chapter, does this bar chart below provide enough information to conclude that unsafe or unsafely driven commercial vehicles pose a serious threat to the motoring public? What might you conclude if 30 percent of all the vehicles on the roads of Texas in 2010 were commercial and accounted for 16 percent of fatal crashes?

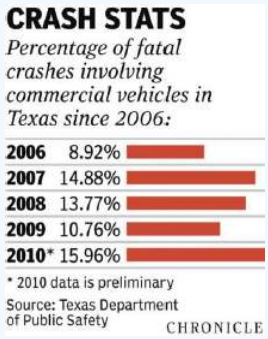


Figure 3.4.1: Crash Statistics for commercial vehicles in Texas

Solution

This bar chart does not provide enough information to draw such a conclusion because we don't know, on the average, in a given year what percentage of all vehicles on the road are commercial vehicles. For example, if 30 percent of all the vehicles on the roads of Texas in 2010 are commercial ones and only 16 percent of fatal crashes involved commercial vehicles, then commercial vehicles are safer than non-commercial ones. Note that in this case 70 percent of vehicles are non-commercial and they are responsible for 84 percent of the fatal crashes.

Linear By Design

Example 3.4.2

Fox News aired the line graph below showing the number unemployed during four quarters between 2007 and 2010.



Figure **3.4.2**: Fox news graph showing job loss by quarter

Does Fox News' line graph provide misleading information? Why or Why not?

Solution:

There are major flaws with the Fox News graph. First, the title of the graph is misleading. Although the data show the number unemployed, Fox News' graph is titled "Job **Loss** by Quarter." Second, the intervals on the X-axis are misleading. Although there are **6** months between September **2008** and March **2009** and **15** months between March **2009** and June **2010**, the intervals are represented in the graph by very similar lengths. This gives the false impression that unemployment increased steadily.

The graph presented below is corrected so that distances on the X-axis are proportional to the number of days between the dates. This graph shows clearly that the rate of increase in the number unemployed is greater between September **2008** and March **2009** than it is between March **2009** and June **2010**.

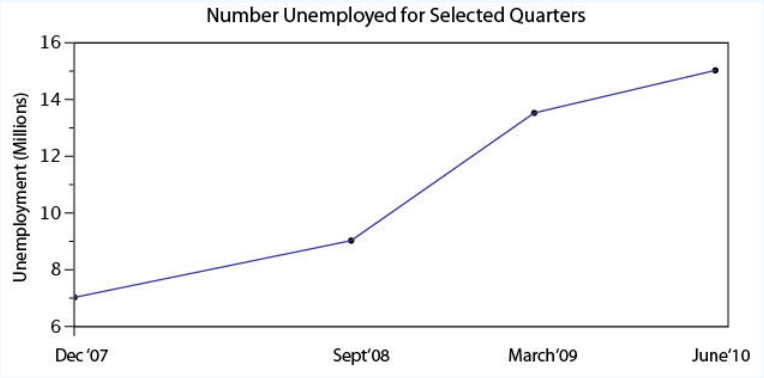


Figure **3.4.3**: Corrected Fox News graph

Contributors and Attributions

- Online Statistics Education: A Multimedia Course of Study (http://onlinestatbook.com/). Project Leader: David M. Lane, Rice University.
- Seyd Ercan and David Lane

This page titled 3.4: Statistical Literacy is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **2.11: Statistical Literacy** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

CHAPTER OVERVIEW

4: Summarizing Data Visually Using R

> *Above all else show the data.*
>
> *–Edward Tufte*[86]

Visualising data is one of the most important tasks facing the data analyst. It's important for two distinct but closely related reasons. Firstly, there's the matter of drawing "presentation graphics": displaying your data in a clean, visually appealing fashion makes it easier for your reader to understand what you're trying to tell them. Equally important, perhaps even more important, is the fact that drawing graphs helps *you* to understand the data. To that end, it's important to draw "exploratory graphics" that help you learn about the data as you go about analysing it. These points might seem pretty obvious, but I cannot count the number of times I've seen people forget them.

```
## Warning: package 'HistData' was built under R version 3.5.2
```

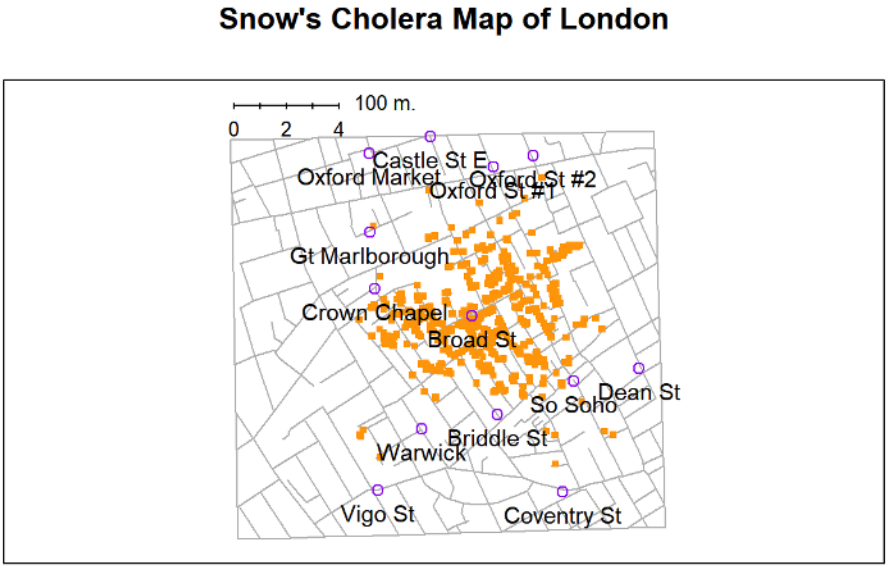


Figure 6.1: A stylised redrawing of John Snow's original cholera map. Each small dot represents the location of a cholera case, and each large circle shows the location of a well. As the plot makes clear, the cholera outbreak is centred very closely on the Broad St pump. This image uses the data from the `HistData` package @[Friendly2011], and was drawn using minor alterations to the commands provided in the help files. Note that Snow's original hand drawn map used different symbols and labels, but you get the idea.

To give a sense of the importance of this chapter, I want to start with a classic illustration of just how powerful a good graph can be. To that end, Figure 6.1 shows a redrawing of one of the most famous data visualisations of all time: John Snow's 1854 map of cholera deaths. The map is elegant in its simplicity. In the background we have a street map, which helps orient the viewer. Over the top, we see a large number of small dots, each one representing the location of a cholera case. The larger symbols show the location of water pumps, labelled by name. Even the most casual inspection of the graph makes it very clear that the source of the outbreak is almost certainly the Broad Street pump. Upon viewing this graph, Dr Snow arranged to have the handle removed from the pump, ending the outbreak that had killed over 500 people. Such is the power of a good data visualisation.

The goals in this chapter are twofold: firstly, to discuss several fairly standard graphs that we use a lot when analysing and presenting data, and secondly, to show you how to create these graphs in R. The graphs themselves tend to be pretty straightforward, so in that respect this chapter is pretty simple. Where people usually struggle is learning how to produce graphs, and especially, learning how to produce good graphs.[87] Fortunately, learning how to draw graphs in R is reasonably simple, as long

as you're not too picky about what your graph looks like. What I mean when I say this is that R has a lot of *very* good graphing functions, and most of the time you can produce a clean, high-quality graphic without having to learn very much about the low-level details of how R handles graphics. Unfortunately, on those occasions when you do want to do something non-standard, or if you need to make highly specific changes to the figure, you actually do need to learn a fair bit about the these details; and those details are both complicated and boring. With that in mind, the structure of this chapter is as follows: I'll start out by giving you a very quick overview of how graphics work in R. I'll then discuss several different kinds of graph and how to draw them, as well as showing the basics of how to customise these plots. I'll then talk in more detail about R graphics, discussing some of those complicated and boring issues. In a future version of this book, I intend to finish this chapter off by talking about what makes a good or a bad graph, but I haven't yet had the time to write that section.

4.1: An Overview of R Graphics
4.2: An Introduction to Plotting
4.3: Histograms
4.4: Stem and Leaf Plots
4.5: Scatterplots
4.6: Bar Graphs
4.7: Saving Image Files Using R and Rstudio
4.8: Summary

This page titled 4: Summarizing Data Visually Using R is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

4.1: An Overview of R Graphics

Reduced to its simplest form, you can think of an R graphic as being much like a painting. You start out with an empty canvas. Every time you use a graphics function, it paints some new things onto your canvas. Later on, you can paint more things over the top if you want; but just like painting, you can't "undo" your strokes. If you make a mistake, you have to throw away your painting and start over. Fortunately, this is way more easy to do when using R than it is when painting a picture in real life: you delete the plot and then type a new set of commands.[88] This way of thinking about drawing graphs is referred to as the ***painter's model***. So far, this probably doesn't sound particularly complicated, and for the vast majority of graphs you'll want to draw it's exactly as simple as it sounds. Much like painting in real life, the headaches usually start when we dig into details. To see why, I'll expand this "painting metaphor" a bit further just to show you the basics of what's going on under the hood, but before I do I want to stress that you really don't need to understand all these complexities in order to draw graphs. I'd been using R for years before I even realised that most of these issues existed! However, I don't want you to go through the same pain I went through every time I inadvertently discovered one of these things, so here's a quick overview.

Firstly, if you want to paint a picture, you need to paint it **on** something. In real life, you can paint on lots of different things. Painting onto canvas isn't the same as painting onto paper, and neither one is the same as painting on a wall. In R, the thing that you paint your graphic onto is called a ***device***. For most applications that we'll look at in this book, this "device" will be a window on your computer. If you're using Windows as your operating system, then the name for this device is `windows`; on a Mac it's called `quartz` because that's the name of the software that the Mac OS uses to draw pretty pictures; and on Linux/Unix, you're probably using `X11`. On the other hand, if you're using Rstudio (regardless of which operating system you're on), there's a separate device called `RStudioGD` that forces R to paint inside the "plots" panel in Rstudio. However, from the computers perspective there's nothing terribly special about drawing pictures on screen: and so R is quite happy to paint pictures directly into a file. R can paint several different types of image files: `jpeg`, `png`, `pdf`, `postscript`, `tiff` and `bmp` files are all among the options that you have available to you. For the most part, these different devices all behave the same way, so you don't really need to know much about the differences between them when learning how to draw pictures. But, just like real life painting, sometimes the specifics do matter. Unless stated otherwise, you can assume that I'm drawing a picture on screen, using the appropriate device (i.e., `windows`, `quartz`, `X11` or `RStudioGD`). One the rare occasions where these behave differently from one another, I'll try to point it out in the text.

Secondly, when you paint a picture you need to paint it **with** something. Maybe you want to do an oil painting, but maybe you want to use watercolour. And, generally speaking, you pretty much have to pick one or the other. The analog to this in R is a "graphics system". A graphics system defines a collection of very ***low-level graphics*** commands about what to draw and where to draw it. Something that surprises most new R users is the discovery that R actually has *two* completely independent graphics systems, known as ***traditional graphics*** (in the `graphics` package) and ***grid graphics*** (in the `grid` package).[89] Not surprisingly, the traditional graphics system is the older of the two: in fact, it's actually older than R since it has it's origins in S, the system from which R is descended. Grid graphics are newer, and in some respects more powerful, so many of the more recent, fancier graphical tools in R make use of grid graphics. However, grid graphics are somewhat more complicated beasts, so most people start out by learning the traditional graphics system. Nevertheless, as long as you don't want to use any low-level commands yourself, then you don't really need to care about whether you're using traditional graphics or grid graphics. However, the moment you do want to tweak your figure by using some low-level commands you do need to care. Because these two different systems are pretty much incompatible with each other, there's a pretty big divide in R graphics universe. Unless stated otherwise, you can assume that everything I'm saying pertains to traditional graphics.

Thirdly, a painting is usually done in a particular **style**. Maybe it's a still life, maybe it's an impressionist piece, or maybe you're trying to annoy me by pretending that cubism is a legitimate artistic style. Regardless, each artistic style imposes some overarching aesthetic and perhaps even constraints on what can (or should) be painted using that style. In the same vein, R has quite a number of different packages, each of which provide a collection of ***high-level graphics*** commands. A single high-level command is capable of drawing an entire graph, complete with a range of customisation options. Most but not all of the high-level commands that I'll talk about in this book come from the `graphics` package itself, and so belong to the world of traditional graphics. These commands all tend to share a common visual style, although there are a few graphics that I'll use that come from other packages that differ in style somewhat. On the other side of the great divide, the grid universe relies heavily on two different

packages – `lattice` and `ggplots2` – each of which provides a quite different visual style. As you've probably guessed, there's a whole separate bunch of functions that you'd need to learn if you want to use `lattice` graphics or make use of the `ggplots2`. However, for the purposes of this book I'll restrict myself to talking about the basic `graphics` tools.

At this point, I think we've covered more than enough background material. The point that I'm trying to make by providing this discussion isn't to scare you with all these horrible details, but rather to try to convey to you the fact that R doesn't really provide a single coherent graphics system. Instead, R itself provides a platform, and different people have built different graphical tools using that platform. As a consequence of this fact, there's two different universes of graphics, and a great multitude of packages that live in them. At this stage you don't need to understand these complexities, but it's useful to know that they're there. But for now, I think we can be happy with a simpler view of things: we'll draw pictures on screen using the traditional graphics system, and as much as possible we'll stick to high level commands only.

So let's start painting.

This page titled 4.1: An Overview of R Graphics is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **6.1: An Overview of R Graphics** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

4.2: An Introduction to Plotting

Before I discuss any specialised graphics, let's start by drawing a few very simple graphs just to get a feel for what it's like to draw pictures using R. To that end, let's create a small vector `Fibonacci` that contains a few numbers we'd like R to draw for us. Then, we'll ask R to `plot()` those numbers:

```
> Fibonacci <- c( 1,1,2,3,5,8,13 )
> plot( Fibonacci )
```

The result is Figure 6.2.

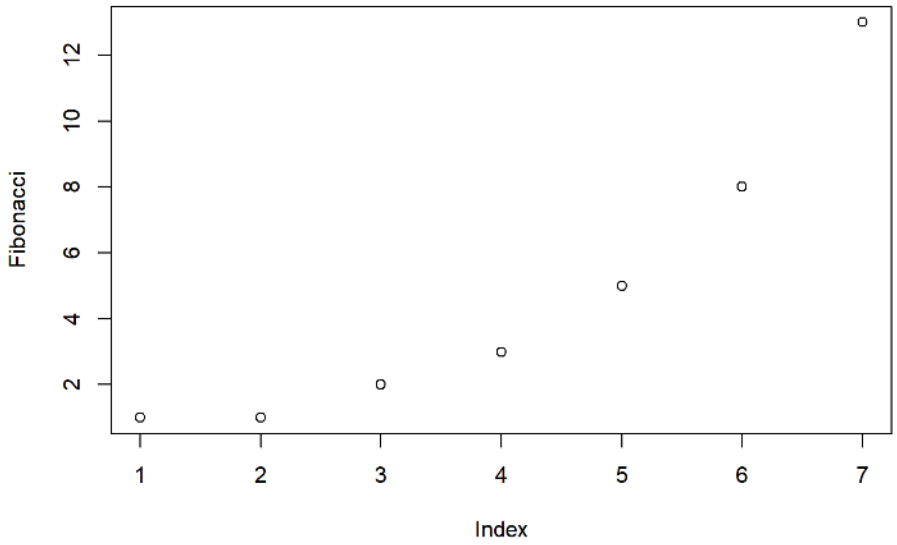


Figure 6.2: Our first plot

As you can see, what R has done is plot the *values* stored in the `Fibonacci` variable on the vertical axis (y-axis) and the corresponding *index* on the horizontal axis (x-axis). In other words, since the 4th element of the vector has a value of 3, we get a dot plotted at the location (4,3). That's pretty straightforward, and the image in Figure 6.2 is probably pretty close to what you would have had in mind when I suggested that we plot the `Fibonacci` data. However, there's quite a lot of customisation options available to you, so we should probably spend a bit of time looking at some of those options. So, be warned: this ends up being a fairly long section, because there's so many possibilities open to you. Don't let it overwhelm you though… while all of the options discussed here are handy to know about, you can get by just fine only knowing a few of them. The only reason I've included all this stuff right at the beginning is that it ends up making the rest of the chapter a lot more readable!

4.2.1 tedious digression

Before we go into any discussion of customising plots, we need a little more background. The important thing to note when using the `plot()` function, is that it's another example of a *generic* function (Section 4.11, much like `print()` and `summary()`, and so its behaviour changes depending on what kind of input you give it. However, the `plot()` function is somewhat fancier than the other two, and its behaviour depends on *two* arguments, `x` (the first input, which is required) and `y` (which is optional). This makes it (a) extremely powerful once you get the hang of it, and (b) hilariously unpredictable, when you're not sure what you're doing. As much as possible, I'll try to make clear what type of inputs produce what kinds of outputs. For now, however, it's enough to note that I'm only doing very basic plotting, and as a consequence all of the work is being done by the `plot.default()` function.

What kinds of customisations might we be interested in? If you look at the help documentation for the default plotting method (i.e., type `?plot.default` or `help("plot.default")`) you'll see a very long list of arguments that you can specify to customise your plot. I'll talk about several of them in a moment, but first I want to point out something that might seem quite wacky. When you look at all the different options that the help file talks about, you'll notice that *some* of the options that it refers to

are “proper” arguments to the `plot.default()` function, but it also goes on to mention a bunch of things that *look* like they’re supposed to be arguments, but they’re not listed in the “Usage” section of the file, and the documentation calls them ***graphical parameters*** instead. Even so, it’s usually possible to treat them as if they were arguments of the plotting function. Very odd. In order to stop my readers trying to find a brick and look up my home address, I’d better explain what’s going on; or at least give the basic gist behind it.

What exactly is a graphical parameter? Basically, the idea is that there are some characteristics of a plot which are pretty universal: for instance, regardless of what kind of graph you’re drawing, you probably need to specify what colour to use for the plot, right? So you’d expect there to be something like a `col` argument to every single graphics function in R? Well, sort of. In order to avoid having hundreds of arguments for every single function, what R does is refer to a bunch of these “graphical parameters” which are pretty general purpose. Graphical parameters can be changed directly by using the low-level `par()` function, which I discuss briefly in Section **??** though not in a lot of detail. If you look at the help files for graphical parameters (i.e., type `?par`) you’ll see that there’s *lots* of them. Fortunately, (a) the default settings are generally pretty good so you can ignore the majority of the parameters, and (b) as you’ll see as we go through this chapter, you very rarely need to use `par()` directly, because you can “pretend” that graphical parameters are just additional arguments to your high-level function (e.g. `plot.default()`). In short… yes, R does have these wacky “graphical parameters” which can be quite confusing. But in most basic uses of the plotting functions, you can act as if they were just undocumented additional arguments to your function.

4.2.2 Customising the title and the axis labels

One of the first things that you’ll find yourself wanting to do when customising your plot is to label it better. You might want to specify more appropriate axis labels, add a title or add a subtitle. The arguments that you need to specify to make this happen are:

- `main`. A character string containing the title.
- `sub`. A character string containing the subtitle.
- `xlab`. A character string containing the x-axis label.
- `ylab`. A character string containing the y-axis label.

These aren’t graphical parameters, they’re arguments to the high-level function. However, because the high-level functions all rely on the same low-level function to do the drawing[90] the names of these arguments are identical for pretty much every high-level function I’ve come across. Let’s have a look at what happens when we make use of all these arguments. Here’s the command…

```
> plot( x = Fibonacci,
+       main = "You specify title using the 'main' argument",
+       sub = "The subtitle appears here! (Use the 'sub' argument for this)",
+       xlab = "The x-axis label is 'xlab'",
+       ylab = "The y-axis label is 'ylab'"
+ )
```

The picture that this draws is shown in Figure 6.3.

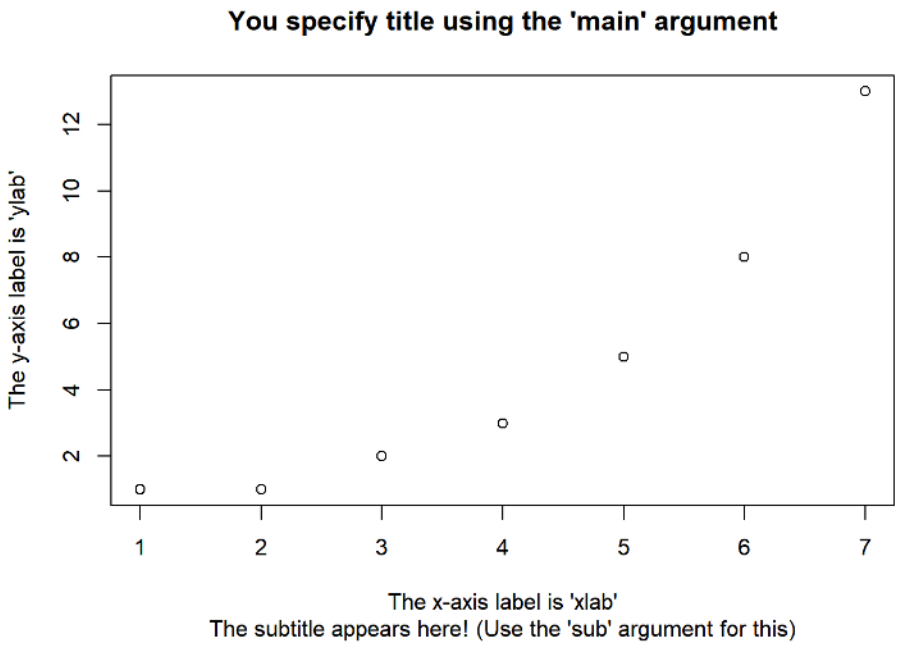


Figure 6.3: How to add your own title, subtitle, x-axis label and y-axis label to the plot.

It's more or less as you'd expect. The plot itself is identical to the one we drew in Figure 6.2, except for the fact that we've changed the axis labels, and added a title and a subtitle. Even so, there's a couple of interesting features worth calling your attention to. Firstly, notice that the subtitle is drawn below the plot, which I personally find annoying; as a consequence I almost never use subtitles. You may have a different opinion, of course, but the important thing is that you remember where the subtitle actually goes. Secondly, notice that R has decided to use boldface text and a larger font size for the title. This is one of my most hated default settings in R graphics, since I feel that it draws too much attention to the title. Generally, while I do want my reader to look at the title, I find that the R defaults are a bit overpowering, so I often like to change the settings. To that end, there are a bunch of graphical parameters that you can use to customise the font style:

- *Font styles:* `font.main`, `font.sub`, `font.lab`, `font.axis`. These four parameters control the font style used for the plot title (`font.main`), the subtitle (`font.sub`), the axis labels (`font.lab`: note that you can't specify separate styles for the x-axis and y-axis without using low level commands), and the numbers next to the tick marks on the axis (`font.axis`). Somewhat irritatingly, these arguments are numbers instead of meaningful names: a value of 1 corresponds to plain text, 2 means boldface, 3 means italic and 4 means bold italic.
- *Font colours:* `col.main`, `col.sub`, `col.lab`, `col.axis`. These parameters do pretty much what the name says: each one specifies a **col**our in which to type each of the different bits of text. Conveniently, R has a very large number of named colours (type `colours()` to see a list of over 650 colour names that R knows), so you can use the English language name of the colour to select it.[91] Thus, the parameter value here string like `"red"`, `"gray25"` or `"springgreen4"` (yes, R really does recognise four different shades of "spring green").
- *Font size:* `cex.main`, `cex.sub`, `cex.lab`, `cex.axis`. Font size is handled in a slightly curious way in R. The "cex" part here is short for "**c**haracter **ex**pansion", and it's essentially a magnification value. By default, all of these are set to a value of 1, except for the font title: `cex.main` has a default magnification of 1.2, which is why the title font is 20% bigger than the others.
- *Font family:* `family`. This argument specifies a font family to use: the simplest way to use it is to set it to `"sans"`, `"serif"`, or `"mono"`, corresponding to a san serif font, a serif font, or a monospaced font. If you want to, you can give the name of a specific font, but keep in mind that different operating systems use different fonts, so it's probably safest to keep it simple. Better yet, unless you have some deep objections to the R defaults, just ignore this parameter entirely. That's what I usually do.

To give you a sense of how you can use these parameters to customise your titles, the following command can be used to draw Figure 6.4:

```
> plot( x = Fibonacci,                          # the data to plot
+       main = "The first 7 Fibonacci numbers",  # the title
+       xlab = "Position in the sequence",       # x-axis label
+       ylab = "The Fibonacci number",           # y-axis label
+       font.main = 1,                           # plain text for title
+       cex.main = 1,                            # normal size for title
+       font.axis = 2,                           # bold text for numbering
+       col.lab = "gray50"                       # grey colour for labels
+ )
```

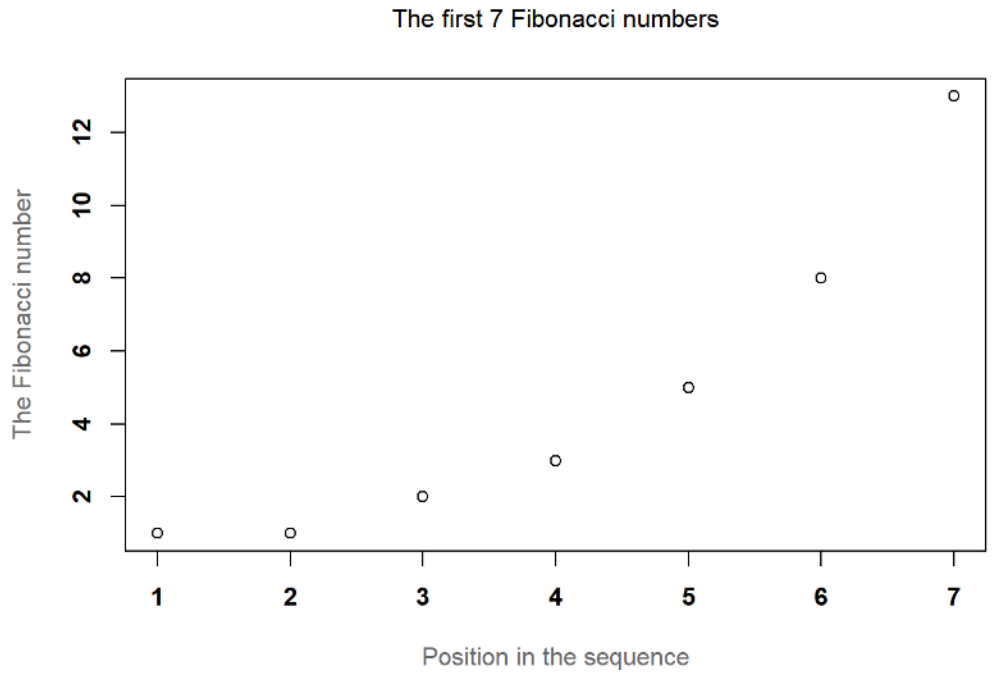


Figure 6.4: How to customise the appearance of the titles and labels.

Although this command is quite long, it's not complicated: all it does is override a bunch of the default parameter values. The only difficult aspect to this is that you have to remember what each of these parameters is called, and what all the different values are. And in practice I never remember: I have to look up the help documentation every time, or else look it up in this book.

4.2.3 Changing the plot type

Adding and customising the titles associated with the plot is one way in which you can play around with what your picture looks like. Another thing that you'll want to do is customise the appearance of the actual plot! To start with, let's look at the single most important options that the `plot()` function (or, recalling that we're dealing with a generic function, in this case the `plot.default()` function, since that's the one doing all the work) provides for you to use, which is the `type` argument. The type argument specifies the visual style of the plot. The possible values for this are:

- `type = "p"` . Draw the **p**oints only.
- `type = "l"` . Draw a **l**ine through the points.
- `type = "o"` . Draw the line **o**ver the top of the points.
- `type = "b"` . Draw **b**oth points and lines, but don't overplot.
- `type = "h"` . Draw "**h**istogram-like" vertical bars.
- `type = "s"` . Draw a **s**taircase, going horizontally then vertically.
- `type = "S"` . Draw a **S**taircase, going vertically then horizontally.
- `type = "c"` . Draw only the **c**onnecting lines from the "b" version.
- `type = "n"` . Draw nothing. (Apparently this is useful sometimes?)

The simplest way to illustrate what each of these really looks like is just to draw them. To that end, Figure 6.5 shows the same Fibonacci data, drawn using six different `types` of plot. As you can see, by altering the type argument you can get a qualitatively different appearance to your plot. In other words, as far as R is concerned, the only difference between a scatterplot (like the ones we drew in Section 5.7 and a line plot is that you draw a scatterplot by setting `type = "p"` and you draw a line plot by setting `type = "l"` . However, that doesn't imply that *you* should think of them as begin equivalent to each other. As you can see by looking at Figure 6.5, a line plot implies that there is some notion of continuity from one point to the next, whereas a scatterplot does not.

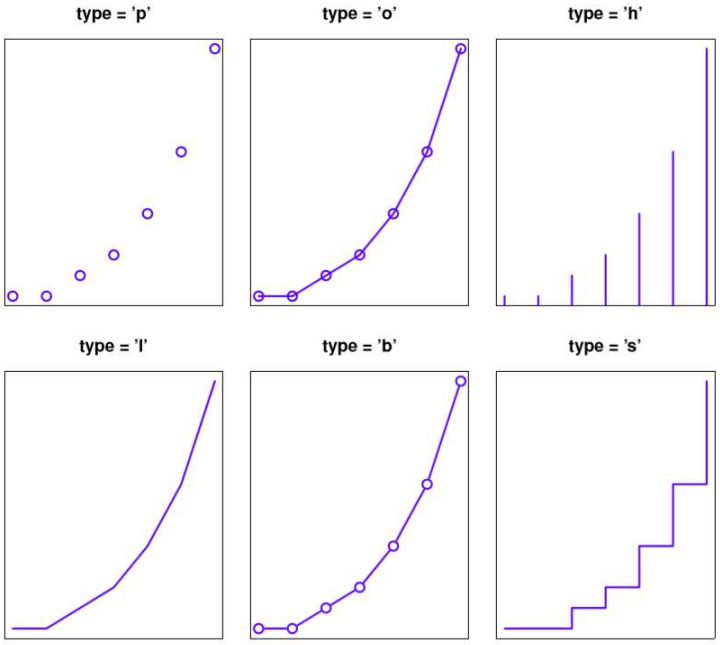


Figure 6.5: Changing the `type` of the plot.

4.2.4 Changing other features of the plot

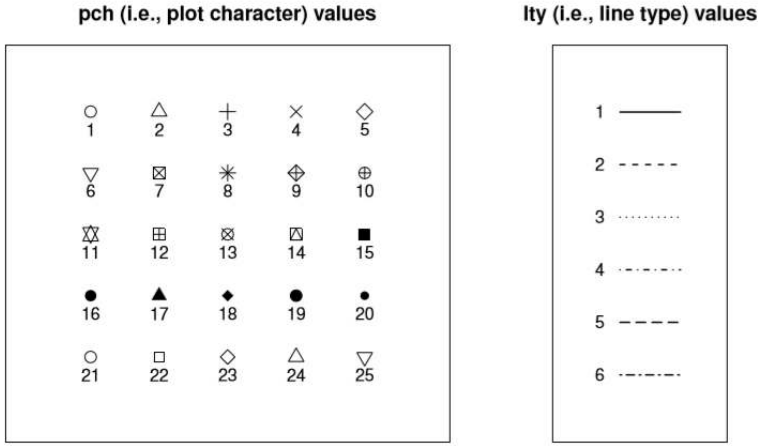


Figure 6.6: Changing the line and plotted characters of the plot.

In Section **??** we talked about a group of graphical parameters that are related to the formatting of titles, axis labels etc. The second group of parameters I want to discuss are those related to the formatting of the plot itself:

- *Colour of the plot*: `col` . As we saw with the previous colour-related parameters, the simplest way to specify this parameter is using a character string: e.g., `col = "blue"` . It's a pretty straightforward parameter to specify: the only real subtlety is that every high-level function tends to draw a different "thing" as it's output, and so this parameter gets interpreted a little

differently by different functions. However, for the `plot.default()` function it's pretty simple: the `col` argument refers to the colour of the points and/or lines that get drawn!

- *Character used to plot points*: `pch` . The **p**lot **ch**aracter parameter is a number, usually between 1 and 25. What it does is tell R what symbol to use to draw the points that it plots. The simplest way to illustrate what the different values do is with a picture. Figure 6.6 a shows the first 25 plotting characters. The default plotting character is a hollow circle (i.e., `pch = 1`).
- *Plot size*: `cex` . This parameter describes a **c**haracter **ex**pansion factor (i.e., magnification) for the plotted characters. By default `cex=1` , but if you want bigger symbols in your graph you should specify a larger value.
- *Line type*: `lty` . The **l**ine **ty**pe parameter describes the kind of line that R draws. It has seven values which you can specify using a number between `0` and `7` , or using a meaningful character string: `"blank"` , `"solid"` , `"dashed"` , `"dotted"` , `"dotdash"` , `"longdash"` , or `"twodash"` . Note that the "blank" version (value 0) just means that R doesn't draw the lines at all. The other six versions are shown in Figure 6.6 b.
- *Line width*: `lwd` . The last graphical parameter in this category that I want to mention is the **l**ine **wi**d**th** parameter, which is just a number specifying the width of the line. The default value is 1. Not surprisingly, larger values produce thicker lines and smaller values produce thinner lines. Try playing around with different values of `lwd` to see what happens.

To illustrate what you can do by altering these parameters, let's try the following command:

```
> plot( x = Fibonacci,   # the data set
+       type = "b",      # plot both points and lines
+       col = "blue",    # change the plot colour to blue
+       pch = 19,        # plotting character is a solid circle
+       cex = 5,         # plot it at 5x the usual size
+       lty = 2,         # change line type to dashed
+       lwd = 4          # change line width to 4x the usual
+ )
```

The output is shown in Figure 6.7.

```
plot( x = Fibonacci,
         type = "b",
         col = "blue",
         pch = 19,
         cex=5,
         lty=2,
         lwd=4)
```

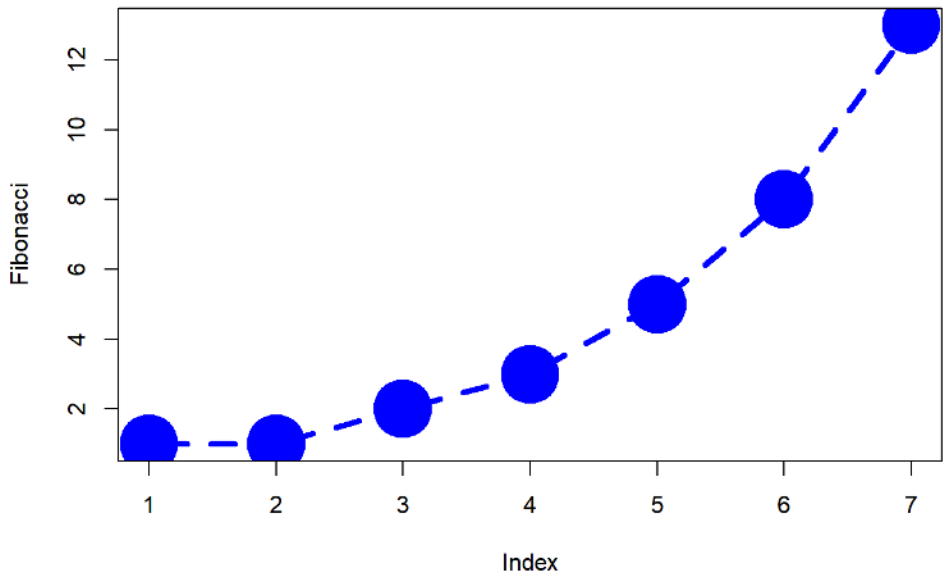


Figure 6.7: Customising various aspects to the plot itself.

4.2.5 Changing the appearance of the axes

There are several other possibilities worth discussing. Ignoring graphical parameters for the moment, there's a few other arguments to the `plot.default()` function that you might want to use. As before, many of these are standard arguments that are used by a lot of high level graphics functions:

- *Changing the axis scales*: `xlim` , `ylim` . Generally R does a pretty good job of figuring out where to set the edges of the plot. However, you can override its choices by setting the `xlim` and `ylim` arguments. For instance, if I decide I want the vertical scale of the plot to run from 0 to 100, then I'd set `ylim = c(0, 100)` .
- *Suppress labelling*: `ann` . This is a logical-valued argument that you can use if you don't want R to include any text for a title, subtitle or axis label. To do so, set `ann = FALSE` . This will stop R from including any text that would normally appear in those places. Note that this will override any of your manual titles. For example, if you try to add a title using the `main` argument, but you also specify `ann = FALSE` , no title will appear.
- *Suppress axis drawing*: `axes` . Again, this is a logical valued argument. Suppose you don't want R to draw any axes at all. To suppress the axes, all you have to do is add `axes = FALSE` . This will remove the axes and the numbering, but not the axis labels (i.e. the `xlab` and `ylab` text). Note that you can get finer grain control over this by specifying the `xaxt` and `yaxt` graphical parameters instead (see below).
- *Include a framing box*: `frame.plot` . Suppose you've removed the axes by setting `axes = FALSE` , but you still want to have a simple box drawn around the plot; that is, you only wanted to get rid of the numbering and the tick marks, but you want to keep the box. To do that, you set `frame.plot = TRUE` .

Note that this list isn't exhaustive. There are a few other arguments to the `plot.default` function that you can play with if you want to, but those are the ones you are probably most likely to want to use. As always, however, if these aren't enough options for you, there's also a number of other graphical parameters that you might want to play with as well. That's the focus of the next section. In the meantime, here's a command that makes use of all these different options:

```
> plot( x = Fibonacci,       # the data
+       xlim = c(0, 15),     # expand the x-scale
+       ylim = c(0, 15),     # expand the y-scale
+       ann = FALSE,         # delete all annotations
+       axes = FALSE,        # delete the axes
+       frame.plot = TRUE    # but include a framing box
+ )
```

The output is shown in Figure 6.8, and it's pretty much exactly as you'd expect. The axis scales on both the horizontal and vertical dimensions have been expanded, the axes have been suppressed as have the annotations, but I've kept a box around the plot.

```
plot( x = Fibonacci,
          xlim = c(0, 15),
          ylim = c(0, 15),
              ann = FALSE,
          axes = FALSE,
            frame.plot = TRUE)
```

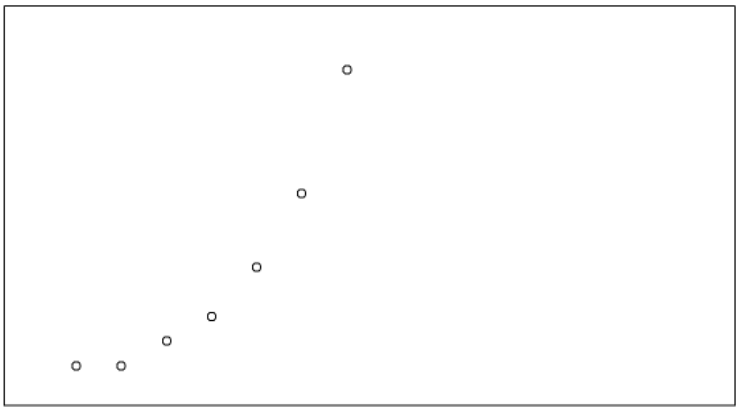


Figure 6.8: Altering the scale and appearance of the plot axes.

Before moving on, I should point out that there are several graphical parameters relating to the axes, the box, and the general appearance of the plot which allow finer grain control over the appearance of the axes and the annotations.

- *Suppressing the axes individually*: `xaxt` , `yaxt` . These graphical parameters are basically just fancier versions of the `axes` argument we discussed earlier. If you want to stop R from drawing the vertical axis but you'd like it to keep the horizontal axis, set `yaxt = "n"` . I trust that you can figure out how to keep the vertical axis and suppress the horizontal one!
- *Box type*: `bty` . In the same way that `xaxt` , `yaxt` are just fancy versions of `axes` , the **b**ox **ty**pe parameter is really just a fancier version of the `frame.plot` argument, allowing you to specify exactly which out of the four borders you want to keep. The way we specify this parameter is a bit stupid, in my opinion: the possible values are `"o"` (the default), `"l"` , `"7"` , `"c"` , `"u"` , or `"]"` , each of which will draw only those edges that the corresponding character suggests. That is, the letter `"c"` has a top, a bottom and a left, but is blank on the right hand side, whereas `"7"` has a top and a right, but is blank on the left and the bottom. Alternatively a value of `"n"` means that no box will be drawn.
- *Orientation of the axis labels* `las` . I presume that the name of this parameter is an acronym of **la**bel **s**tyle or something along those lines; but what it actually does is govern the orientation of the text used to label the individual tick marks (i.e., the numbering, not the `xlab` and `ylab` axis labels). There are four possible values for `las` : A value of 0 means that the labels of both axes are printed parallel to the axis itself (the default). A value of 1 means that the text is always horizontal. A value of 2 means that the labelling text is printed at right angles to the axis. Finally, a value of 3 means that the text is always vertical.

Again, these aren't the only possibilities. There are a few other graphical parameters that I haven't mentioned that you could use to customise the appearance of the axes,[92] but that's probably enough (or more than enough) for now. To give a sense of how you could use these parameters, let's try the following command:

```
> plot( x = Fibonacci,    # the data
+       xaxt = "n",       # don't draw the x-axis
+       bty = "]",        # keep bottom, right and top of box only
+       las = 1           # rotate the text
+ )
```

The output is shown in Figure 6.9. As you can see, this isn't a very useful plot at all. However, it does illustrate the graphical parameters we're talking about, so I suppose it serves its purpose.

```
plot( x = Fibonacci,
        xaxt = "n",
        bty = "]",
        las = 1 )
```

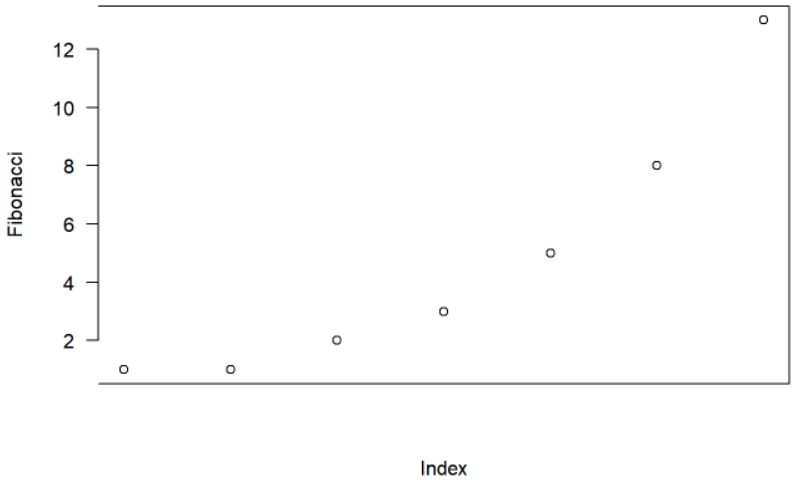


Figure 6.9: Other ways to customise the axes

4.2.6 Don't panic

At this point, a lot of readers will be probably be thinking something along the lines of, "if there's this much detail just for drawing a simple plot, how horrible is it going to get when we start looking at more complicated things?" Perhaps, contrary to my earlier pleas for mercy, you've found a brick to hurl and are right now leafing through an Adelaide phone book trying to find my address. Well, fear not! And please, put the brick down. In a lot of ways, we've gone through the hardest part: we've already covered vast majority of the plot customisations that you might want to do. As you'll see, each of the other high level plotting commands we'll talk about will only have a smallish number of additional options. Better yet, even though I've told you about a billion different ways of tweaking your plot, you don't usually need them. So in practice, now that you've read over it once to get the gist, the majority of the content of this section is stuff you can safely forget: just remember to come back to this section later on when you want to tweak your plot.

This page titled 4.2: An Introduction to Plotting is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **6.2: An Introduction to Plotting** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

4.3: Histograms

Now that we've tamed (or possibly fled from) the beast that is R graphical parameters, let's talk more seriously about some real life graphics that you'll want to draw. We begin with the humble ***histogram***. Histograms are one of the simplest and most useful ways of visualising data. They make most sense when you have an interval or ratio scale (e.g., the `afl.margins` data from Chapter 5 and what you want to do is get an overall impression of the data. Most of you probably know how histograms work, since they're so widely used, but for the sake of completeness I'll describe them. All you do is divide up the possible values into ***bins***, and then count the number of observations that fall within each bin. This count is referred to as the frequency of the bin, and is displayed as a bar: in the AFL winning margins data, there are 33 games in which the winning margin was less than 10 points, and it is this fact that is represented by the height of the leftmost bar in Figure 6.10. Drawing this histogram in R is pretty straightforward. The function you need to use is called `hist()`, and it has pretty reasonable default settings. In fact, Figure 6.10 is exactly what you get if you just type this:

```
> hist( afl.margins )   # panel a
```

```
load("./rbook-master/data/aflsmall.Rdata")
hist(afl.margins)   # panel a
```

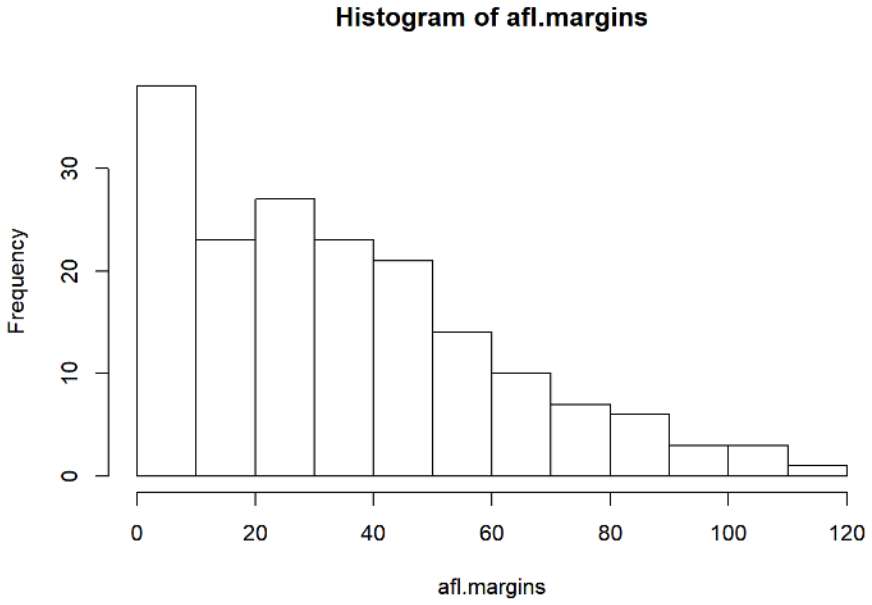


Figure 6.10: The default histogram that R produces

Although this image would need a lot of cleaning up in order to make a good presentation graphic (i.e., one you'd include in a report), it nevertheless does a pretty good job of describing the data. In fact, the big strength of a histogram is that (properly used) it does show the entire spread of the data, so you can get a pretty good sense about what it looks like. The downside to histograms is that they aren't very compact: unlike some of the other plots I'll talk about it's hard to cram 20-30 histograms into a single image without overwhelming the viewer. And of course, if your data are nominal scale (e.g., the `afl.finalists` data) then histograms are useless.

The main subtlety that you need to be aware of when drawing histograms is determining where the `breaks` that separate bins should be located, and (relatedly) how many breaks there should be. In Figure 6.10, you can see that R has made pretty sensible choices all by itself: the breaks are located at 0, 10, 20, … 120, which is exactly what I would have done had I been forced to make a choice myself. On the other hand, consider the two histograms in Figure 6.11 and 6.12, which I produced using the following two commands:

```
hist( x = afl.margins, breaks = 3 )      # panel b
```

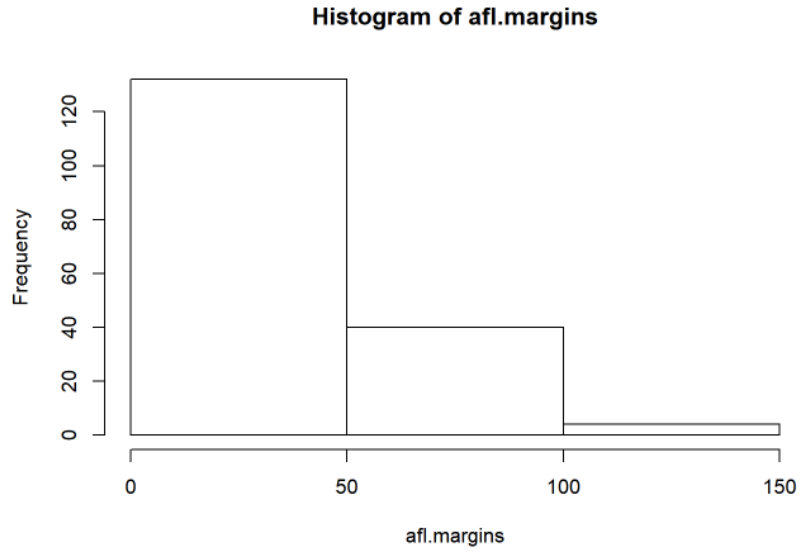


Figure 6.11: A histogram with too few bins

```
hist( x = afl.margins, breaks = 0:116 )  # panel c
```

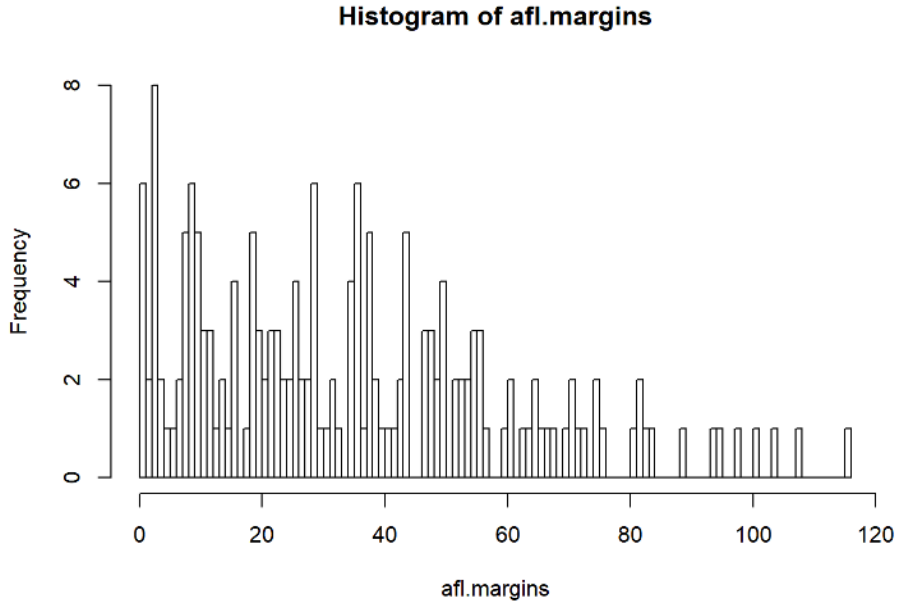


Figure 6.12: A histogram with too many bins

In Figure 6.12, the bins are only 1 point wide. As a result, although the plot is very informative (it displays the entire data set with no loss of information at all!) the plot is very hard to interpret, and feels quite cluttered. On the other hand, the plot in Figure 6.11 has a bin width of 50 points, and has the opposite problem: it's very easy to "read" this plot, but it doesn't convey a lot of information. One gets the sense that this histogram is hiding too much. In short, the way in which you specify the breaks has a big effect on what the histogram looks like, so it's important to make sure you choose the breaks sensibly. In general R does a pretty good job of selecting the breaks on its own, since it makes use of some quite clever tricks that statisticians have devised for automatically selecting the right bins for a histogram, but nevertheless it's usually a good idea to play around with the breaks a bit to see what happens.

There is one fairly important thing to add regarding how the `breaks` argument works. There are two different ways you can specify the breaks. You can either specify *how many* breaks you want (which is what I did for panel b when I typed `breaks = 3`) and let R figure out where they should go, or you can provide a vector that tells R exactly where the breaks should be placed (which is what I did for panel c when I typed `breaks = 0:116`). The behaviour of the `hist()` function is slightly different depending on which version you use. If all you do is tell it *how many* breaks you want, R treats it as a "suggestion" not as a demand. It assumes you want "approximately 3" breaks, but if it doesn't think that this would look very pretty on screen, it picks a different (but similar) number. It does this for a sensible reason – it tries to make sure that the breaks are located at sensible values (like 10) rather than stupid ones (like 7.224414). And most of the time R is right: usually, when a human researcher says "give me 3 breaks", he or she really does mean "give me approximately 3 breaks, and don't put them in stupid places". However, sometimes R is dead wrong. Sometimes you really do mean "exactly 3 breaks", and you know precisely where you want them to go. So you need to invoke "real person privilege", and order R to do what it's bloody well told. In order to do that, you *have* to input the full vector that tells R exactly where you want the breaks. If you do that, R will go back to behaving like the nice little obedient calculator that it's supposed to be.

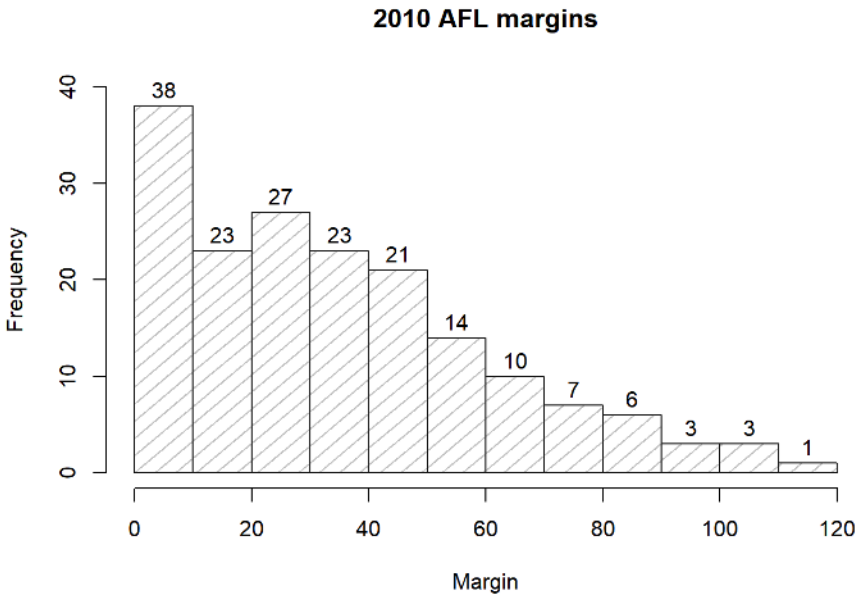
4.3.1 Visual style of your histogram

Okay, so at this point we can draw a basic histogram, and we can alter the number and even the location of the `breaks`. However, the visual style of the histograms shown in Figure @ref(fig:hist1a; hist1b; hist1c) could stand to be improved. We can fix this by making use of some of the other arguments to the `hist()` function. Most of the things you might want to try doing have already been covered in Section 6.2, but there's a few new things:

- *Shading lines*: `density`, `angle`. You can add diagonal lines to shade the bars: the `density` value is a number indicating how many lines per inch R should draw (the default value of `NULL` means no lines), and the `angle` is a number indicating how many degrees from horizontal the lines should be drawn at (default is `angle = 45` degrees).
- *Specifics regarding colours*: `col`, `border`. You can also change the colours: in this instance the `col` parameter sets the colour of the shading (either the shading lines if there are any, or else the colour of the interior of the bars if there are not), and the `border` argument sets the colour of the edges of the bars.
- *Labelling the bars*: `labels`. You can also attach labels to each of the bars using the `labels` argument. The simplest way to do this is to set `labels = TRUE`, in which case R will add a number just above each bar, that number being the exact number of observations in the bin. Alternatively, you can choose the labels yourself, by inputting a vector of strings, e.g., `labels = c("label 1","label 2","etc")`

Not surprisingly, this doesn't exhaust the possibilities. If you type `help("hist")` or `?hist` and have a look at the help documentation for histograms, you'll see a few more options. A histogram that makes use of the histogram-specific customisations as well as several of the options we discussed in Section **??** is shown in Figure **??**. The R command that I used to draw it is this:

```
hist( x = afl.margins,
      main = "2010 AFL margins", # title of the plot
      xlab = "Margin",           # set the x-axis label
      density = 10,              # draw shading lines: 10 per inch
      angle = 40,                # set the angle of the shading lines is 40 degrees
      border = "gray20",         # set the colour of the borders of the bars
      col = "gray80",            # set the colour of the shading lines
      labels = TRUE,             # add frequency labels to each bar
      ylim = c(0,40)             # change the scale of the y-axis
)
```



Overall, this is a much nicer histogram than the default ones.

This page titled 4.3: Histograms is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **6.3: Histograms** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

4.4: Stem and Leaf Plots

Histograms are one of the most widely used methods for displaying the observed values for a variable. They're simple, pretty, and very informative. However, they do take a little bit of effort to draw. Sometimes it can be quite useful to make use of simpler, if less visually appealing, options. One such alternative is the ***stem and leaf plot***. To a first approximation you can think of a stem and leaf plot as a kind of text-based histogram. Stem and leaf plots aren't used as widely these days as they were 30 years ago, since it's now just as easy to draw a histogram as it is to draw a stem and leaf plot. Not only that, they don't work very well for larger data sets. As a consequence you probably won't have as much of a need to use them yourself, though you may run into them in older publications. These days, the only real world situation where I use them is if I have a small data set with 20-30 data points and I don't have a computer handy, because it's pretty easy to quickly sketch a stem and leaf plot by hand.

With all that as background, lets have a look at stem and leaf plots. The AFL margins data contains 176 observations, which is at the upper end for what you can realistically plot this way. The function in R for drawing stem and leaf plots is called `stem()` and if we ask for a stem and leaf plot of the `afl.margins` data, here's what we get:

```
stem( afl.margins )
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##    0 | 00111122333333334456778888899999
##    1 | 00000111222344566668999 9
##    2 | 000112223334455666677889999 9
##    3 | 01223555566666678888899
##    4 | 012334444477788899
##    5 | 00002233445556667
##    6 | 0113455678
##    7 | 01123556
##    8 | 122349
##    9 | 458
##   10 | 148
##   11 | 6
```

The values to the left of the `|` are called ***stems*** and the values to the right are called ***leaves***. If you just look at the shape that the leaves make, you can see something that looks a lot like a histogram made out of numbers, just rotated by 90 degrees. But if you know how to read the plot, there's quite a lot of additional information here. In fact, it's also giving you the actual values of *all* of the observations in the data set. To illustrate, let's have a look at the last line in the stem and leaf plot, namely `11 | 6` . Specifically, let's compare this to the largest values of the `afl.margins` data set:

```
> max( afl.margins )
[1] 116
```

Hm… `11 | 6` versus `116` . Obviously the stem and leaf plot is trying to tell us that the largest value in the data set is 116. Similarly, when we look at the line that reads `10 | 148` , the way we interpret it to note that the stem and leaf plot is telling us that the data set contains observations with values 101, 104 and 108. Finally, when we see something like `5 | 00002233445556667` the four `0` s in the the stem and leaf plot are telling us that there are four observations with value 50.

I won't talk about them in a lot of detail, but I should point out that some customisation options are available for stem and leaf plots in R. The two arguments that you can use to do this are:

- `scale` . Changing the `scale` of the plot (default value is 1), which is analogous to changing the number of breaks in a histogram. Reducing the scale causes R to reduce the number of stem values (i.e., the number of breaks, if this were a histogram) that the plot uses.
- `width` . The second way that to can customise a stem and leaf plot is to alter the `width` (default value is 80). Changing the width alters the maximum number of leaf values that can be displayed for any given stem.

However, since stem and leaf plots aren't as important as they used to be, I'll leave it to the interested reader to investigate these options. Try the following two commands to see what happens:

```
> stem( x = afl.margins, scale = .25 )
> stem( x = afl.margins, width = 20 )
```

The only other thing to note about stem and leaf plots is the line in which R tells you where the decimal point is. If our data set had included only the numbers .11, .15, .23, .35 and .59 and we'd drawn a stem and leaf plot of these data, then R would move the decimal point: the stem values would be 1,2,3,4 and 5, but R would tell you that the decimal point has moved to the left of the `|` symbol. If you want to see this in action, try the following command:

```
> stem( x = afl.margins / 1000 )
```

The stem and leaf plot itself will look identical to the original one we drew, except for the fact that R will tell you that the decimal point has moved.

This page titled 4.4: Stem and Leaf Plots is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **6.4: Stem and Leaf Plots** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

4.5: Scatterplots

Scatterplots are a simple but effective tool for visualising data. We've already seen scatterplots in this chapter, when using the `plot()` function to draw the `Fibonacci` variable as a collection of dots (Section 6.2. However, for the purposes of this section I have a slightly different notion in mind. Instead of just plotting one variable, what I want to do with my scatterplot is display the relationship between *two* variables, like we saw with the figures in the section on correlation (Section 5.7. It's this latter application that we usually have in mind when we use the term "scatterplot". In this kind of plot, each observation corresponds to one dot: the horizontal location of the dot plots the value of the observation on one variable, and the vertical location displays its value on the other variable. In many situations you don't really have a clear opinions about what the *causal* relationship is (e.g., does A cause B, or does B cause A, or does some other variable C control both A and B). If that's the case, it doesn't really matter which variable you plot on the x-axis and which one you plot on the y-axis. However, in many situations you do have a pretty strong idea which variable you think is most likely to be causal, or at least you have some suspicions in that direction. If so, then it's conventional to plot the cause variable on the x-axis, and the effect variable on the y-axis. With that in mind, let's look at how to draw scatterplots in R, using the same `parenthood` data set (i.e. `parenthood.Rdata`) that I used when introducing the idea of correlations.

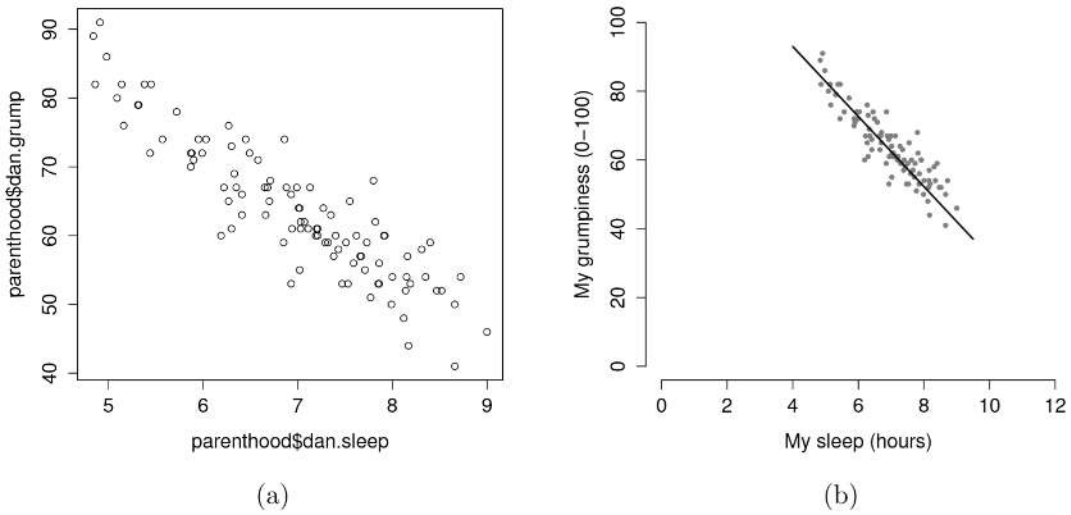


Figure 6.18: {Two different scatterplots: (a) the default scatterplot that R produces, (b) one that makes use of several options for fancier display.

Suppose my goal is to draw a scatterplot displaying the relationship between the amount of sleep that I get (`dan.sleep`) and how grumpy I am the next day (`dan.grump`). As you might expect given our earlier use of `plot()` to display the `Fibonacci` data, the function that we use is the `plot()` function, but because it's a generic function all the hard work is still being done by the `plot.default()` function. In any case, there are two different ways in which we can get the plot that we're after. The first way is to specify the name of the variable to be plotted on the `x` axis and the variable to be plotted on the `y` axis. When we do it this way, the command looks like this:

```
plot( x = parenthood$dan.sleep,   # data on the x-axis
      y = parenthood$dan.grump    # data on the y-axis
 )
```

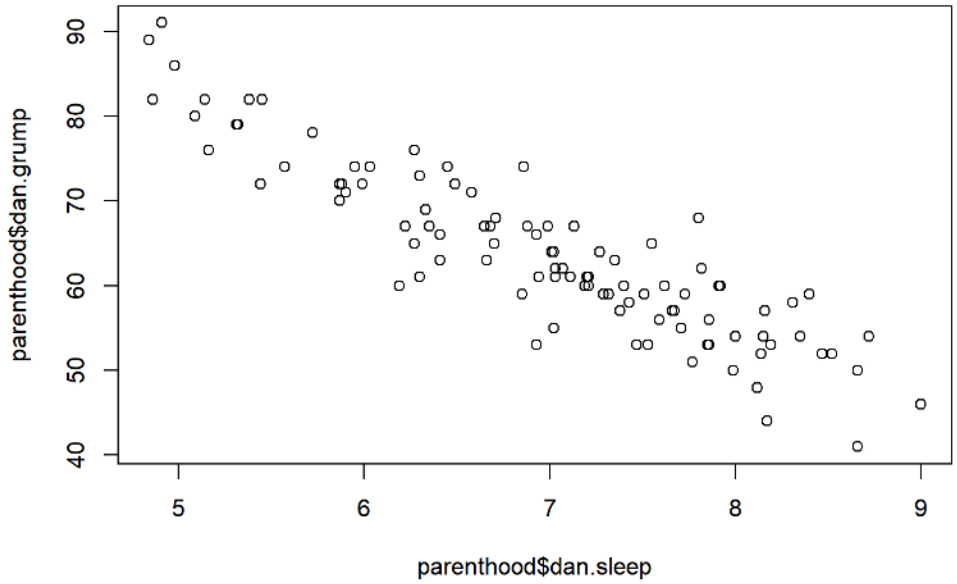


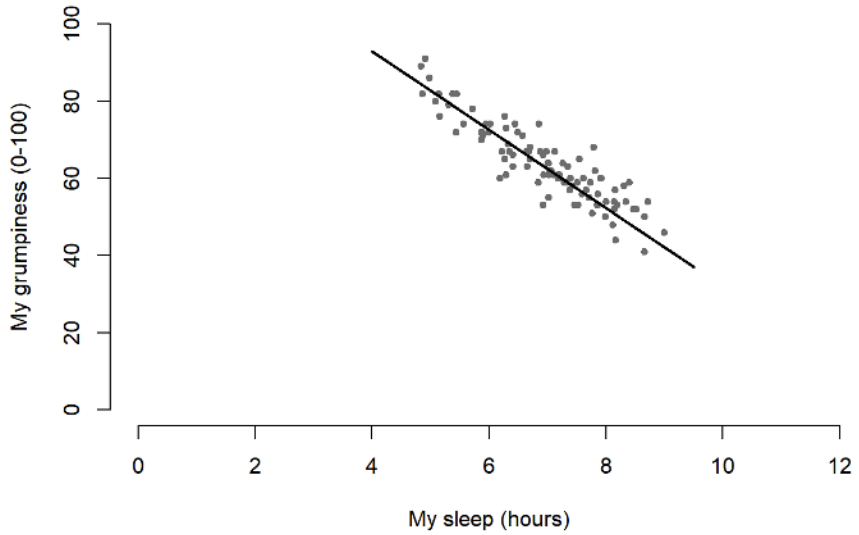
Figure 6.19: the default scatterplot that R produces

The second way do to it is to use a "formula and data frame" format, but I'm going to avoid using it.[99] For now, let's just stick with the `x` and `y` version. If we do this, the result is the very basic scatterplot shown in Figure 6.19. This serves fairly well, but there's a few customisations that we probably want to make in order to have this work properly. As usual, we want to add some labels, but there's a few other things we might want to do as well. Firstly, it's sometimes useful to rescale the plots. In Figure 6.19 R has selected the scales so that the data fall neatly in the middle. But, in this case, we happen to know that the grumpiness measure falls on a scale from 0 to 100, and the hours slept falls on a natural scale between 0 hours and about 12 or so hours (the longest I can sleep in real life). So the command I might use to draw this is:

```
plot( x = parenthood$dan.sleep,          # data on the x-axis
      y = parenthood$dan.grump,          # data on the y-axis
      xlab = "My sleep (hours)",         # x-axis label
      ylab = "My grumpiness (0-100)",    # y-axis label
      xlim = c(0,12),                    # scale the x-axis
      ylim = c(0,100),                   # scale the y-axis
      pch = 20,                          # change the plot type
      col = "gray50",                    # dim the dots slightly
      frame.plot = FALSE                 # don't draw a box
 )
```

This command produces the scatterplot in Figure **??**, or at least very nearly. What it doesn't do is draw the line through the middle of the points. Sometimes it can be very useful to do this, and I can do so using `lines()`, which is a low level plotting function. Better yet, the arguments that I need to specify are pretty much the exact same ones that I use when calling the `plot()` function. That is, suppose that I want to draw a line that goes from the point (4,93) to the point (9.5,37). Then the `x` locations can be specified by the vector `c(4,9.5)` and the `y` locations correspond to the vector `c(93,37)`. In other words, I use this command:

```
plot( x = parenthood$dan.sleep,          # data on the x-axis
      y = parenthood$dan.grump,          # data on the y-axis
      xlab = "My sleep (hours)",         # x-axis label
      ylab = "My grumpiness (0-100)",    # y-axis label
      xlim = c(0,12),                    # scale the x-axis
      ylim = c(0,100),                   # scale the y-axis
      pch = 20,                          # change the plot type
      col = "gray50",                    # dim the dots slightly
      frame.plot = FALSE                 # don't draw a box
 )
 lines( x = c(4,9.5),   # the horizontal locations
        y = c(93,37),   # the vertical locations
        lwd = 2         # line width
 )
```



And when I do so, R plots the line over the top of the plot that I drew using the previous command. In most realistic data analysis situations you absolutely don't want to just guess where the line through the points goes, since there's about a billion different ways in which you can get R to do a better job. However, it does at least illustrate the basic idea.

One possibility, if you do want to get R to draw nice clean lines through the data for you, is to use the `scatterplot()` function in the `car` package. Before we can use `scatterplot()` we need to load the package:

```
> library( car )
```

Having done so, we can now use the function. The command we need is this one:

```
> scatterplot( dan.grump ~ dan.sleep,
+              data = parenthood,
+              smooth = FALSE
+ )
```

```
## Loading required package: carData
```

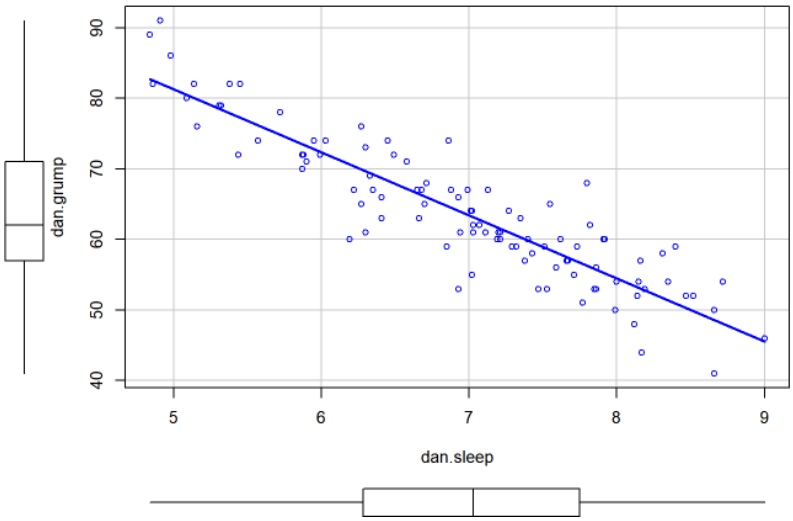


Figure 6.20: A fancy scatterplot drawn using the `scatterplot()` function in the `car` package.

The first two arguments should be familiar: the first input is a formula `dan.grump ~ dan.sleep` telling R what variables to plot,[100] and the second specifies a `data` frame. The third argument `smooth` I've set to `FALSE` to stop the `scatterplot()` function from drawing a fancy "smoothed" trendline (since it's a bit confusing to beginners). The scatterplot itself is shown in Figure 6.20. As you can see, it's not only drawn the scatterplot, but its also drawn boxplots for each of the two variables, as well as a simple line of best fit showing the relationship between the two variables.

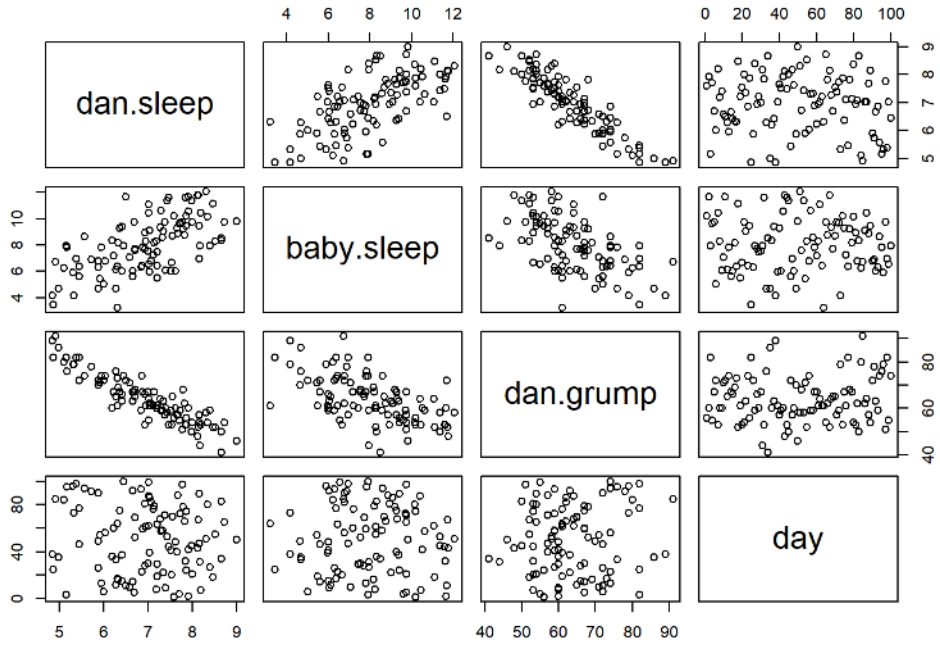
4.5.1 More elaborate options

Often you find yourself wanting to look at the relationships between several variables at once. One useful tool for doing so is to produce a ***scatterplot matrix***, analogous to the correlation matrix.

```
> cor( x = parenthood ) # calculate correlation matrix
             dan.sleep  baby.sleep   dan.grump         day
dan.sleep   1.00000000  0.62794934 -0.90338404 -0.09840768
baby.sleep  0.62794934  1.00000000 -0.56596373 -0.01043394
dan.grump  -0.90338404 -0.56596373  1.00000000  0.07647926
day        -0.09840768 -0.01043394  0.07647926  1.00000000
```

We can get a the corresponding scatterplot matrix by using the `pairs()` function:[101]

```
pairs( x = parenthood ) # draw corresponding scatterplot matrix
```



The output of the `pairs()` command is shown in Figure **??**. An alternative way of calling the `pairs()` function, which can be useful in some situations, is to specify the variables to include using a one-sided formula. For instance, this

```
> pairs( formula = ~ dan.sleep + baby.sleep + dan.grump,
+        data = parenthood
+ )
```

would produce a 3×3 scatterplot matrix that only compare `dan.sleep`, `dan.grump` and `baby.sleep`. Obviously, the first version is much easier, but there are cases where you really only want to look at a few of the variables, so it's nice to use the formula interface.

This page titled 4.5: Scatterplots is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **6.6: Scatterplots** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

4.6: Bar Graphs

Another form of graph that you often want to plot is the ***bar graph***. The main function that you can use in R to draw them is the `barplot()` function.[102] And to illustrate the use of the function, I'll use the `finalists` variable that I introduced in Section 5.1.7. What I want to do is draw a bar graph that displays the number of finals that each team has played in over the time spanned by the `afl` data set. So, let's start by creating a vector that contains this information. I'll use the `tabulate()` function to do this (which will be discussed properly in Section **??**, since it creates a simple numeric vector:

```
> freq <- tabulate( afl.finalists )
> print( freq )
 [1] 26 25 26 28 32  0  6 39 27 28 28 17  6 24 26 39 24
```

This isn't exactly the prettiest of frequency tables, of course. I'm only doing it this way so that you can see the `barplot()` function in it's "purest" form: when the input is just an ordinary numeric vector. That being said, I'm obviously going to need the team names to create some labels, so let's create a variable with those. I'll do this using the `levels()` function, which outputs the names of all the levels of a factor (see Section 4.7:

```
> teams <- levels( afl.finalists )
> print( teams )
 [1] "Adelaide"         "Brisbane"         "Carlton"          "Collingwood"
 [5] "Essendon"         "Fitzroy"          "Fremantle"        "Geelong"
 [9] "Hawthorn"         "Melbourne"        "North Melbourne"  "Port Adelaide"
[13] "Richmond"         "St Kilda"         "Sydney"           "West Coast"
[17] "Western Bulldogs"
```

Okay, so now that we have the information we need, let's draw our bar graph. The main argument that you need to specify for a bar graph is the `height` of the bars, which in our case correspond to the values stored in the `freq` variable:

```
> barplot( height = freq )  # specifying the argument name (panel a)
> barplot( freq )   # the lazier version (panel a)
```

Either of these two commands will produce the simple bar graph shown in Figure 6.21.

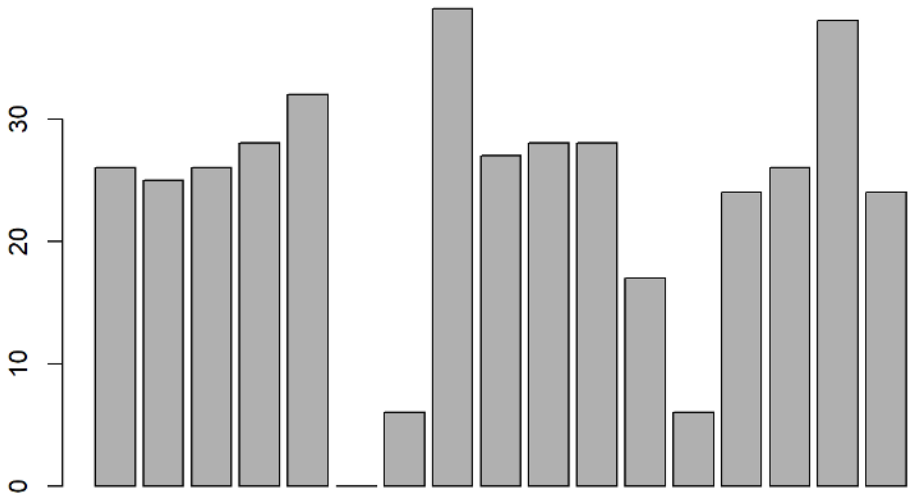


Figure 6.21: the simplest version of a bargraph, containing the data but no labels

As you can see, R has drawn a pretty minimal plot. It doesn't have any labels, obviously, because we didn't actually tell the `barplot()` function what the labels are! To do this, we need to specify the `names.arg` argument. The `names.arg`

argument needs to be a vector of character strings containing the text that needs to be used as the label for each of the items. In this case, the `teams` vector is exactly what we need, so the command we're looking for is:

```
barplot( height = freq, names.arg = teams )
```

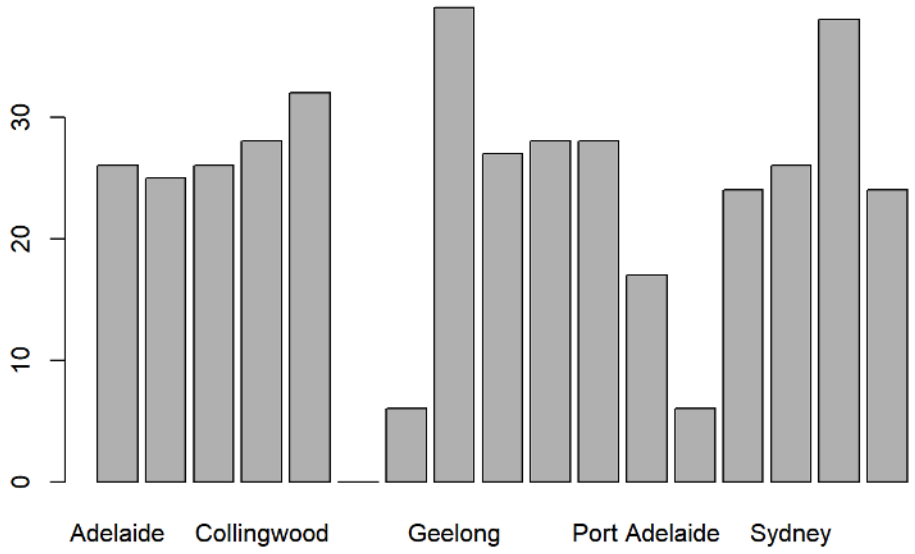


Figure 6.22: we've added the labels, but because the text runs horizontally R only includes a few of them

This is an improvement, but not much of an improvement. R has only included a few of the labels, because it can't fit them in the plot. This is the same behaviour we saw earlier with the multiple-boxplot graph in Figure 6.16. However, in Figure 6.16 it wasn't an issue: it's pretty obvious from inspection that the two unlabelled plots in between 1987 and 1990 must correspond to the data from 1988 and 1989. However, the fact that `barplot()` has omitted the names of every team in between Adelaide and Fitzroy is a lot more problematic.

The simplest way to fix this is to rotate the labels, so that the text runs vertically not horizontally. To do this, we need to alter set the `las` parameter, which I discussed briefly in Section **??**. What I'll do is tell R to rotate the text so that it's always perpendicular to the axes (i.e., I'll set `las = 2`). When I do that, as per the following command…

```
barplot(height = freq,  # the frequencies
        names.arg = teams,  # the label
        las = 2)            # rotate the labels
```

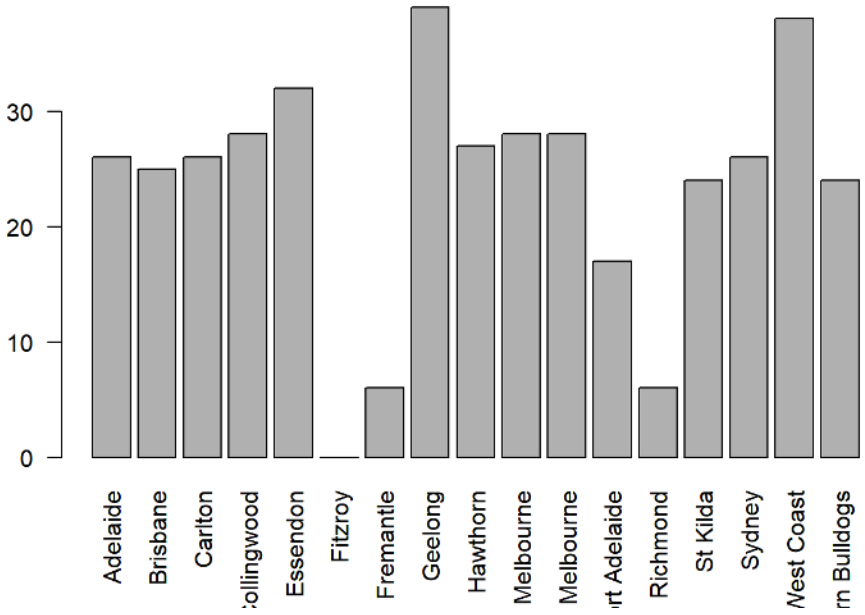


Figure 6.23: we've rotated the labels, but now the text is too long to fit

… the result is the bar graph shown in Figure 6.23. We've fixed the problem, but we've created a new one: the axis labels don't quite fit anymore. To fix this, we have to be a bit cleverer again. A simple fix would be to use shorter names rather than the full name of all teams, and in many situations that's probably the right thing to do. However, at other times you really do need to create a bit more space to add your labels, so I'll show you how to do that.

4.6.1 Changing global settings using `par()`

Altering the margins to the plot is actually a somewhat more complicated exercise than you might think. In principle it's a very simple thing to do: the size of the margins is governed by a graphical parameter called `mar`, so all we need to do is alter this parameter. First, let's look at what the `mar` argument specifies. The `mar` argument is a vector containing four numbers: specifying the amount of space at the bottom, the left, the top and then the right. The units are "number of `lines'". The default value for` mar `is` c(5.1, 4.1, 4.1, 2.1)`, meaning that R leaves 5.1"lines" empty at the bottom, 4.1 lines on the left and the bottom, and only 2.1 lines on the right. In order to make more room at the bottom, what I need to do is change the first of these numbers. A value of 10.1 should do the trick.

So far this doesn't seem any different to the other graphical parameters that we've talked about. However, because of the way that the traditional graphics system in R works, you need to specify what the margins will be *before* calling your high-level plotting function. Unlike the other cases we've see, you can't treat `mar` as if it were just another argument in your plotting function. Instead, you have to use the `par()` function to change the graphical parameters beforehand, and only then try to draw your figure. In other words, the first thing I would do is this:

```
> par( mar = c( 10.1, 4.1, 4.1, 2.1) )
```

There's no visible output here, but behind the scenes R has changed the graphical parameters associated with the current device (remember, in R terminology all graphics are drawn onto a "device"). Now that this is done, we could use the exact same command as before, but this time you'd see that the labels all fit, because R now leaves twice as much room for the labels at the bottom. However, since I've now figured out how to get the labels to display properly, I might as well play around with some of the other options, all of which are things you've seen before:

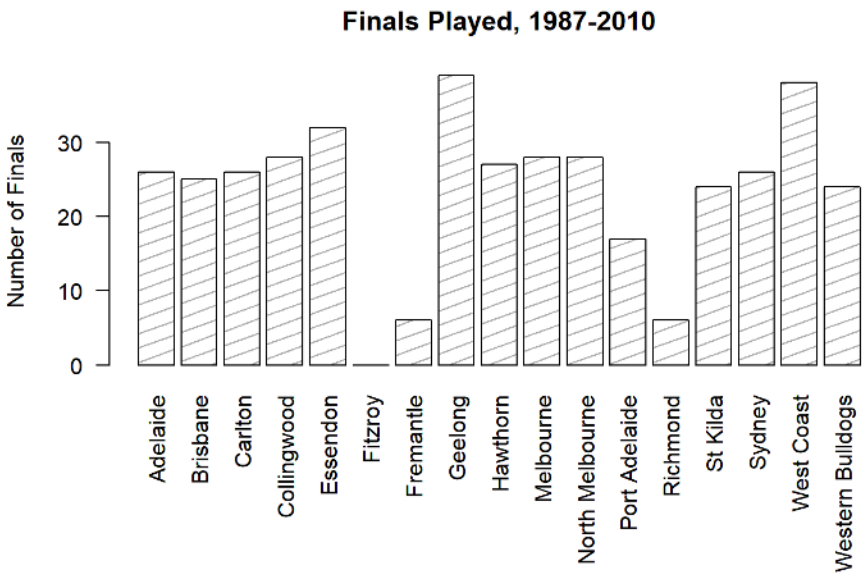


Figure 6.24: we fix this by expanding the margin at the bottom, and add several other customisations to make the chart a bit nicer

```
barplot( height = freq,
        names.arg = teams,
        las=2,
        ylab = "Number of Finals",
        main = "Finals Played, 1987-2010",
        density = 10,
        angle = 20)
```

However, one thing to remember about the `par()` function is that it doesn't just change the graphical parameters for the current *plot*. Rather, the changes pertain to any subsequent plot that you draw onto the same *device*. This might be exactly what you want, in which case there's no problem. But if not, you need to reset the graphical parameters to their original settings. To do this, you can either close the device (e.g., close the window, or click the "Clear All" button in the Plots panel in Rstudio) or you can reset the graphical parameters to their original values, using a command like this:

```
> par( mar = c(5.1, 4.1, 4.1, 2.1) )
```

This page titled 4.6: Bar Graphs is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **6.7: Bar Graphs** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

4.7: Saving Image Files Using R and Rstudio

Hold on, you might be thinking. What's the good of being able to draw pretty pictures in R if I can't save them and send them to friends to brag about how awesome my data is? How do I save the picture? This is another one of those situations where the easiest thing to do is to use the RStudio tools.

If you're running R through Rstudio, then the easiest way to save your image is to click on the "Export" button in the Plot panel (i.e., the area in Rstudio where all the plots have been appearing). When you do that you'll see a menu that contains the options "Save Plot as PDF" and "Save Plot as Image". Either version works. Both will bring up dialog boxes that give you a few options that you can play with, but besides that it's pretty simple.

This works pretty nicely for most situations. So, unless you're filled with a burning desire to learn the low level details, feel free to skip the rest of this section.

4.7.1 ugly details (advanced)

As I say, the menu-based options should be good enough for most people most of the time. However, one day you might want to be a bit more sophisticated, and make use of R's image writing capabilities at a lower level. In this section I'll give you a very basic introduction to this. In all honesty, this barely scratches the surface, but it will help a little bit in getting you started if you want to learn the details.

Okay, as I hinted earlier, whenever you're drawing pictures in R you're deemed to be drawing *to* a device of some kind. There are devices that correspond to a figure drawn on screen, and there are devices that correspond to graphics files that R will produce for you. For the purposes of this section I'll assume that you're using the default application in either Windows or Mac OS, not Rstudio. The reason for this is that my experience with the graphical device provided by Rstudio has led me to suspect that it still has a bunch on non-standard (or possibly just undocumented) features, and so I don't quite trust that it always does what I expect. I've no doubt they'll smooth it out later, but I can honestly say that I don't quite get what's going on with the `RStudioGD` device. In any case, we can ask R to list all of the graphics devices that currently exist, simply by using the command `dev.list()`. If there are no figure windows open, then you'll see this:

```
> dev.list()
NULL
```

which just means that R doesn't have any graphics devices open. However, suppose if you've just drawn a histogram and you type the same command, R will now give you a different answer. For instance, if you're using Windows:

```
> hist( afl.margins )
> dev.list()
windows
      2
```

What this means is that there is one graphics device (device 2) that is currently open, and it's a figure window. If you did the same thing on a Mac, you get basically the same answer, except that the name of the device would be `quartz` rather than `windows`. If you had several graphics windows open (which, incidentally, you can do by using the `dev.new()` command) then you'd see something like this:

```
> dev.list()
windows windows windows
      2       3       4
```

Okay, so that's the basic idea behind graphics devices. The key idea here is that graphics files (like JPEG images etc) are *also* graphics devices as far as R is concerned. So what you want to do is to *copy* the contents of one graphics device to another one.

There's a command called `dev.copy()` that does this, but what I'll explain to you is a simpler one called `dev.print()`. It's pretty simple:

```
> dev.print( device = jpeg,              # what are we printing to?
+            filename = "thisfile.jpg",  # name of the image file
+            width = 480,                # how many pixels wide should it be
+            height = 300                # how many pixels high should it be
+ )
```

This takes the "active" figure window, copies it to a jpeg file (which R treats as a device) and then closes that device. The `filename = "thisfile.jpg"` part tells R what to name the graphics file, and the `width = 480` and `height = 300` arguments tell R to draw an image that is 300 pixels high and 480 pixels wide. If you want a different kind of file, just change the device argument from `jpeg` to something else. R has devices for `png`, `tiff` and `bmp` that all work in exactly the same way as the `jpeg` command, but produce different kinds of files. Actually, for simple cartoonish graphics like this histogram, you'd be better advised to use PNG or TIFF over JPEG. The JPEG format is very good for natural images, but is wasteful for simple line drawings. The information above probably covers most things you might want to. However, if you want more information about what kinds of options you can specify using R, have a look at the help documentation by typing `?jpeg` or `?tiff` or whatever.

This page titled 4.7: Saving Image Files Using R and Rstudio is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **6.8: Saving Image Files Using R and Rstudio** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

4.8: Summary

Perhaps I'm a simple minded person, but I love pictures. Every time I write a new scientific paper, one of the first things I do is sit down and think about what the pictures will be. In my head, an article is really just a sequence of pictures, linked together by a story. All the rest of it is just window dressing. What I'm really trying to say here is that the human visual system is a very powerful data analysis tool. Give it the right kind of information and it will supply a human reader with a massive amount of knowledge very quickly. Not for nothing do we have the saying "a picture is worth a thousand words". With that in mind, I think that this is one of the most important chapters in the book. The topics covered were:

- *Basic overview to R graphics*. In Section 6.1 we talked about how graphics in R are organised, and then moved on to the basics of how they're drawn in Section 6.2.
- *Common plots*. Much of the chapter was focused on standard graphs that statisticians like to produce: histograms (Section 6.3, stem and leaf plots (Section 6.4, boxplots (Section 6.5, scatterplots (Section 6.6 and bar graphs (Section 6.7.
- *Saving image files*. The last part of the chapter talked about how to export your pictures (Section 6.8

One final thing to point out. At the start of the chapter I mentioned that R has several completely distinct systems for drawing figures. In this chapter I've focused on the *traditional* graphics system. It's the easiest one to get started with: you can draw a histogram with a command as simple as `hist(x)`. However, it's not the most powerful tool for the job, and after a while most R users start looking to shift to fancier systems. One of the most popular graphics systems is provided by the `ggplot2` package (see), which is loosely based on "The grammar of graphics" @[Wilkinson2006]. It's not for novices: you need to have a pretty good grasp of R before you can start using it, and even then it takes a while to really get the hang of it. But when you're finally at that stage, it's worth taking the time to teach yourself, because it's a much cleaner system.

---

86. The origin of this quote is Tufte's lovely book *The Visual Display of Quantitative Information*.
87. I should add that this isn't unique to R. Like everything in R there's a pretty steep learning curve to learning how to draw graphs, and like always there's a massive payoff at the end in terms of the quality of what you can produce. But to be honest, I've seen the same problems show up regardless of what system people use. I suspect that the hardest thing to do is to force yourself to take the time to think deeply about what your graphs are doing. I say that in full knowledge that only about half of my graphs turn out as well as they ought to. Understanding what makes a good graph is easy: actually designing a good graph is *hard*.
88. Or, since you can always use the up and down keys to scroll through your recent command history, you can just pull up your most recent commands and edit them to fix your mistake. It becomes even easier once you start using scripts (Section 8.1, since all you have to do is edit your script and then run it again.
89. Of course, even that is a slightly misleading description, since some R graphics tools make use of external graphical rendering systems like OpenGL (e.g., the `rgl` package). I absolutely will not be talking about OpenGL or the like in this book, but as it happens there is one graph in this book that relies on them: Figure 15.6.
90. The low-level function that does this is called `title()` in case you ever need to know, and you can type `?title` to find out a bit more detail about what these arguments do.
91. On the off chance that this isn't enough freedom for you, you can select a colour directly as a "red, green, blue" specification using the `rgb()` function, or as a "hue, saturation, value" specification using the `hsv()` function.
92. Also, there's a low level function called `axis()` that allows a lot more control over the appearance of the axes.
93. R being what it is, it's no great surprise that there's also a `fivenum()` function that does much the same thing.
94. I realise there's a kind of logic to the way R names are constructed, but they still sound dumb. When I typed this sentence, all I could think was that it sounded like the name of a kids movie if it had been written by Lewis Carroll: "The frabjous gambolles of Staplewex and Whisklty" or something along those lines.
95. Sometimes it's convenient to have the boxplot automatically label the outliers for you. The original `boxplot()` function doesn't allow you to do this; however, the `Boxplot()` function in the `car` package does. The design of the `Boxplot()` function is very similar to `boxplot()`. It just adds a few new arguments that allow you to tweak the labelling scheme. I'll leave it to the reader to check this out.
96. Sort of. The game was played in Launceston, which is a de facto home away from home for Hawthorn.

97. Contrast this situation with the next largest winning margin in the data set, which was Geelong's 108 point demolition of Richmond in round 6 at their home ground, Kardinia Park. Geelong have been one of the most dominant teams over the last several years, a period during which they strung together an incredible 29-game winning streak at Kardinia Park. Richmond have been useless for several years. This is in no meaningful sense an outlier. Geelong have been winning by these margins (and Richmond losing by them) for quite some time. Frankly I'm surprised that the result wasn't more lopsided: as happened to Melbourne in 2011 when Geelong won by a modest 186 points.
98. Actually, there's other ways to do this. If the input argument `x` is a list object (see Section 4.9, the `boxplot()` function will draw a separate boxplot for each variable in that list. Relatedly, since the `plot()` function – which we'll discuss shortly – is a generic (see Section 4.11, you might not be surprised to learn that one of its special cases is a boxplot: specifically, if you use `plot()` where the first argument `x` is a factor and the second argument `y` is numeric, then the result will be a boxplot, showing the values in `y`, with a separate boxplot for each level. For instance, something like `plot(x = afl2\$year, y = afl2\$margin)` would work.
99. The reason is that there's an annoying design flaw in the way the `plot()` function handles this situation. The problem is that the `plot.formula()` function uses different names to for the arguments than the `plot()` function expects. As a consequence, you can't specify the formula argument by name. If you just specify a formula as the first argument without using the name it works fine, because the `plot()` function thinks the formula corresponds to the `x` argument, and the `plot.formula()` function thinks it corresponds to the `formula` argument; and surprisingly, everything works nicely. But the moment that you, the user, tries to be unambiguous about the name, one of those two functions is going to cry.
100. You might be wondering why I haven't specified the argument name for the formula. The reason is that there's a bug in how the `scatterplot()` function is written: under the hood there's one function that expects the argument to be named `x` and another one that expects it to be called `formula`. I don't know why the function was written this way, but it's not an isolated problem: this particular kind of bug repeats itself in a couple of other functions (you'll see it again in Chapter 13. The solution in such cases is to omit the argument name: that way, one function "thinks" that you've specified `x` and the other one "thinks" you've specified `formula` and everything works the way it's supposed to. It's not a great state of affairs, I'll admit, but it sort of works.
101. Yet again, we could have produced this output using the `plot()` function: when the `x` argument is a data frame containing numeric variables only, then the output is a scatterplot matrix. So, once again, what I could have done is just type `plot( parenthood )`.
102. Once again, it's worth noting the link to the generic `plot()` function. If the `x` argument to `plot()` is a factor (and no `y` argument is given), the result is a bar graph. So you could use `plot( afl.finalists )` and get the same output as `barplot( afl.finalists )`.

This page titled 4.8: Summary is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **6.9: Summary** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

CHAPTER OVERVIEW

5: Summarizing Data With Numbers

Descriptive statistics often involves using a few numbers to summarize a distribution. One important aspect of a distribution is where its center is located. Measures of central tendency are discussed first. A second aspect of a distribution is how spread out it is. In other words, how much the numbers in the distribution vary from one another. The second section describes measures of variability. Distributions can differ in shape. Some distributions are symmetric whereas others have long tails in just one direction. The third section describes measures of the shape of distributions. The final two sections concern (1) how transformations affect measures summarizing distributions and (2) the variance sum law, an important relationship involving a measure of variability.

5.1: Central Tendency
5.2: What is Central Tendency
5.3: Measures of Central Tendency
5.4: Median and Mean
5.5: Measures of the Location of the Data
5.6: Additional Measures
5.7: Comparing Measures
5.8: Variability
5.9: Measures of Variability
5.10: Shapes of Distributions
5.11: Effects of Linear Transformations
5.12: Variance Sum Law I - Uncorrelated Variables
5.13: Statistical Literacy
5.14: Case Study- Using Stents to Prevent Strokes
5.15: Measures of the Location of the Data (Exercises)
5.E: Summarizing Distributions (Exercises)

This page titled 5: Summarizing Data With Numbers is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

5.1: Central Tendency

Central tendency is a loosely defined concept that has to do with the location of the center of a distribution. The section "What is Central Tendency" presents three definitions of the center of a distribution. "Measures of Central Tendency" presents the three most common measures of the center of the distribution. The three simulations that follow relate the definitions of the center of a distribution to the commonly used measures of central tendency. The findings from these simulations are summarized in the section "Mean and Median." The "Mean and Median" allows you to explore how the relative size of the mean and the median depends on the skew of the distribution.

Less frequently used measures of central tendency can be valuable supplements to the more commonly used measures. Some of these measures are presented in "Additional Measures." Finally, the last section compares and summarizes differences among measures of central tendency.

This page titled 5.1: Central Tendency is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **3.1: Central Tendency** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

5.2: What is Central Tendency

Learning Objectives

- Identify situations in which knowing the center of a distribution would be valuable
- Give three different ways the center of a distribution can be defined
- Describe how the balance is different for symmetric distributions than it is for asymmetric distributions

What is "central tendency," and why do we want to know the central tendency of a group of scores? Let us first try to answer these questions intuitively. Then we will proceed to a more formal discussion.

Imagine this situation: You are in a class with just four other students, and the five of you took a 5-point pop quiz. Today your instructor is walking around the room, handing back the quizzes. She stops at your desk and hands you your paper. Written in bold black ink on the front is "3/5." How do you react? Are you happy with your score of 3 or disappointed? How do you decide? You might calculate your percentage correct, realize it is 60%, and be appalled. But it is more likely that when deciding how to react to your performance, you will want additional information. What additional information would you like?

If you are like most students, you will immediately ask your neighbors, "Whad'ja get?" and then ask the instructor, "How did the class do?" In other words, the additional information you want is how your quiz score compares to other students' scores. You therefore understand the importance of comparing your score to the class distribution of scores. Should your score of 3 turn out to be among the higher scores, then you'll be pleased after all. On the other hand, if 3 is among the lower scores in the class, you won't be quite so happy.

This idea of comparing individual scores to a distribution of scores is fundamental to statistics. So let's explore it further, using the same example (the pop quiz you took with your four classmates). Three possible outcomes are shown in Table 5.2.1. They are labeled "Dataset A," "Dataset B," and "Dataset C." Which of the three datasets would make you happiest? In other words, in comparing your score with your fellow students' scores, in which dataset would your score of 3 be the most impressive?

In Dataset A, everyone's score is 3. This puts your score at the exact center of the distribution. You can draw satisfaction from the fact that you did as well as everyone else. But of course it cuts both ways: everyone else did just as well as you.

Table 5.2.1: Three possible datasets for the 5-point make-up quiz

Student	Dataset A	Dataset B	Dataset C
You	3	3	3
John's	3	4	2
Maria's	3	4	2
Shareecia's	3	4	2
Luther's	3	5	1

Now consider the possibility that the scores are described as in Dataset B. This is a depressing outcome even though your score is no different than the one in Dataset A. The problem is that the other four students had higher grades, putting yours below the center of the distribution.

Finally, let's look at Dataset C. This is more like it! All of your classmates score lower than you so your score is above the center of the distribution.

Now let's change the example in order to develop more insight into the center of a distribution. Figure 5.2.1 shows the results of an experiment on memory for chess positions. Subjects were shown a chess position and then asked to reconstruct it on an empty chess board. The number of pieces correctly placed was recorded. This was repeated for two more chess positions. The scores represent the total number of chess pieces correctly placed for the three chess positions. The maximum possible score was 89.

Two groups are compared. On the left are people who don't play chess. On the right are people who play a great deal (tournament players). It is clear that the location of the center of the distribution for the non-players is much lower than the center of the distribution for the tournament players.

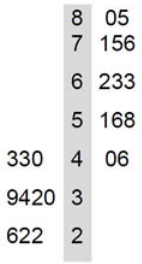


Figure 5.2.1: Back-to-back stem and leaf display. The left side shows the memory scores of the non-players. The right side shows the scores of the tournament players.

We're sure you get the idea now about the center of a distribution. It is time to move beyond intuition. We need a formal definition of the center of a distribution. In fact, we'll offer you three definitions! This is not just generosity on our part. There turn out to be (at least) three different ways of thinking about the center of a distribution, all of them useful in various contexts. In the remainder of this section we attempt to communicate the idea behind each concept. In the succeeding sections we will give statistical measures for these concepts of central tendency.

Definitions of Center

Now we explain the three different ways of defining the center of a distribution. All three are called measures of central tendency.

Balance Scale

One definition of central tendency is the point at which the distribution is in balance. Figure 5.2.2 shows the distribution of the five numbers $2, 3, 4, 9, 16$ placed upon a balance scale. If each number weighs one pound, and is placed at its position along the number line, then it would be possible to balance them by placing a fulcrum at 6.8.

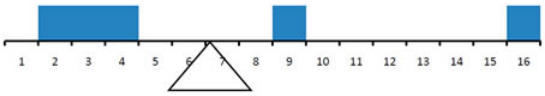


Figure 5.2.2: A balance scale

For another example, consider the distribution shown in Figure 5.2.3. It is balanced by placing the fulcrum in the geometric middle.

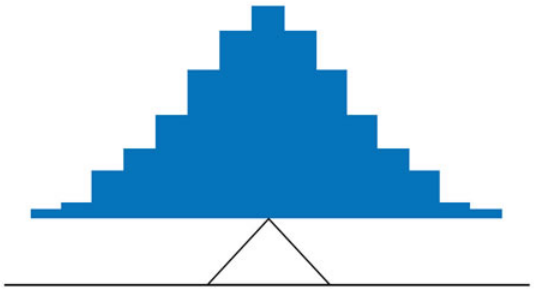


Figure 5.2.3: A distribution balanced on the tip of a triangle.

Figure 5.2.4 illustrates that the same distribution can't be balanced by placing the fulcrum to the left of center.

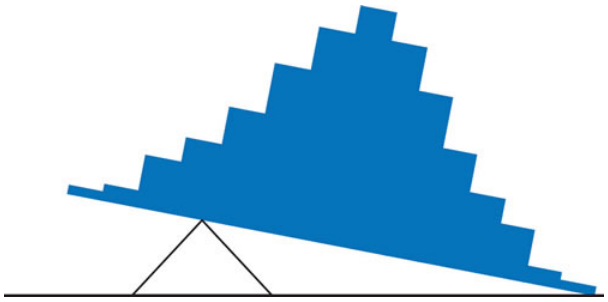


Figure 5.2.4: The distribution is not balanced

Figure 5.2.5 shows an asymmetric distribution. To balance it, we cannot put the fulcrum halfway between the lowest and highest values (as we did in Figure 5.2.3). Placing the fulcrum at the "half way" point would cause it to tip towards the left.

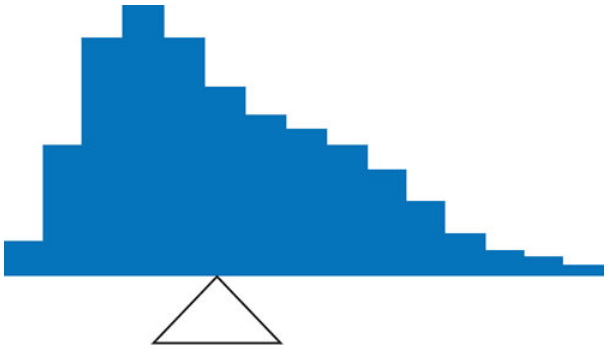


Figure 5.2.5: An asymmetric distribution balanced on the tip of a triangle

The balance point defines one sense of a distribution's center. The simulation in the next section "Balance Scale Simulation" shows how to find the point at which the distribution balances.

Smallest Absolute Deviation

Another way to define the center of a distribution is based on the concept of the sum of the absolute deviations (differences). Consider the distribution made up of the five numbers $2, 3, 4, 9, 16$. Let's see how far the distribution is from 10 (picking a number arbitrarily). Table 5.2.2 shows the sum of the absolute deviations of these numbers from the number 10.

Table 5.2.2: An example of the sum of absolute deviations

Values	Absolute Deviations from 10
2	8
3	7
4	6
9	1
16	6
Sum	28

The first row of the table shows that the absolute value of the difference between 2 and 10 is 8; the second row shows that the absolute difference between 3 and 10 is 7, and similarly for the other rows. When we add up the five absolute deviations, we get 28. So, the sum of the absolute deviations from 10 is 28. Likewise, the sum of the absolute deviations from 5 equals $3 + 2 + 1 + 4 + 11 = 21$. So, the sum of the absolute deviations from 5 is smaller than the sum of the absolute deviations from 10. In this sense, 5 is closer, overall, to the other numbers than is 10.

We are now in a position to define a second measure of central tendency, this time in terms of absolute deviations. Specifically, according to our second definition, the center of a distribution is the number for which the sum of the absolute deviations is

smallest. As we just saw, the sum of the absolute deviations from 10 is 28 and the sum of the absolute deviations from 5 is 21. Is there a value for which the sum of the absolute deviations is even smaller than 21? Yes. For these data, there is a value for which the sum of absolute deviations is only 20. See if you can find it. A general method for finding the center of a distribution in the sense of absolute deviations is provided in the simulation "Absolute Differences Simulation."

Smallest Squared Deviation

We shall discuss one more way to define the center of a distribution. It is based on the concept of the sum of squared deviations (differences). Again, consider the distribution of the five numbers $2, 3, 4, 9, 16$. Table $5.2.3$ shows the sum of the squared deviations of these numbers from the number 10.

Table $5.2.3$: An example of the sum of squared deviations

Values	Squared Deviations from 10
2	64
3	49
4	36
9	1
16	36
Sum	186

The first row in the table shows that the squared value of the difference between 2 and 10 is 64; the second row shows that the squared difference between 3 and 10 is 49, and so forth. When we add up all these squared deviations, we get 186. Changing the target from 10 to 5, we calculate the sum of the squared deviations from 5 as $9 + 4 + 1 + 16 + 121 = 151$. So, the sum of the squared deviations from 5 is smaller than the sum of the squared deviations from 10. Is there a value for which the sum of the squared deviations is even smaller than 151? Yes, it is possible to reach 134.8. Can you find the target number for which the sum of squared deviations is 134.8?

The target that minimizes the sum of squared deviations provides another useful definition of central tendency (the last one to be discussed in this section). It can be challenging to find the value that minimizes this sum. You will see how you do it in the upcoming section "Squared Differences Simulation."

Contributor

- Online Statistics Education: A Multimedia Course of Study (http://onlinestatbook.com/). Project Leader: David M. Lane, Rice University.
- David M. Lane and Heidi Ziemer

This page titled 5.2: What is Central Tendency is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **3.2: What is Central Tendency** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

5.3: Measures of Central Tendency

Learning Objectives

- Compute mode

In the previous section we saw that there are several ways to define central tendency. This section defines the three most common measures of central tendency: the mean, the median, and the mode. The relationships among these measures of central tendency and the definitions given in the previous section will probably not be obvious to you. Rather than just tell you these relationships, we will allow you to discover them in the simulations in the sections that follow. This section gives only the basic definitions of the mean, median and mode. A further discussion of the relative merits and proper applications of these statistics is presented in a later section.

Arithmetic Mean

The arithmetic mean is the most common measure of central tendency. It is simply the sum of the numbers divided by the number of numbers. The symbol "μ" is used for the mean of a population. The symbol "M" is used for the mean of a sample. The formula for μ is shown below:

$$\mu = \frac{\sum X}{N} \tag{5.3.1}$$

where $\sum X$ is the sum of all the numbers in the population and N is the number of numbers in the population.

The formula for M is essentially identical:

$$M = \frac{\sum X}{N} \tag{5.3.2}$$

where $\sum X$ is the sum of all the numbers in the sample and N is the number of numbers in the sample.

As an example, the mean of the numbers $1, 2, 3, 6, 8$ is $20/5 = 4$ regardless of whether the numbers constitute the entire population or just a sample from the population.

Table 5.3.1 shows the number of touchdown (TD) passes thrown by each of the 31 teams in the National Football League in the 2000 season.

***Table* 5.3.1**: *Number of touchdown passes*

37	33	33	32	29	28	28	23	22	22	22	21
21	21	20	20	19	19	18	18	18	18	16	15
14	14	14	12	12	9	6					

(5.3.3)

The mean number of touchdown passes thrown is 20.4516 as shown below.

$$\begin{aligned} \mu &= \sum X/N \\ &= 634/31 \\ &= 20.4516 \end{aligned}$$

Although the arithmetic mean is not the only "mean" (there is also a geometric mean), it is by far the most commonly used. Therefore, if the term "mean" is used without specifying whether it is the arithmetic mean, the geometric mean, or some other mean, it is assumed to refer to the arithmetic mean.

Median

The median is also a frequently used measure of central tendency. The median is the midpoint of a distribution: the same number of scores is above the median as below it. For the data in Table 5.3.1, there are 31 scores. The 16th highest score (which equals 20) is the median because there are 15 scores below the 16th score and 15 scores above the 16th score. The median can also be thought of as the 50th percentile.

Computation of the Median

When there is an odd number of numbers, the median is simply the middle number. For example, the median of 2, 4, and 7 is 4. When there is an even number of numbers, the median is the mean of the two middle numbers. Thus, the median of the numbers 2, 4, 7, 12 is $(4 + 7)/2 = 5.5$. When there are numbers with the same values, then the formula for the third definition of the 50th percentile should be used.

Mode

The mode is the most frequently occurring value. For the data in Table 5.3.2, the mode is 18 since more teams (4) had 18 touchdown passes than any other number of touchdown passes. With continuous data such as response time measured to many decimals, the frequency of each value is one since no two scores will be exactly the same (see discussion of continuous variables). Therefore the mode of continuous data is normally computed from a grouped frequency distribution. Table 5.3.2 shows a grouped frequency distribution for the target response time data. Since the interval with the highest frequency is 600 – 700, the mode is the middle of that interval (650).

Table 5.3.2: Grouped frequency distribution

Range	Frequency
500-600	3
600-700	6
700-800	5
800-900	5
900-1000	0
1000-1100	1

This page titled 5.3: Measures of Central Tendency is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **3.3: Measures of Central Tendency** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

5.4: Median and Mean

Learning Objectives

- State whether it is the mean or median that minimizes the mean absolute deviation
- State whether it is the mean or median that is the balance point on a balance scale

In the section "What is central tendency," we saw that the center of a distribution could be defined three ways:

1. the point on which a distribution would balance
2. the value whose average absolute deviation from all the other values is minimized
3. the value whose average squared difference from all the other values is minimized

From the simulation in this chapter, you discovered (we hope) that the mean is the point on which a distribution would balance, the median is the value that minimizes the sum of absolute deviations, and the mean is the value that minimizes the sum of the squared deviations.

Table 5.4.1 shows the absolute and squared deviations of the numbers $2, 3, 4, 9$ and 16 from their median of 4 and their mean of 6.8. You can see that the sum of absolute deviations from the median (20) is smaller than the sum of absolute deviations from the mean (22.8). On the other hand, the sum of squared deviations from the median (174) is larger than the sum of squared deviations from the mean (134.8).

Table 5.4.1: Absolute and squared deviations from the median of 4 and the mean of 6.8

Value	Absolute Deviation from Median	Absolute Deviation from Mean	Squared Deviation from Median	Squared Deviation from Mean
2	2	4.8	4	23.04
3	1	3.8	1	14.44
4	0	2.8	0	7.84
9	5	2.2	25	4.84
16	12	9.2	144	84.64
Total	20	22.8	174	134.8

Figure 5.4.1 shows that the distribution balances at the mean of 6.8 and not at the median of 4. The relative advantages and disadvantages of the mean and median are discussed in the section "Comparing Measures" later in this chapter.

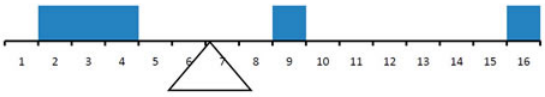


Figure 5.4.1: The distribution balances at the mean of 6.8 and not at the median of 4.0.

When a distribution is symmetric, then the mean and the median are the same. Consider the following distribution: $1, 3, 4, 5, 6, 7, 9$. The mean and median are both 5. The mean, median, and mode are identical in the bell-shaped normal distribution.

This page titled 5.4: Median and Mean is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **3.7: Median and Mean** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

5.5: Measures of the Location of the Data

The common measures of location are **quartiles** and **percentiles.** Quartiles are special percentiles. The first quartile, Q_1, is the same as the 25th percentile, and the third quartile, Q_3, is the same as the 75th percentile. The median, M, is called both the second quartile and the 50th percentile.

To calculate quartiles and percentiles, the data must be ordered from smallest to largest. Quartiles divide ordered data into quarters. Percentiles divide ordered data into hundredths. To score in the 90th percentile of an exam does not mean, necessarily, that you received 90% on a test. It means that 90% of test scores are the same or less than your score and 10% of the test scores are the same or greater than your test score.

Percentiles are useful for comparing values. For this reason, universities and colleges use percentiles extensively. One instance in which colleges and universities use percentiles is when SAT results are used to determine a minimum testing score that will be used as an acceptance factor. For example, suppose Duke accepts SAT scores at or above the 75th percentile. That translates into a score of at least 1220.

Percentiles are mostly used with very large populations. Therefore, if you were to say that 90% of the test scores are less (and not the same or less) than your score, it would be acceptable because removing one particular data value is not significant.

The **median** is a number that measures the "center" of the data. You can think of the median as the "middle value," but it does not actually have to be one of the observed values. It is a number that separates ordered data into halves. Half the values are the same number or smaller than the median, and half the values are the same number or larger. For example, consider the following data.

1; 11.5; 6; 7.2; 4; 8; 9; 10; 6.8; 8.3; 2; 2; 10; 1

Ordered from smallest to largest:

1; 1; 2; 2; 4; 6; 6.8; 7.2; 8; 8.3; 9; 10; 10; 11.5

Since there are 14 observations, the median is between the seventh value, 6.8, and the eighth value, 7.2. To find the median, add the two values together and divide by two.

$$\frac{6.8 + 7.2}{2} = 7 \tag{5.5.1}$$

The median is seven. Half of the values are smaller than seven and half of the values are larger than seven.

Quartiles are numbers that separate the data into quarters. Quartiles may or may not be part of the data. To find the quartiles, first find the median or second quartile. The first quartile, Q_1, is the middle value of the lower half of the data, and the third quartile, Q_3, is the middle value, or median, of the upper half of the data. To get the idea, consider the same data set:

1; 1; 2; 2; 4; 6; 6.8; 7.2; 8; 8.3; 9; 10; 10; 11.5

The median or **second quartile** is seven. The lower half of the data are 1, 1, 2, 2, 4, 6, 6.8. The middle value of the lower half is two.

1; 1; 2; 2; 4; 6; 6.8

The number two, which is part of the data, is the **first quartile**. One-fourth of the entire sets of values are the same as or less than two and three-fourths of the values are more than two.

The upper half of the data is 7.2, 8, 8.3, 9, 10, 10, 11.5. The middle value of the upper half is nine.

The **third quartile**, $Q3$, is nine. Three-fourths (75%) of the ordered data set are less than nine. One-fourth (25%) of the ordered data set are greater than nine. The third quartile is part of the data set in this example.

The **interquartile range** is a number that indicates the spread of the middle half or the middle 50% of the data. It is the difference between the third quartile (Q_3) and the first quartile (Q_1).

$$IQR = Q_3 – Q_1 \tag{2.4.1}$$

The *IQR* can help to determine potential **outliers**. **A value is suspected to be a potential outlier if it is less than (1.5)(*IQR*) below the first quartile or more than (1.5)(*IQR*) above the third quartile**. Potential outliers always require further investigation.

Definition: Outliers

A potential outlier is a data point that is significantly different from the other data points. These special data points may be errors or some kind of abnormality or they may be a key to understanding the data.

Example 2.4.1

For the following 13 real estate prices, calculate the *IQR* and determine if any prices are potential outliers. Prices are in dollars.

389,950; 230,500; 158,000; 479,000; 639,000; 114,950; 5,500,000; 387,000; 659,000; 529,000; 575,000; 488,800; 1,095,000

Answer

Order the data from smallest to largest.

114,950; 158,000; 230,500; 387,000; 389,950; 479,000; 488,800; 529,000; 575,000; 639,000; 659,000; 1,095,000; 5,500,000

$$M = 488,800$$

$$Q_1 = \frac{230,500 + 387,000}{2} = 308,750$$

$$Q_3 = \frac{639,000 + 659,000}{2} = 649,000$$

$$IQR = 649,000 - 308,750 = 340,250$$

$$(1.5)(IQR) = (1.5)(340,250) = 510,375$$

$$Q_1 - (1.5)(IQR) = 308,750 - 510,375 = -201,625$$

$$Q_3 + (1.5)(IQR) = 649,000 + 510,375 = 1,159,375$$

No house price is less than –201,625. However, 5,500,000 is more than 1,159,375. Therefore, 5,500,000 is a potential **outlier**.

Exercise 5.5.1

For the following 11 salaries, calculate the *IQR* and determine if any salaries are outliers. The salaries are in dollars.

$33,000; $64,500; $28,000; $54,000; $72,000; $68,500; $69,000; $42,000; $54,000; $120,000; $40,500

Answer

Order the data from smallest to largest.

$28,000; $33,000; $40,500; $42,000; $54,000; $54,000; $64,500; $68,500; $69,000; $72,000; $120,000

Median = $54,000

$$Q_1 = \$40,500$$

$$Q_3 = \$69,000$$

$$IQR = \$69,000 - \$40,500 = \$28,500$$

$$(1.5)(IQR) = (1.5)(\$28,500) = \$42,750$$

$$Q_1 - (1.5)(IQR) = \$40,500 - \$42,750 = -\$2,250$$

$$Q_3 + (1.5)(IQR) = \$69,000 + \$42,750 = \$111,750$$

No salary is less than –$2,250. However, $120,000 is more than $11,750, so $120,000 is a potential outlier.

Example 2.4.2

For the two data sets in the test scores example, find the following:

a. The interquartile range. Compare the two interquartile ranges.
b. Any outliers in either set.

Answer

The five number summary for the day and night classes is

	Minimum	Q_1	Median	Q_3	Maximum
Day	32	56	74.5	82.5	99
Night	25.5	78	81	89	98

a. The *IQR* for the day group is $Q_3 - Q_1 = 82.5 - 56 = 26.5$

The *IQR* for the night group is $Q_3 - Q_1 = 89 - 78 = 11$

The interquartile range (the spread or variability) for the day class is larger than the night class *IQR*. This suggests more variation will be found in the day class's class test scores.

b. Day class outliers are found using the IQR times 1.5 rule. So,

- $Q_1 - IQR(1.5) = 56–26.5(1.5) = 16.25$
- $Q_3 + IQR(1.5) = 82.5 + 26.5(1.5) = 122.25$

Since the minimum and maximum values for the day class are greater than 16.25 and less than 122.25, there are no outliers.

Night class outliers are calculated as:

- $Q_1 - IQR(1.5) = 78–11(1.5) = 61.5$
- $Q_3 + IQR(1.5) = 89 + 11(1.5) = 105.5$

For this class, any test score less than 61.5 is an outlier. Therefore, the scores of 45 and 25.5 are outliers. Since no test score is greater than 105.5, there is no upper end outlier.

Exercise 5.5.2

Find the interquartile range for the following two data sets and compare them.

Test Scores for Class *A*

69; 96; 81; 79; 65; 76; 83; 99; 89; 67; 90; 77; 85; 98; 66; 91; 77; 69; 80; 94

Test Scores for Class *B*

90; 72; 80; 92; 90; 97; 92; 75; 79; 68; 70; 80; 99; 95; 78; 73; 71; 68; 95; 100

Answer

Class *A*

Order the data from smallest to largest.

65; 66; 67; 69; 69; 76; 77; 77; 79; 80; 81; 83; 85; 89; 90; 91; 94; 96; 98; 99

$Median = \frac{80 + 81}{2} = 80.5$

$Q_1 = \frac{69 + 76}{2} = 72.5$

$Q_3 = \frac{90 + 91}{2} = 90.5$

$IQR = 90.5 - 72.5 = 18$

Class B

Order the data from smallest to largest.

68; 68; 70; 71; 72; 73; 75; 78; 79; 80; 80; 90; 90; 92; 92; 95; 95; 97; 99; 100

$Median = \frac{80+80}{2} = 80$

$Q_1 = \frac{72+73}{2} = 72.5$

$Q_3 = \frac{92+95}{2} = 93.5$

$IQR = 93.5 - 72.5 = 21$

The data for Class B has a larger IQR, so the scores between Q_3 and Q_1 (middle 50%) for the data for Class B are more spread out and not clustered about the median.

Example 5.5.3

Fifty statistics students were asked how much sleep they get per school night (rounded to the nearest hour). The results were:

AMOUNT OF SLEEP PER SCHOOL NIGHT (HOURS)	FREQUENCY	RELATIVE FREQUENCY	CUMULATIVE RELATIVE FREQUENCY
4	2	0.04	0.04
5	5	0.10	0.14
6	7	0.14	0.28
7	12	0.24	0.52
8	14	0.28	0.80
9	7	0.14	0.94
10	3	0.06	1.00

Find the 28th percentile. Notice the 0.28 in the "cumulative relative frequency" column. Twenty-eight percent of 50 data values is 14 values. There are 14 values less than the 28th percentile. They include the two 4s, the five 5s, and the seven 6s. The 28th percentile is between the last six and the first seven. **The 28th percentile is 6.5.**

Find the median. Look again at the "cumulative relative frequency" column and find 0.52. The median is the 50th percentile or the second quartile. 50% of 50 is 25. There are 25 values less than the median. They include the two 4s, the five 5s, the seven 6s, and eleven of the 7s. The median or 50th percentile is between the 25th, or seven, and 26th, or seven, values. **The median is seven.**

Find the third quartile. The third quartile is the same as the 75th percentile. You can "eyeball" this answer. If you look at the "cumulative relative frequency" column, you find 0.52 and 0.80. When you have all the fours, fives, sixes and sevens, you have 52% of the data. When you include all the 8s, you have 80% of the data. **The 75th percentile, then, must be an eight**. Another way to look at the problem is to find 75% of 50, which is 37.5, and round up to 38. The third quartile, Q_3, is the 38th value, which is an eight. You can check this answer by counting the values. (There are 37 values below the third quartile and 12 values above.)

Exercise 5.5.3

Forty bus drivers were asked how many hours they spend each day running their routes (rounded to the nearest hour). Find the 65th percentile.

Amount of time spent on route (hours)	Frequency	Relative Frequency	Cumulative Relative Frequency
2	12	0.30	0.30
3	14	0.35	0.65
4	10	0.25	0.90
5	4	0.10	1.00

Answer

The 65th percentile is between the last three and the first four.

The 65th percentile is 3.5.

Example 2.4.4

Using the table above in Example 5.5.3

a. Find the 80th percentile.
b. Find the 90th percentile.
c. Find the first quartile. What is another name for the first quartile?

Solution

Using the data from the frequency table, we have:

a. The 80th percentile is between the last eight and the first nine in the table (between the 40th and 41st values). Therefore, we need to take the mean of the 40th an 41st values. The 80th percentile $= \frac{8+9}{2} = 8.5$
b. The 90th percentile will be the 45th data value (location is $0.90(50) = 45$) and the 45th data value is nine.
c. Q_1 is also the 25th percentile. The 25th percentile location calculation: $P_{25} = 0.25(50) = 12.5 \approx 13$ the 13th data value. Thus, the 25th percentile is six.

Exercise 5.5.4

Refer to the table above in Exercise 5.5.3. Find the third quartile. What is another name for the third quartile?

Answer

The third quartile is the 75th percentile, which is four. The 65th percentile is between three and four, and the 90th percentile is between four and 5.75. The third quartile is between 65 and 90, so it must be four.

COLLABORATIVE STATISTICS

Your instructor or a member of the class will ask everyone in class how many sweaters they own. Answer the following questions:

a. How many students were surveyed?
b. What kind of sampling did you do?
c. Construct two different histograms. For each, starting value = _____ ending value = ____.
d. Find the median, first quartile, and third quartile.
e. Construct a table of the data to find the following:
   i. the 10th percentile
   ii. the 70th percentile
   iii. the percent of students who own less than four sweaters

A Formula for Finding the *k*th Percentile

If you were to do a little research, you would find several formulas for calculating the kth percentile. Here is one of them.

- $k =$ the kth percentile. It may or may not be part of the data.
- $i =$ the index (ranking or position of a data value)
- $n =$ the total number of data

Order the data from smallest to largest.

Calculate $i = \frac{k}{100}(n+1)$

If i is an integer, then the kth percentile is the data value in the ith position in the ordered set of data.

If i is not an integer, then round i up and round i down to the nearest integers. Average the two data values in these two positions in the ordered data set. This is easier to understand in an example.

Example 2.4.5

Listed are 29 ages for Academy Award winning best actors *in order from smallest to largest.*

18; 21; 22; 25; 26; 27; 29; 30; 31; 33; 36; 37; 41; 42; 47; 52; 55; 57; 58; 62; 64; 67; 69; 71; 72; 73; 74; 76; 77

a. Find the 70th percentile.
b. Find the 83rd percentile.

Solution

a.
- $k = 70$
- $i =$ the index
- $n = 29$

$i = \frac{k}{100}(n+1) = \frac{70}{100}(29+1) = 21$. Twenty-one is an integer, and the data value in the 21st position in the ordered data set is 64. The 70th percentile is 64 years.

b.
- $k =$ 83rd percentile
- $i =$ the index
- $n = 29$

$i = \frac{k}{100}(n+1) = (\frac{83}{100})(29+1) = 24.9$, which is NOT an integer. Round it down to 24 and up to 25. The age in the 24th position is 71 and the age in the 25th position is 72. Average 71 and 72. The 83rd percentile is 71.5 years.

Exercise 5.5.5

Listed are 29 ages for Academy Award winning best actors *in order from smallest to largest.*

18; 21; 22; 25; 26; 27; 29; 30; 31; 33; 36; 37; 41; 42; 47; 52; 55; 57; 58; 62; 64; 67; 69; 71; 72; 73; 74; 76; 77

Calculate the 20th percentile and the 55th percentile.

Answer

$k = 20$. Index $= i = \frac{k}{100}(n+1) = \frac{20}{100}(29+1) = 6$. The age in the sixth position is 27. The 20th percentile is 27 years.

$k = 55$. Index $= i = \frac{k}{100}(n+1) = \frac{55}{100}(29+1) = 16.5$. Round down to 16 and up to 17. The age in the 16th position is 52 and the age in the 17th position is 55. The average of 52 and 55 is 53.5. The 55th percentile is 53.5 years.

Note 2.4.2

You can calculate percentiles using calculators and computers. There are a variety of online calculators.

A Formula for Finding the Percentile of a Value in a Data Set

- Order the data from smallest to largest.
- $x =$ the number of data values counting from the bottom of the data list up to but not including the data value for which you want to find the percentile.
- $y =$ the number of data values equal to the data value for which you want to find the percentile.
- $n =$ the total number of data.
- Calculate $\frac{x + 0.5y}{n}(100)$. Then round to the nearest integer.

Example 2.4.6

Listed are 29 ages for Academy Award winning best actors *in order from smallest to largest.*

18; 21; 22; 25; 26; 27; 29; 30; 31; 33; 36; 37; 41; 42; 47; 52; 55; 57; 58; 62; 64; 67; 69; 71; 72; 73; 74; 76; 77

a. Find the percentile for 58.
b. Find the percentile for 25.

Solution

a. Counting from the bottom of the list, there are 18 data values less than 58. There is one value of 58.

$x = 18$ and $y = 1$. $\frac{x + 0.5y}{n}(100) = \frac{18 + 0.5(1)}{29}(100) = 63.80$. 58 is the 64th percentile.

b. Counting from the bottom of the list, there are three data values less than 25. There is one value of 25.

$x = 3$ and $y = 1$. $\frac{x + 0.5y}{n}(100) = \frac{3 + 0.5(1)}{29}(100) = 12.07$. Twenty-five is the 12thpercentile.

Exercise 5.5.6

Listed are 30 ages for Academy Award winning best actors in order from smallest to largest.

18; 21; 22; 25; 26; 27; 29; 30; 31, 31; 33; 36; 37; 41; 42; 47; 52; 55; 57; 58; 62; 64; 67; 69; 71; 72; 73; 74; 76; 77

Find the percentiles for 47 and 31.

Answer

Percentile for 47: Counting from the bottom of the list, there are 15 data values less than 47. There is one value of 47.

$x = 15$ and $y = 1$. $\frac{x + 0.5y}{n}(100) = \frac{15 + 0.5(1)}{30}(100) = 51.67$. 47 is the 52nd percentile.

Percentile for 31: Counting from the bottom of the list, there are eight data values less than 31. There are two values of 31.

$x = 8$ and $y = 2$. $\frac{x + 0.5y}{n}(100) = \frac{8 + 0.5(2)}{30}(100) = 30$. 31 is the 30th percentile.

Interpreting Percentiles, Quartiles, and Median

A percentile indicates the relative standing of a data value when data are sorted into numerical order from smallest to largest. Percentages of data values are less than or equal to the pth percentile. For example, 15% of data values are less than or equal to the 15th percentile.

- Low percentiles always correspond to lower data values.
- High percentiles always correspond to higher data values.

A percentile may or may not correspond to a value judgment about whether it is "good" or "bad." The interpretation of whether a certain percentile is "good" or "bad" depends on the context of the situation to which the data applies. In some situations, a low percentile would be considered "good;" in other contexts a high percentile might be considered "good". In many situations, there is no value judgment that applies.

Understanding how to interpret percentiles properly is important not only when describing data, but also when calculating probabilities in later chapters of this text.

GUIDELINE

When writing the interpretation of a percentile in the context of the given data, the sentence should contain the following information.

- information about the context of the situation being considered
- the data value (value of the variable) that represents the percentile
- the percent of individuals or items with data values below the percentile
- the percent of individuals or items with data values above the percentile.

On a timed math test, the first quartile for time it took to finish the exam was 35 minutes. Interpret the first quartile in the context of this situation.

Answer

- Twenty-five percent of students finished the exam in 35 minutes or less.
- Seventy-five percent of students finished the exam in 35 minutes or more.
- A low percentile could be considered good, as finishing more quickly on a timed exam is desirable. (If you take too long, you might not be able to finish.)

? Exercise 5.5.7

For the 100-meter dash, the third quartile for times for finishing the race was 11.5 seconds. Interpret the third quartile in the context of the situation.

Answer

Twenty-five percent of runners finished the race in 11.5 seconds or more. Seventy-five percent of runners finished the race in 11.5 seconds or less. A lower percentile is good because finishing a race more quickly is desirable.

On a 20 question math test, the 70th percentile for number of correct answers was 16. Interpret the 70th percentile in the context of this situation.

Answer

- Seventy percent of students answered 16 or fewer questions correctly.
- Thirty percent of students answered 16 or more questions correctly.
- A higher percentile could be considered good, as answering more questions correctly is desirable.

? Exercise 5.5.8

On a 60 point written assignment, the 80th percentile for the number of points earned was 49. Interpret the 80th percentile in the context of this situation.

Answer

Eighty percent of students earned 49 points or fewer. Twenty percent of students earned 49 or more points. A higher percentile is good because getting more points on an assignment is desirable.

Example 2.4.9

At a community college, it was found that the 30th percentile of credit units that students are enrolled for is seven units. Interpret the 30th percentile in the context of this situation.

Answer

- Thirty percent of students are enrolled in seven or fewer credit units.
- Seventy percent of students are enrolled in seven or more credit units.
- In this example, there is no "good" or "bad" value judgment associated with a higher or lower percentile. Students attend community college for varied reasons and needs, and their course load varies according to their needs.

Exercise 5.5.9

During a season, the 40th percentile for points scored per player in a game is eight. Interpret the 40th percentile in the context of this situation.

Answer

Forty percent of players scored eight points or fewer. Sixty percent of players scored eight points or more. A higher percentile is good because getting more points in a basketball game is desirable.

Example 2.4.10

Sharpe Middle School is applying for a grant that will be used to add fitness equipment to the gym. The principal surveyed 15 anonymous students to determine how many minutes a day the students spend exercising. The results from the 15 anonymous students are shown.

0 minutes; 40 minutes; 60 minutes; 30 minutes; 60 minutes

10 minutes; 45 minutes; 30 minutes; 300 minutes; 90 minutes;

30 minutes; 120 minutes; 60 minutes; 0 minutes; 20 minutes

Determine the following five values.

- Min = 0
- Q_1 = 20
- Med = 40
- Q_3 = 60
- Max = 300

If you were the principal, would you be justified in purchasing new fitness equipment? Since 75% of the students exercise for 60 minutes or less daily, and since the *IQR* is 40 minutes (60 – 20 = 40), we know that half of the students surveyed exercise between 20 minutes and 60 minutes daily. This seems a reasonable amount of time spent exercising, so the principal would be justified in purchasing the new equipment.

However, the principal needs to be careful. The value 300 appears to be a potential outlier.

$$Q_3 + 1.5(IQR) = 60 + (1.5)(40) = 120 \tag{5.5.2}$$

.

The value 300 is greater than 120 so it is a potential outlier. If we delete it and calculate the five values, we get the following values:

- Min = 0
- Q_1 = 20
- Q_3 = 60
- Max = 120

We still have 75% of the students exercising for 60 minutes or less daily and half of the students exercising between 20 and 60 minutes a day. However, 15 students is a small sample and the principal should survey more students to be sure of his survey results.

References

1. Cauchon, Dennis, Paul Overberg. “Census data shows minorities now a majority of U.S. births.” USA Today, 2012. Available online at usatoday30.usatoday.com/news/...sus/55029100/1 (accessed April 3, 2013).
2. Data from the United States Department of Commerce: United States Census Bureau. Available online at http://www.census.gov/ (accessed April 3, 2013).
3. “1990 Census.” United States Department of Commerce: United States Census Bureau. Available online at http://www.census.gov/main/www/cen1990.html (accessed April 3, 2013).
4. Data from *San Jose Mercury News*.
5. Data from *Time Magazine*; survey by Yankelovich Partners, Inc.

Review

The values that divide a rank-ordered set of data into 100 equal parts are called percentiles. Percentiles are used to compare and interpret data. For example, an observation at the 50th percentile would be greater than 50 percent of the other obeservations in the set. Quartiles divide data into quarters. The first quartile (Q_1) is the 25th percentile,the second quartile (Q_2 or median) is 50th percentile, and the third quartile (Q_3) is the the 75th percentile. The interquartile range, or *IQR*, is the range of the middle 50 percent of the data values. The *IQR* is found by subtracting Q_1 from Q_3, and can help determine outliers by using the following two expressions.

- $Q_3 + IQR(1.5)$
- $Q_1 - IQR(1.5)$

Formula Review

$$i = \frac{k}{100}(n+1)$$

where i = the ranking or position of a data value,

- k = the kth percentile,
- n = total number of data.

Expression for finding the percentile of a data value: $\left(\frac{x+0.5y}{n}\right)(100)$

where $x =$ the number of values counting from the bottom of the data list up to but not including the data value for which you want to find the percentile,

$y =$ the number of data values equal to the data value for which you want to find the percentile,

$n =$ total number of data

Glossary

Interquartile Range

or *IQR*, is the range of the middle 50 percent of the data values; the *IQR* is found by subtracting the first quartile from the third quartile.

Outlier

an observation that does not fit the rest of the data

Percentile

a number that divides ordered data into hundredths; percentiles may or may not be part of the data. The median of the data is the second quartile and the 50th percentile. The first and third quartiles are the 25th and the 75th percentiles, respectively.

Quartiles

the numbers that separate the data into quarters; quartiles may or may not be part of the data. The second quartile is the median of the data.

This page titled 5.5: Measures of the Location of the Data is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by OpenStax via source content that was edited to the style and standards of the LibreTexts platform.

- **2.4: Measures of the Location of the Data** by OpenStax is licensed CC BY 4.0. Original source: https://openstax.org/details/books/introductory-statistics.

5.6: Additional Measures

Learning Objectives

- Compute the trimean
- Compute the geometric mean directly
- Compute the geometric mean using logs
- Use the geometric to compute annual portfolio returns
- Compute a trimmed mean

Although the mean, median, and mode are by far the most commonly used measures of central tendency, they are by no means the only measures. This section defines three additional measures of central tendency: the trimean, the geometric mean, and the trimmed mean. These measures will be discussed again in the section "Comparing Measures of Central Tendency."

Trimean

The trimean is a weighted average of the 25th percentile, the 50th percentile, and the 75th percentile. Letting $P25$ be the 25th percentile, $P50$ be the $50^{th}$and $P75$ be the 75th percentile, the formula for the trimean is:

$$Trimean = \frac{P25 + 2P50 + P75}{4} \tag{5.6.1}$$

As you can see from the formula, the median is weighted twice as much as the 25th and 75th percentiles. Table 5.6.1 shows the number of touchdown (TD) passes thrown by each of the 31 teams in the National Football League in the 2000 season. The relevant percentiles are shown in Table 5.6.2.

Table 5.6.1: *Number of touchdown passes*

37	33	33	32	29	28	28	23
22	22	22	21	21	21	20	20
19	19	18	18	18	18	16	15
14	14	14	12	12	9	6	

(5.6.2)

Table 5.6.2: *Percentiles*

Percentile	Value
25	15
50	20
75	23

The trimean is therefore

$$\frac{15 + 2 \times 20 + 23}{4} = \frac{78}{4} = 19.5. \tag{5.6.3}$$

Geometric Mean

The geometric mean is computed by multiplying all the numbers together and then taking the nth root of the product. For example, for the numbers $1, 10$ and 100, the product of all the numbers is:

$$1 \times 10 \times 100 = 1,000. \tag{5.6.4}$$

Since there are three numbers, we take the cubed root of the product $(1,000)$ which is equal to 10. The formula for the geometric mean is therefore

$$(\Pi X)^{1/N} \tag{5.6.5}$$

where the symbol Π means to multiply. Therefore, the equation says to multiply all the values of X and then raise the result to the $1/N$th power. Raising a value to the $\frac{1}{N}^{th}$ power is, of course, the same as taking the Nth root of the value. In this case, $1000^{1/3}$ is the cube root of $1,000$.

The geometric mean has a close relationship with logarithms. Table 5.6.3 shows the logs (base 10) of these three numbers. The arithmetic mean of the three logs is 1. The anti-log of this **arithmetic mean** of 1 is the **geometric mean**. The anti-log of 1 is $10^1 = 10$. Note that the geometric mean only makes sense if all the numbers are positive.

Table 5.6.3: *Logarithms*

X	$\log_{10}(X)$
1	0
10	1
100	2

The geometric mean is an appropriate measure to use for averaging rates. For example, consider a stock portfolio that began with a value of $\$1,000$ and had annual returns of $13\%, 22\%, 12\%, -5\%, and -13\%$. Table 5.6.4 shows the value after each of the five years.

Table 5.6.4: *Portfolio Returns*

Year	Return	Value
1	13%	1,130
2	22%	1,379
3	12%	1,544
4	-5%	1,467
5	-13%	1,276

The question is how to compute average annual rate of return. The answer is to compute the geometric mean of the returns. Instead of using the percents, each return is represented as a multiplier indicating how much higher the value is after the year. This multiplier is 1.13 for a 13% return and 0.95 for a 5% loss. The multipliers for this example are $1.13, 1.22, 1.12, 0.95, and\ 0.87$. The geometric mean of these multipliers is 1.05. Therefore, the average annual rate of return is 5%. Table 5.6.5 shows how a portfolio gaining 5% a year would end up with the same value ($\$1,276$) as shown in Table 5.6.4.

Table 5.6.5: *Portfolio Returns*

Year	Return	Value
1	5%	1,050
2	5%	1,103
3	5%	1,158
4	5%	1,216
5	5%	1,276

Trimmed Mean

To compute a trimmed mean, you remove some of the higher and lower scores and compute the mean of the remaining scores. A mean trimmed 10% is a mean computed with 10% of the scores trimmed off: 5% from the bottom and 5% from the top. A mean trimmed 50% is computed by trimming the upper 25% of the scores and the lower 25% of the scores and computing the mean of the

remaining scores. The trimmed mean is similar to the median which, in essence, trims the upper 49% and the lower 49% of the scores. Therefore the trimmed mean is a hybrid of the mean and the median. To compute the mean trimmed 20% for the touchdown pass data shown in Table $5.6.1$, you remove the lower 10% of the scores ($6, 9,$ *and* 12) as well as the upper 10% of the scores ($33, 33,$ *and* 37) and compute the mean of the remaining 25 scores. This mean is 20.16.

This page titled 5.6: Additional Measures is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **3.9: Additional Measures** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

5.7: Comparing Measures

Learning Objectives

- State how the measures differ in symmetric distributions
- State which measure(s) should be used to describe the center of a skewed distribution

How do the various measures of central tendency compare with each other? For symmetric distributions, the mean, median, trimean, and trimmed mean are equal, as is the mode except in bimodal distributions. Differences among the measures occur with skewed distributions. Figure 5.7.1 shows the distribution of 642 scores on an introductory psychology test. Notice this distribution has a slight positive skew.

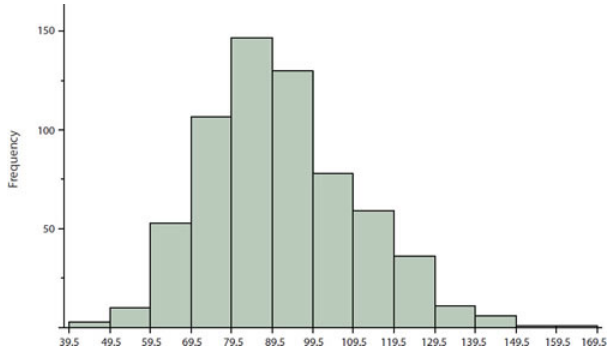


Figure 5.7.1: *A distribution with a positive skew.*

Measures of central tendency are shown in Table 5.7.1. Notice they do not differ greatly, with the exception that the mode is considerably lower than the other measures. When distributions have a positive skew, the mean is typically higher than the median, although it may not be in bimodal distributions. For these data, the mean of 91.58 is higher than the median of 90. Typically the trimean and trimmed mean will fall between the median and the mean, although in this case, the trimmed mean is slightly lower than the median. The geometric mean is lower than all measures except the mode.

Table 5.7.1: *Measures of central tendency for the test scores*

Measure	Value
Mode	84.00
Median	90.00
Geometric Mean	89.70
Trimean	90.25
Mean Trimmed 50%	89.81
Mean	91.58

The distribution of baseball salaries (in 1994) shown in Figure 5.7.2 has a much more pronounced skew than the distribution in Figure 5.7.2.

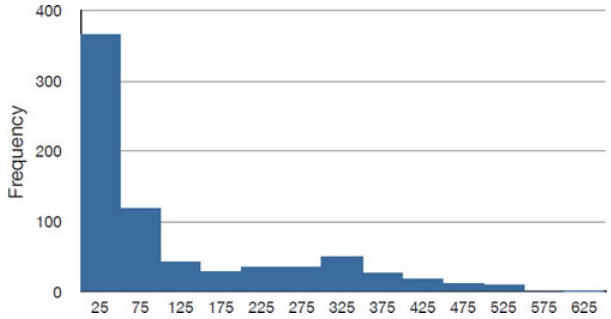


Figure 5.7.2: A distribution with a very large positive skew. This histogram shows the salaries of major league baseball players (in thousands of dollars: 25 equals 250,000).

Table 5.7.2 shows the measures of central tendency for these data. The large skew results in very different values for these measures. No single measure of central tendency is sufficient for data such as these. If you were asked the very general question: "So, what do baseball players make?" and answered with the mean of $1,183,000, you would not have told the whole story since only about one third of baseball players make that much. If you answered with the mode of $250,000 or the median of $500,000, you would not be giving any indication that some players make many millions of dollars. Fortunately, there is no need to summarize a distribution with a single number. When the various measures differ, our opinion is that you should report the mean, median, and either the trimean or the mean trimmed 50%. Sometimes it is worth reporting the mode as well. In the media, the median is usually reported to summarize the center of skewed distributions. You will hear about median salaries and median prices of houses sold, etc. This is better than reporting only the mean, but it would be informative to hear more statistics.

Table 5.7.2: Measures of central tendency for baseball salaries (in thousands of dollars)

Measure	Value
Mode	250
Median	500
Geometric Mean	555
Trimean	792
Mean Trimmed 50%	619
Mean	1,183

This page titled 5.7: Comparing Measures is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **3.10: Comparing Measures** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

5.8: Variability

Learning Objectives

- To study how much the numbers in a distribution differ from each other

Variability refers to how much the numbers in a distribution differ from each other. The most common measures are presented in "Measures of Variability." The "variability demo" allows you to change the standard deviation of a distribution and view a graph of the changed distribution.

One of the more counter-intuitive facts in introductory statistics is that the formula for variance when computed in a population is biased when applied in a sample. The "Estimating Variance Simulation" shows concretely why this is the case.

This page titled 5.8: Variability is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **3.11: Variability** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

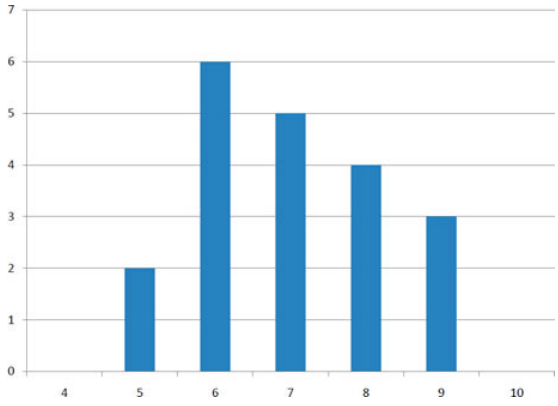
5.9: Measures of Variability

Learning Objectives

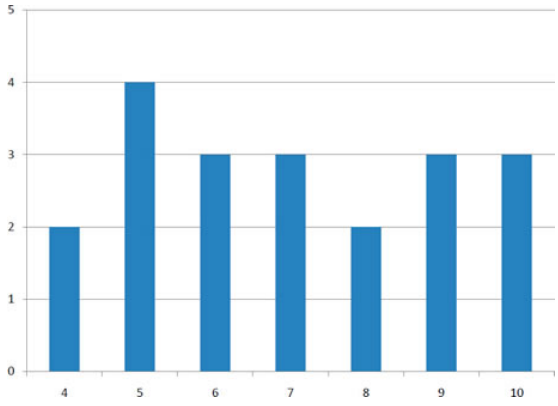
- Compute the range
- Compute the variance in the population
- Compute the standard deviation from the variance

What is Variability?

Variability refers to how "spread out" a group of scores is. To see what we mean by spread out, consider graphs in Figure 5.9.1. These graphs represent the scores on two quizzes. The mean score for each quiz is 7.0. Despite the equality of means, you can see that the distributions are quite different. Specifically, the scores on **Quiz 1** are more densely packed and those on **Quiz 2** are more spread out. The differences among students were much greater on **Quiz 2** than on **Quiz 1**.



Quiz 1



Quiz 1

***Figure* 5.9.1**: *Bar charts of two quizzes*

The terms variability, spread, and dispersion are synonyms, and refer to how spread out a distribution is. Just as in the section on central tendency where we discussed measures of the center of a distribution of scores, in this chapter we will discuss measures of the variability of a distribution. There are four frequently used measures of variability: the range, interquartile range, variance, and standard deviation. In the next few paragraphs, we will look at each of these four measures of variability in more detail.

Range

The range is the simplest measure of variability to calculate, and one you have probably encountered many times in your life. The range is simply the highest score minus the lowest score. Let's take a few examples. What is the range of the following group of numbers: $10, 2, 5, 6, 7, 3, 4$? Well, the highest number is 10, and the lowest number is 2, so $10 - 2 = 8$. The range is 8. Let's take

another example. Here's a dataset with 10 numbers: 99, 45, 23, 67, 45, 91, 82, 78, 62, 51. What is the range? The highest number is 99 and the lowest number is 23, so 99 − 23 equals 76; the range is 76. Now consider the two quizzes shown in Figure 5.9.1. On Quiz 1, the lowest score is 5 and the highest score is 9. Therefore, the range is 4. The range on Quiz 2 was larger: the lowest score was 4 and the highest score was 10. Therefore the range is 6.

Interquartile Range

The interquartile range (IQR) is the range of the middle 50% of the scores in a distribution. It is computed as follows:

$$IQR = 75^{th} percentile - 25^{th} percentile \tag{5.9.1}$$

For Quiz 1, the 75th percentile is 8 and the 25th percentile is 6. The interquartile range is therefore 2. For Quiz 2, which has greater spread, the 75th percentile is 9, the 25th percentile is 5, and the interquartile range is 4. Recall that in the discussion of box plots, the 75th percentile was called the upper hinge and the 25th percentile was called the lower hinge. Using this terminology, the interquartile range is referred to as the H-spread.

A related measure of variability is called the semi-interquartile range. The semi-interquartile range is defined simply as the interquartile range divided by 2. If a distribution is symmetric, the median plus or minus the semi-interquartile range contains half the scores in the distribution.

Variance

Variability can also be defined in terms of how close the scores in the distribution are to the middle of the distribution. Using the mean as the measure of the middle of the distribution, the variance is defined as the average squared difference of the scores from the mean. The data from Quiz 1 are shown in Table 5.9.1. The mean score is 7.0. Therefore, the column "Deviation from Mean" contains the score minus 7. The column "Squared Deviation" is simply the previous column squared.

Table 5.9.1: Calculation of Variance for Quiz 1 scores

Scores	Deviation from Mean	Squared Deviation
9	2	4
9	2	4
9	2	4
8	1	1
8	1	1
8	1	1
8	1	1
7	0	0
7	0	0
7	0	0
7	0	0
7	0	0
6	-1	1
6	-1	1
6	-1	1
6	-1	1
6	-1	1
6	-1	1
5	-2	4
5	-2	4
Means		
7	0	1.5

One thing that is important to notice is that the mean deviation from the mean is 0. This will always be the case. The mean of the squared deviations is 1.5. Therefore, the variance is 1.5. Analogous calculations with Quiz 2 show that its variance is 6.7. The formula for the variance is:

$$\sigma^2 = \frac{\sum(X-\mu)^2}{N} \tag{5.9.2}$$

where σ^2 is the variance, μ is the mean, and N is the number of numbers. For Quiz 1, $\mu = 7$ and $N = 20$.

If the variance in a sample is used to estimate the variance in a population, then the previous formula underestimates the variance and the following formula should be used:

$$s^2 = \frac{\sum(X-M)^2}{N-1} \tag{5.9.3}$$

where s^2 is the estimate of the variance and M is the sample mean. Note that M is the mean of a sample taken from a population with a mean of μ. Since, in practice, the variance is usually computed in a sample, this formula is most often used. The simulation "estimating variance" illustrates the bias in the formula with N in the denominator.

Let's take a concrete example. Assume the scores $1, 2, 4, \text{ and } 5$ were sampled from a larger population. To estimate the variance in the population you would compute s^2 as follows:

$$M = \frac{1+2+4+5}{4} = \frac{12}{4} = 3 \tag{5.9.4}$$

$$\begin{aligned} s^2 &= \frac{[(1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2]}{(4-1)} \\ &= \frac{(4+1+1+4)}{3} \\ &= \frac{10}{3} \\ &= 3.333 \end{aligned}$$

There are alternate formulas that can be easier to use if you are doing your calculations with a hand calculator. You should note that these formulas are subject to rounding error if your values are very large and/or you have an extremely large number of observations.

$$\sigma^2 = \frac{\sum X^2 - \frac{(\sum X)^2}{N}}{N} \tag{5.9.5}$$

and

$$s^2 = \frac{\sum X^2 - \frac{(\sum X)^2}{N}}{N-1} \tag{5.9.6}$$

For this example,

$$\sum X^2 = 1^2 + 2^2 + 4^2 + 5^2 = 46 \tag{5.9.7}$$

$$\frac{(\sum X)^2}{N} = \frac{(1+2+4+5)^2}{4} = \frac{144}{4} = 36 \tag{5.9.8}$$

$$\sigma^2 = \frac{(46-36)}{4} = 2.5 \tag{5.9.9}$$

$$s^2 = \frac{(46-36)}{3} = 3.333 \text{ as with the other formula} \tag{5.9.10}$$

Standard Deviation

The standard deviation is simply the square root of the variance. This makes the standard deviations of the two quiz distributions 1.225 and 2.588. The standard deviation is an especially useful measure of variability when the distribution is normal or approximately normal (see Chapter on Normal Distributions) because the proportion of the distribution within a given number of standard deviations from the mean can be calculated. For example, 68% of the distribution is within one standard deviation of the mean and approximately 95% of the distribution is within two standard deviations of the mean. Therefore, if you had a normal distribution with a mean of 50 and a standard deviation of 10, then 68% of the distribution would be between $50 - 10 = 40$ and $50 + 10 = 60$. Similarly, about 95% of the distribution would be between $50 - 2 \times 10 = 30$ and $50 + 2 \times 10 = 70$. The symbol for the population standard deviation is σ; the symbol for an estimate computed in a sample is s. Figure $5.9.2$ shows two normal distributions. The red distribution has a mean of 40 and a standard deviation of 5; the blue distribution has a mean of 60 and a standard deviation of 10. For the red distribution, 68% of the distribution is between 35 and 45; for the blue distribution, 68% is between 50 and 70.

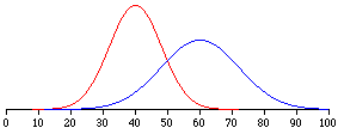


Figure 5.9.2: *Normal distributions with standard deviations of 5 and 10*

This page titled 5.9: Measures of Variability is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **3.12: Measures of Variability** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

5.10: Shapes of Distributions

Learning Objectives

- Compute skew using two different formulas
- Compute kurtosis

We saw in the section on distributions in Chapter 1 that shapes of distributions can differ in skew and/or kurtosis. This section presents numerical indexes of these two measures of shape.

Skew

Figure 5.10.1 shows a distribution with a very large positive skew. Recall that distributions with positive skew have tails that extend to the right.

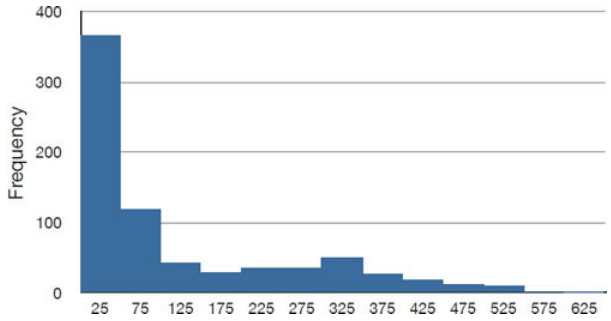


Figure 5.10.1: A distribution with a very large positive skew. This histogram shows the salaries of major league baseball players (in tens of thousands of dollars).

Distributions with positive skew normally have larger means than medians. The mean and median of the baseball salaries shown in Figure 5.10.1 are $\$1,183,417$ and $\$500,000$ respectively. Thus, for this highly-skewed distribution, the mean is more than twice as high as the median. The relationship between skew and the relative size of the mean and median led the statistician Pearson to propose the following simple and convenient numerical index of skew:

$$\frac{3(Mean - Median)}{\sigma} \tag{5.10.1}$$

The standard deviation of the baseball salaries is $1,390,922$. Therefore, Pearson's measure of skew for this distribution is $\frac{3(1,183,417 - 500,000)}{1,390,922} = 1.47$.

Just as there are several measures of central tendency, there is more than one measure of skew. Although Pearson's measure is a good one, the following measure is more commonly used. It is sometimes referred to as the third moment about the mean.

$$\sum \frac{(X - \mu)^3}{\sigma^3} \tag{5.10.2}$$

Kurtosis

The following measure of kurtosis is similar to the definition of skew. The value "3" is subtracted to define "no kurtosis" as the kurtosis of a normal distribution. Otherwise, a normal distribution would have a kurtosis of 3.

$$\sum \frac{(X - \mu)^4}{\sigma^4} - 3 \tag{5.10.3}$$

This page titled 5.10: Shapes of Distributions is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **3.15: Shapes of Distributions** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

5.11: Effects of Linear Transformations

Learning Objectives

- Compute the mean of a transformed variable
- Compute the variance of a transformed variable

This section covers the effects of linear transformations on measures of central tendency and variability. Let's start with an example we saw before in the section that defined linear transformation: temperatures of cities. Table 5.11.1 shows the temperatures of 5 cities.

Table 5.11.1: Temperatures in 5 cities on 11/16/2002

City	Degrees Fahrenheit	Degrees Centigrade
Houston	54	12.22
Chicago	37	2.78
Minneapolis	31	-0.56
Miami	78	25.56
Phoenix	70	21.11
Mean	54.000	12.220
Median	54.000	12.220
Variance	330.00	101.852
SD	18.166	10.092

Recall that to transform the degrees Fahrenheit to degrees Centigrade, we use the formula

$$C = 0.556F - 17.778 \tag{5.11.1}$$

which means we multiply each temperature Fahrenheit by 0.556 and then subtract 17.778. As you might have expected, you multiply the mean temperature in Fahrenheit by 0.556 and then subtract 17.778 to get the mean in Centigrade. That is, $(0.556)(54) - 17.778 = 12.22$. The same is true for the median. Note that this relationship holds even if the mean and median are not identical as they are in Table 5.11.1.

The formula for the standard deviation is just as simple: the standard deviation in degrees Centigrade is equal to the standard deviation in degrees Fahrenheit times 0.556. Since the variance is the standard deviation squared, the variance in degrees Centigrade is equal to 0.5562^2 times the variance in degrees Fahrenheit.

To sum up, if a variable X has a mean of μ, a standard deviation of σ, and a variance of σ^2, then a new variable Y created using the linear transformation

$$Y = bX + A \tag{5.11.2}$$

will have a mean of $b\mu + A$, a standard deviation of $b\sigma$, and a variance of $b^2\sigma^2$.

It should be noted that the term "linear transformation" is defined differently in the field of linear algebra. For details, follow this link.

This page titled 5.11: Effects of Linear Transformations is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **3.17: Effects of Linear Transformations** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

5.12: Variance Sum Law I - Uncorrelated Variables

Learning Objectives

- Compute the variance of the sum of two uncorrelated variables
- Compute the variance of the difference between two uncorrelated variables

As you will see in later sections, there are many occasions in which it is important to know the variance of the sum of two variables. Consider the following situation:

a. you have two populations,
b. you sample one number from each population, and
c. you add the two numbers together.

The question is, "What is the variance of this sum?" For example, suppose the two populations are the populations of 8-year-old males and 8-year-old females in Houston, Texas, and that the variable of interest is memory span. You repeat the following steps thousands of times:

1. sample one male and one female
2. measure the memory span of each
3. sum the two memory spans.

After you have done this thousands of times, you compute the variance of the sum. It turns out that the variance of this sum can be computed according to the following formula:

$$\sigma^2_{sum} = \sigma^2_M + \sigma^2_F \tag{5.12.1}$$

where the first term is the variance of the sum, the second term is the variance of the males and the third term is the variance of the females. Therefore, if the variances on the memory span test for the males and females were 0.9 and 0.8 respectively, then the variance of the sum would be 1.7.

The formula for the variance of the difference between the two variables (memory span in this example) is shown below. Notice that the expression for the difference is the same as the formula for the sum.

$$\sigma^2_{difference} = \sigma^2_M + \sigma^2_F \tag{5.12.2}$$

More generally, the variance sum law can be written as follows:

$$\sigma^2_{Z \pm Y} = \sigma^2_X + \sigma^2_Y \tag{5.12.3}$$

which is read: The variance of X plus or minus Y is equal to the variance of X plus the variance of Y.

These formulas for the sum and difference of variables given above only apply when the variables are independent.

In this example, we have thousands of randomly-paired scores. Since the scores are paired randomly, there is no relationship between the memory span of one member of the pair and the memory span of the other. Therefore the two scores are independent. Contrast this situation with one in which thousands of people are sampled and two measures (such as verbal and quantitative SAT) are taken from each. In this case, there would be a relationship between the two variables since higher scores on the verbal SAT are associated with higher scores on the quantitative SAT (although there are many examples of people who score high on one test and low on the other). Thus the two variables are not independent and the variance of the total SAT score would not be the sum of the variances of the verbal SAT and the quantitative SAT. The general form of the variance sum law is presented in Section 4.7 in the chapter on correlation.

This page titled 5.12: Variance Sum Law I - Uncorrelated Variables is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **3.18: Variance Sum Law I - Uncorrelated Variables** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

5.13: Statistical Literacy

Learning Objectives

- How to select between Mean and Median

The Mean or the Median?

Example 5.13.1

The playbill for the Alley Theatre in Houston wants to appeal to advertisers. They reported the mean household income and the median age of theatergoers. What might have guided their choice of the mean or median?

Solution

It is likely that they wanted to emphasize that theatergoers had high income but de-emphasize how old they are. The distributions of income and age of theatergoers probably have positive skew. Therefore the mean is probably higher than the median, which results in higher income and lower age than if the median household income and mean age had been presented.

This page titled 5.13: Statistical Literacy is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **3.19: Statistical Literacy** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

5.14: Case Study- Using Stents to Prevent Strokes

Section 1.1 introduces a classic challenge in statistics: evaluating the efficacy of a medical treatment. Terms in this section, and indeed much of this chapter, will all be revisited later in the text. The plan for now is simply to get a sense of the role statistics can play in practice.

In this section we will consider an experiment that studies effectiveness of stents in treating patients at risk of stroke[1]. Stents are devices put inside blood vessels that assist in patient recovery after cardiac events and reduce the risk of an additional heart attack or death. Many doctors have hoped that there would be similar bene ts for patients at risk of stroke. We start by writing the principle question the researchers hope to answer:

[1]*Chimowitz MI, Lynn MJ, Derdeyn CP, et al. 2011. Stenting versus Aggressive Medical Therapy for Intracranial Arterial Stenosis. New England Journal of Medicine 365:993-1003. http://www.nejm.org/doi/full/10.1056/NEJMoa1105335. NY Times article reporting on the study: http://www.nytimes.com/2011/09/08/health/research/08stent.html.*

Does the use of stents reduce the risk of stroke?

The researchers who asked this question collected data on 451 at-risk patients. Each volunteer patient was randomly assigned to one of two groups:

- **Treatment group**. Patients in the treatment group received a stent and medical management. The medical management included medications, management of risk factors, and help in lifestyle modi cation.
- **Control group**. Patients in the control group received the same medical manage-ment as the treatment group, but they did not receive stents.

Researchers randomly assigned 224 patients to the treatment group and 227 to the control group. In this study, the control group provides a reference point against which we can measure the medical impact of stents in the treatment group.

Researchers studied the effect of stents at two time points: 30 days after enrollment and 365 days after enrollment. The results of 5 patients are summarized in Table 1.1. Patient outcomes are recorded as "stroke" or "no event", representing whether or not the patient had a stroke at the end of a time period.

Table 1.1: Results for ve patients from the stent study.

Patient	group	0-30 days	0-365 days
1	treatment	no event	no event
2	treatment	stroke	stroke
3	treatment	no event	no event
..	..	..	..
450	control	no event	no event
451	control	no event	no event

Considering data from each patient individually would be a long, cumbersome path towards answering the original research question. Instead, performing a statistical data analysis allows us to consider all of the data at once. Table 1.2 summarizes the raw data in a more helpful way. In this table, we can quickly see what happened over the entire study. For instance, to identify the number of patients in the treatment group who had a stroke within 30 days, we look on the left-side of the table at the intersection of the treatment and stroke: 33.

Table 1.2: Descriptive statistics for the stent study.

	0-30 days		0-365 days	
	stroke	no event	stroke	no event
treatment	33	191	45	179
control	13	214	28	199
Total	46	405	73	378

Exercise

Exercise 1.1 Of the 224 patients in the treatment group, 45 had a stroke by the end of the first year. Using these two numbers, compute the proportion of patients in the treatment group who had a stroke by the end of their rst year. (Please note: answers to all in-text exercises are provided using footnotes.)[2]

Answer

[2]*The proportion of the 224 patients who had a stroke within 365 days:* $\frac{45}{224} = 0.20$.

We can compute summary statistics from the table. A **summary statistic** is a single number summarizing a large amount of data (formally, a summary statistic is a value computed from the data. Some summary statistics are more useful than others). For instance, the primary results of the study after 1 year could be described by two summary statistics: the proportion of people who had a stroke in the treatment and control groups.

- Proportion who had a stroke in the treatment (stent) group: $\frac{45}{224} = 0.20 = 20\%$.
- Proportion who had a stroke in the control group: $\frac{28}{227} = 0.12 = 12\%$.

These two summary statistics are useful in looking for differences in the groups, and we are in for a surprise: an additional 8% of patients in the treatment group had a stroke! This is important for two reasons. First, it is contrary to what doctors expected, which was that stents would reduce the rate of strokes. Second, it leads to a statistical question: do the data show a "real" difference between the groups?

This second question is subtle. Suppose you flip a coin 100 times. While the chance a coin lands heads in any given coin flip is 50%, we probably won't observe exactly 50% heads. This type of fluctuation is part of almost any type of data generating process. It is possible that the 8% difference in the stent study is due to this natural variation. However, the larger the difference we observe (for a particular sample size), the less believable it is that the difference is due to chance. So what we are really asking is the following: is the difference so large that we should reject the notion that it was due to chance?

While we don't yet have our statistical tools to fully address this question on our own, we can comprehend the conclusions of the published analysis: there was compelling evidence of harm by stents in this study of stroke patients.

Be careful

Do not generalize the results of this study to all patients and all stents. This study looked at patients with very speci c characteristics who volunteered to be a part of this study and who may not be representative of all stroke patients. In addition, there are many types of stents and this study only considered the self-expanding Wingspan stent (Boston Scientific). However, this study does leave us with an important lesson: we should keep our eyes open for surprises.

This page titled 5.14: Case Study- Using Stents to Prevent Strokes is shared under a CC BY-SA 3.0 license and was authored, remixed, and/or curated by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel via source content that was edited to the style and standards of the LibreTexts platform.

- **1.2: Case Study- Using Stents to Prevent Strokes** by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel is licensed CC BY-SA 3.0. Original source: https://www.openintro.org/book/os.

5.15: Measures of the Location of the Data (Exercises)

Exercise 5.15.10

Listed are 29 ages for Academy Award winning best actors *in order from smallest to largest.*

18; 21; 22; 25; 26; 27; 29; 30; 31; 33; 36; 37; 41; 42; 47; 52; 55; 57; 58; 62; 64; 67; 69; 71; 72; 73; 74; 76; 77

a. Find the 40th percentile.
b. Find the 78th percentile.

Answer

a. The 40th percentile is 37 years.
b. The 78th percentile is 70 years.

Exercise 5.15.11

Listed are 32 ages for Academy Award winning best actors *in order from smallest to largest.*

18; 18; 21; 22; 25; 26; 27; 29; 30; 31; 31; 33; 36; 37; 37; 41; 42; 47; 52; 55; 57; 58; 62; 64; 67; 69; 71; 72; 73; 74; 76; 77

a. Find the percentile of 37.
b. Find the percentile of 72.

Exercise 5.15.12

Jesse was ranked 37th in his graduating class of 180 students. At what percentile is Jesse's ranking?

Answer

Jesse graduated 37th out of a class of 180 students. There are 180 – 37 = 143 students ranked below Jesse. There is one rank of 37.

$x = 143$ and $y = 1$. $\frac{x + 0.5y}{n}(100) = \frac{143 + 0.5(1)}{180}(100) = 79.72$. Jesse's rank of 37 puts him at the 80th percentile.

Exercise 5.15.13

a. For runners in a race, a low time means a faster run. The winners in a race have the shortest running times. Is it more desirable to have a finish time with a high or a low percentile when running a race?
b. The 20th percentile of run times in a particular race is 5.2 minutes. Write a sentence interpreting the 20th percentile in the context of the situation.
c. A bicyclist in the 90th percentile of a bicycle race completed the race in 1 hour and 12 minutes. Is he among the fastest or slowest cyclists in the race? Write a sentence interpreting the 90th percentile in the context of the situation.

Exercise 5.15.14

a. For runners in a race, a higher speed means a faster run. Is it more desirable to have a speed with a high or a low percentile when running a race?
b. The 40th percentile of speeds in a particular race is 7.5 miles per hour. Write a sentence interpreting the 40th percentile in the context of the situation.

Answer

a. For runners in a race it is more desirable to have a high percentile for speed. A high percentile means a higher speed which is faster.

b. 40% of runners ran at speeds of 7.5 miles per hour or less (slower). 60% of runners ran at speeds of 7.5 miles per hour or more (faster).

Exercise 5.15.15

On an exam, would it be more desirable to earn a grade with a high or low percentile? Explain.

Exercise 5.15.16

Mina is waiting in line at the Department of Motor Vehicles (DMV). Her wait time of 32 minutes is the 85th percentile of wait times. Is that good or bad? Write a sentence interpreting the 85th percentile in the context of this situation.

Answer

When waiting in line at the DMV, the 85th percentile would be a long wait time compared to the other people waiting. 85% of people had shorter wait times than Mina. In this context, Mina would prefer a wait time corresponding to a lower percentile. 85% of people at the DMV waited 32 minutes or less. 15% of people at the DMV waited 32 minutes or longer.

Exercise 5.15.17

In a survey collecting data about the salaries earned by recent college graduates, Li found that her salary was in the 78th percentile. Should Li be pleased or upset by this result? Explain.

Exercise 5.15.18

In a study collecting data about the repair costs of damage to automobiles in a certain type of crash tests, a certain model of car had $1,700 in damage and was in the 90th percentile. Should the manufacturer and the consumer be pleased or upset by this result? Explain and write a sentence that interprets the 90th percentile in the context of this problem.

Answer

The manufacturer and the consumer would be upset. This is a large repair cost for the damages, compared to the other cars in the sample. INTERPRETATION: 90% of the crash tested cars had damage repair costs of $1700 or less; only 10% had damage repair costs of $1700 or more.

Exercise 5.15.19

The University of California has two criteria used to set admission standards for freshman to be admitted to a college in the UC system:

a. Students' GPAs and scores on standardized tests (SATs and ACTs) are entered into a formula that calculates an "admissions index" score. The admissions index score is used to set eligibility standards intended to meet the goal of admitting the top 12% of high school students in the state. In this context, what percentile does the top 12% represent?
b. Students whose GPAs are at or above the 96th percentile of all students at their high school are eligible (called eligible in the local context), even if they are not in the top 12% of all students in the state. What percentage of students from each high school are "eligible in the local context"?

Exercise 5.15.20

Suppose that you are buying a house. You and your realtor have determined that the most expensive house you can afford is the 34th percentile. The 34th percentile of housing prices is $240,000 in the town you want to move to. In this town, can you afford 34% of the houses or 66% of the houses?

Answer

You can afford 34% of houses. 66% of the houses are too expensive for your budget. INTERPRETATION: 34% of houses cost $240,000 or less. 66% of houses cost $240,000 or more.

Use Exercise to calculate the following values:

Exercise 5.15.21

First quartile = _____

Exercise 5.15.22

Second quartile = median = 50th percentile = _____

Answer

4

Exercise 5.15.23

Third quartile = _____

Exercise 5.15.24

Interquartile range (*IQR*) = _____ – _____ = _____

Answer

$6 - 4 = 2$

Exercise 5.15.25

10th percentile = _____

Exercise 5.15.26

70th percentile = _____

Answer

6

This page titled 5.15: Measures of the Location of the Data (Exercises) is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by OpenStax via source content that was edited to the style and standards of the LibreTexts platform.

5.E: Summarizing Distributions (Exercises)

General questions

Q1

Make up a dataset of 12 numbers with a positive skew. Use a statistical program to compute the skew. Is the mean larger than the median as it usually is for distributions with a positive skew? What is the value for skew? (relevant section & relevant section)

Q2

Repeat **Q1** only this time make the dataset have a negative skew. (relevant section & relevant section)

Q3

Make up three data sets with 5 numbers each that have: (relevant section & relevant section)

a. the same mean but different standard deviations.
b. the same mean but different medians.
c. the same median but different means.

Q4

Find the mean and median for the following three variables: (relevant section)

A	B	C
8	4	6
5	4	2
7	6	3
1	3	4
3	4	1

(5.E.1)

Q5

A sample of 30 distance scores measured in yards has a mean of 7, a variance of 16, and a standard deviation of 4.

a. You want to convert all your distances from yards to feet, so you multiply each score in the sample by 3. What are the new mean, variance, and standard deviation?
b. You then decide that you only want to look at the distance past a certain point. Thus, after multiplying the original scores by 3, you decide to subtract 4 feet from each of the scores. Now what are the new mean, variance, and standard deviation? (relevant section)

Q6

You recorded the time in seconds it took for 8 participants to solve a puzzle. These times appear below. However, when the data was entered into the statistical program, the score that was supposed to be 22.1 was entered as 21.2. You had calculated the following measures of central tendency: the mean, the median, and the mean trimmed 25%. Which of these measures of central tendency will change when you correct the recording error? (relevant section & relevant section)

15.2
18.8
19.3
19.7
20.2
21.8
22.1
29.4

(5.E.2)

Q7

For the test scores in question **Q6**, which measures of variability (range, standard deviation, variance) would be changed if the **22.1** data point had been erroneously recorded as **21.2**? (relevant section)

Q8

You know the minimum, the maximum, and the 25th, 50th, and 75th percentiles of a distribution. Which of the following measures of central tendency or variability can you determine? (relevant section, relevant section & relevant section)

mean, median, mode, trimean, geometric mean,
range, interquartile range, variance, standard deviation

Q9

a. Find the value (v) for which $\sum (X - v)^2$ is minimized.
b. Find the value (v) for which $\sum |X - v|$ is minimized.

Q10

Your younger brother comes home one day after taking a science test. He says that someone at school told him that "**60%** of the students in the class scored above the median test grade." What is wrong with this statement? What if he said "**60%** of the students scored below the mean?" (relevant section)

Q11

An experiment compared the ability of three groups of participants to remember briefly-presented chess positions. The data are shown below. The numbers represent the number of pieces correctly remembered from three chess positions. Compare the performance of each group. Consider spread as well as central tendency. (relevant section, relevant section & relevant section)

Non-players	Beginners	Tournament players
22.1	32.5	40.1
22.3	37.1	45.6
26.2	39.1	51.2
29.6	40.5	56.4
31.7	45.5	58.1
33.5	51.3	71.1
38.9	52.6	74.9
39.7	55.7	75.9
43.2	55.9	80.3
43.2	57.7	85.3

Q12

True/False: A bimodal distribution has two modes and two medians. (relevant section)

Q13

True/False: The best way to describe a skewed distribution is to report the mean. (relevant section)

Q14

True/False: When plotted on the same graph, a distribution with a mean of **50** and a standard deviation of **10** will look more spread out than will a distribution with a mean of **60** and a standard deviation of **5**. (relevant section)

Q15

Compare the mean, median, trimean in terms of their sensitivity to extreme scores (relevant section).

Q16

If the mean time to respond to a stimulus is much higher than the median time to respond, what can you say about the shape of the distribution of response times? (relevant section)

Q17

A set of numbers is transformed by taking the log base 10 of each number. The mean of the transformed data is 1.65. What is the geometric mean of the untransformed data? (relevant section)

Q18

Which measure of central tendency is most often used for returns on investment?

Q19

The histogram is in balance on the fulcrum. What are the mean, median, and mode of the distribution (approximate where necessary)?

Questions from Case Studies

The following questions are from the Angry Moods (AM) case study.

Q20

(AM#4) Does Anger-Out have a positive skew, a negative skew, or no skew? (relevant section)

Q21

(AM#8) What is the range of the Anger-In scores? What is the interquartile range? (relevant section)

Q22

(AM#12) What is the overall mean Control-Out score? What is the mean Control-Out score for the athletes? What is the mean Control-Out score for the non-athletes? (relevant section)

Q23

(AM#15) What is the variance of the Control-In scores for the athletes? What is the variance of the Control-In scores for the non-athletes? (relevant section)

The following question is from the Flatulence (F) case study.

Q24

(F#2) Based on a histogram of the variable "perday", do you think the mean or median of this variable is larger? Calculate the mean and median to see if you are right. (relevant section & relevant section)

The following questions are from the Stroop (S) case study.

Q25

(S#1) Compute the mean for "words". (relevant section)

Q26

(S#2) Compute the mean and standard deviation for "colors". (relevant section & relevant section)

The following questions are from the Physicians' Reactions (PR) case study.

Q27

(PR#2) What is the mean expected time spent for the average-weight patients? What is the mean expected time spent for the overweight patients? (relevant section)

Q28

(PR#3) What is the difference in means between the groups? By approximately how many standard deviations do the means differ? (relevant section & relevant section)

The following question is from the Smiles and Leniency (SL) case study.

Q29

(SL#2) Find the mean, median, standard deviation, and interquartile range for the leniency scores of each of the four groups. (relevant section & relevant section)

The following questions are from the ADHD Treatment (AT) case study.

Q30

(AT#4) What is the mean number of correct responses of the participants after taking the placebo (0 mg/kg)? (relevant section)

Q31

(AT#7) What are the standard deviation and the interquartile range of the $d0$ condition? (relevant section)

Selected Answers

S4

Variable A: Mean = 4.8, Median = 5

S5

a. Mean = 21, Var = 144, SD = 12

S9

a. 5.2

S22

Non-athletes: 23.2

S23

Athletes: 20.5

S26

Mean = 20.2

S27

Ave. weight: 31.4

S29

False smile group:

Mean = 5.37

Median = 5.50

SD = 1.83

IQR = 3.0

This page titled 5.E: Summarizing Distributions (Exercises) is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **3.E: Summarizing Distributions (Exercises)** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

CHAPTER OVERVIEW

6: Describing Data With Numbers Using R

Any time that you get a new data set to look at, one of the first tasks that you have to do is find ways of summarising the data in a compact, easily understood fashion. This is what ***descriptive statistics*** (as opposed to inferential statistics) is all about. In fact, to many people the term "statistics" is synonymous with descriptive statistics. It is this topic that we'll consider in this chapter, but before going into any details, let's take a moment to get a sense of why we need descriptive statistics. To do this, let's load the `aflsmall.Rdata` file, and use the `who()` function in the `lsr` package to see what variables are stored in the file:

```
load( "./data/aflsmall.Rdata" )
library(lsr)
```

```
## Warning: package 'lsr' was built under R version 3.5.2
```

```
who()
```

```
##    -- Name --       -- Class --    -- Size --
##    afl.finalists    factor         400
##    afl.margins      numeric        176
```

There are two variables here, `afl.finalists` and `afl.margins`. We'll focus a bit on these two variables in this chapter, so I'd better tell you what they are. Unlike most of data sets in this book, these are actually real data, relating to the Australian Football League (AFL)[64] The `afl.margins` variable contains the winning margin (number of points) for all 176 home and away games played during the 2010 season. The `afl.finalists` variable contains the names of all 400 teams that played in all 200 finals matches played during the period 1987 to 2010. Let's have a look at the `afl.margins` variable:

```
print(afl.margins)
```

```
##   [1]  56  31  56   8  32  14  36  56  19   1   3 104  43  44  72   9  28
##  [18]  25  27  55  20  16  16   7  23  40  48  64  22  55  95  15  49  52
##  [35]  50  10  65  12  39  36   3  26  23  20  43 108  53  38   4   8   3
##  [52]  13  66  67  50  61  36  38  29   9  81   3  26  12  36  37  70   1
##  [69]  35  12  50  35   9  54  47   8  47   2  29  61  38  41  23  24   1
##  [86]   9  11  10  29  47  71  38  49  65  18   0  16   9  19  36  60  24
## [103]  25  44  55   3  57  83  84  35   4  35  26  22   2  14  19  30  19
## [120]  68  11  75  48  32  36  39  50  11   0  63  82  26   3  82  73  19
## [137]  33  48   8  10  53  20  71  75  76  54  44   5  22  94  29   8  98
## [154]   9  89   1 101   7  21  52  42  21 116   3  44  29  27  16   6  44
## [171]   3  28  38  29  10  10
```

This output doesn't make it easy to get a sense of what the data are actually saying. Just "looking at the data" isn't a terribly effective way of understanding data. In order to get some idea about what's going on, we need to calculate some descriptive statistics (this chapter) and draw some nice pictures (Chapter 6. Since the descriptive statistics are the easier of the two topics, I'll start with those, but nevertheless I'll show you a histogram of the `afl.margins` data, since it should help you get a sense of what the data we're trying to describe actually look like. But for what it's worth, this histogram – which is shown in Figure 5.1 –

was generated using the `hist()` function. We'll talk a lot more about how to draw histograms in Section 6.3. For now, it's enough to look at the histogram and note that it provides a fairly interpretable representation of the `afl.margins` data.

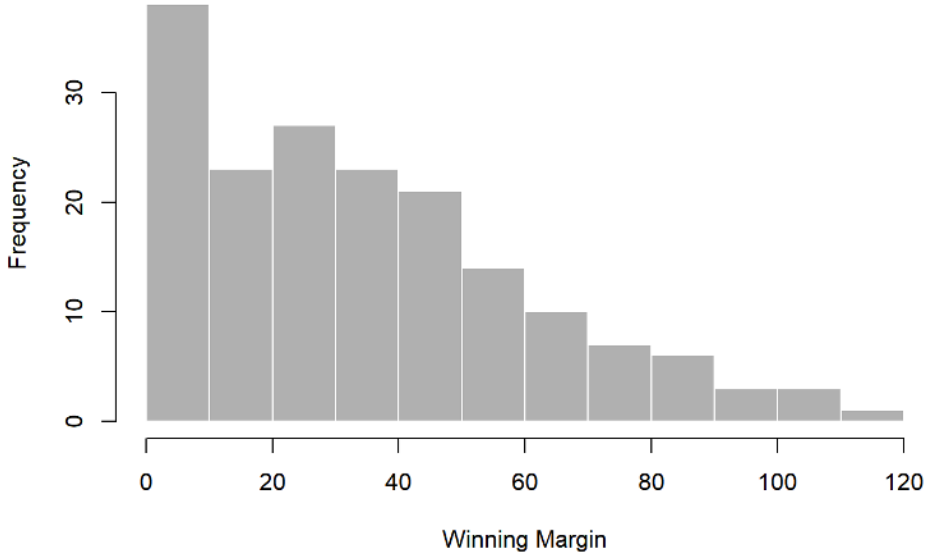


Figure 5.1: A histogram of the AFL 2010 winning margin data (the `afl.margins` variable). As you might expect, the larger the margin the less frequently you tend to see it.

6.1: Measures of Central Tendency
6.2: Measures of Variability
6.3: Skew and Kurtosis
6.4: Getting an Overall Summary of a Variable
6.5: Descriptive Statistics Separately for each Group
6.6: Standard Scores
6.7: Epilogue- Good Descriptive Statistics Are Descriptive!

This page titled 6: Describing Data With Numbers Using R is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

6.1: Measures of Central Tendency

Drawing pictures of the data, as I did in Figure 5.1 is an excellent way to convey the "gist" of what the data is trying to tell you, it's often extremely useful to try to condense the data into a few simple "summary" statistics. In most situations, the first thing that you'll want to calculate is a measure of ***central tendency***. That is, you'd like to know something about the "average" or "middle" of your data lies. The two most commonly used measures are the mean, median and mode; occasionally people will also report a trimmed mean. I'll explain each of these in turn, and then discuss when each of them is useful.

6.1.1 mean

The ***mean*** of a set of observations is just a normal, old-fashioned average: add all of the values up, and then divide by the total number of values. The first five AFL margins were 56, 31, 56, 8 and 32, so the mean of these observations is just:

$$\frac{56+31+56+8+32}{5} = \frac{183}{5} = 36.60$$

Of course, this definition of the mean isn't news to anyone: averages (i.e., means) are used so often in everyday life that this is pretty familiar stuff. However, since the concept of a mean is something that everyone already understands, I'll use this as an excuse to start introducing some of the mathematical notation that statisticians use to describe this calculation, and talk about how the calculations would be done in R.

The first piece of notation to introduce is N, which we'll use to refer to the number of observations that we're averaging (in this case N=5). Next, we need to attach a label to the observations themselves. It's traditional to use X for this, and to use subscripts to indicate which observation we're actually talking about. That is, we'll use X_1 to refer to the first observation, X_2 to refer to the second observation, and so on, all the way up to X_N for the last one. Or, to say the same thing in a slightly more abstract way, we use X_i to refer to the i-th observation. Just to make sure we're clear on the notation, the following table lists the 5 observations in the `afl.margins` variable, along with the mathematical symbol used to refer to it, and the actual value that the observation corresponds to:

The Observation	Its Symbol	The Observed Value
winning margin, game 1	X_1	56 points
winning margin, game 2	X_2	31 points
winning margin, game 3	X_3	56 points
winning margin, game 4	X_4	8 points
winning margin, game 5	X_5	32 points

Okay, now let's try to write a formula for the mean. By tradition, we use $\bar{X}$ as the notation for the mean. So the calculation for the mean could be expressed using the following formula:

$$\bar{X} = \frac{X_1 + X_2 + \ldots + X_{N-1} + X_N}{N}$$

This formula is entirely correct, but it's terribly long, so we make use of the ***summation symbol*** $\sum$ to shorten it.[65] If I want to add up the first five observations, I could write out the sum the long way, X1+X2+X3+X4+X5 or I could use the summation symbol to shorten it to this:

$$\sum_{i=1}^{5} X_i$$

Taken literally, this could be read as "the sum, taken over all i values from 1 to 5, of the value X_i". But basically, what it means is "add up the first five observations". In any case, we can use this notation to write out the formula for the mean, which looks like this:

$$\bar{X} = \frac{1}{N} \sum_{i=1}^{N} X_i$$

In all honesty, I can't imagine that all this mathematical notation helps clarify the concept of the mean at all. In fact, it's really just a fancy way of writing out the same thing I said in words: add all the values up, and then divide by the total number of items. However, that's not really the reason I went into all that detail. My goal was to try to make sure that everyone reading this book is clear on the notation that we'll be using throughout the book: $\bar{X}$ for the mean, $\sum$ for the idea of summation, X_i for the ith observation, and N for the total number of observations. We're going to be re-using these symbols a fair bit, so it's important that you understand them well enough to be able to "read" the equations, and to be able to see that it's just saying "add up lots of things and then divide by another thing".

6.1.2 Calculating the mean in R

Okay that's the maths, how do we get the magic computing box to do the work for us? If you really wanted to, you could do this calculation directly in R. For the first 5 AFL scores, do this just by typing it in as if R were a calculator…

```
(56 + 31 + 56 + 8 + 32) / 5
```

```
## [1] 36.6
```

… in which case R outputs the answer 36.6, just as if it were a calculator. However, that's not the only way to do the calculations, and when the number of observations starts to become large, it's easily the most tedious. Besides, in almost every real world scenario, you've already got the actual numbers stored in a variable of some kind, just like we have with the `afl.margins` variable. Under those circumstances, what you want is a function that will just add up all the values stored in a numeric vector. That's what the `sum()` function does. If we want to add up all 176 winning margins in the data set, we can do so using the following command:[66]

```
sum( afl.margins )
```

```
## [1] 6213
```

If we only want the sum of the first five observations, then we can use square brackets to pull out only the first five elements of the vector. So the command would now be:

```
sum( afl.margins[1:5] )
```

```
## [1] 183
```

To calculate the mean, we now tell R to divide the output of this summation by five, so the command that we need to type now becomes the following:

```
sum( afl.margins[1:5] ) / 5
```

```
## [1] 36.6
```

Although it's pretty easy to calculate the mean using the `sum()` function, we can do it in an even easier way, since R also provides us with the `mean()` function. To calculate the mean for all 176 games, we would use the following command:

```
mean( x = afl.margins )
```

```
## [1] 35.30114
```

However, since `x` is the first argument to the function, I could have omitted the argument name. In any case, just to show you that there's nothing funny going on, here's what we would do to calculate the mean for the first five observations:

```
mean( afl.margins[1:5] )
```

```
## [1] 36.6
```

As you can see, this gives exactly the same answers as the previous calculations.

6.1.3 median

The second measure of central tendency that people use a lot is the ***median***, and it's even easier to describe than the mean. The median of a set of observations is just the middle value. As before let's imagine we were interested only in the first 5 AFL winning margins: 56, 31, 56, 8 and 32. To figure out the median, we sort these numbers into ascending order:

8,31,**32**,56,56

From inspection, it's obvious that the median value of these 5 observations is 32, since that's the middle one in the sorted list (I've put it in bold to make it even more obvious). Easy stuff. But what should we do if we were interested in the first 6 games rather than the first 5? Since the sixth game in the season had a winning margin of 14 points, our sorted list is now

8,14,**31**,32,56,56

and there are *two* middle numbers, 31 and 32. The median is defined as the average of those two numbers, which is of course 31.5. As before, it's very tedious to do this by hand when you've got lots of numbers. To illustrate this, here's what happens when you use R to sort all 176 winning margins. First, I'll use the `sort()` function (discussed in Chapter 7) to display the winning margins in increasing numerical order:

```
sort( x = afl.margins )
```

```
##   [1]   0   0   1   1   1   1   2   2   3   3   3   3   3   3   3   3   4
##  [18]   4   5   6   7   7   8   8   8   8   8   9   9   9   9   9   9  10
##  [35]  10  10  10  10  11  11  11  12  12  12  13  14  14  15  16  16  16
##  [52]  16  18  19  19  19  19  19  20  20  20  21  21  22  22  22  23  23
##  [69]  23  24  24  25  25  26  26  26  26  27  27  28  28  29  29  29  29
##  [86]  29  29  30  31  32  32  33  35  35  35  35  36  36  36  36  36  36
## [103]  37  38  38  38  38  38  39  39  40  41  42  43  43  44  44  44  44
## [120]  44  47  47  47  48  48  48  49  49  50  50  50  50  52  52  53  53
## [137]  54  54  55  55  55  56  56  56  57  60  61  61  63  64  65  65  66
## [154]  67  68  70  71  71  72  73  75  75  76  81  82  82  83  84  89  94
## [171]  95  98 101 104 108 116
```

The middle values are 30 and 31, so the median winning margin for 2010 was 30.5 points. In real life, of course, no-one actually calculates the median by sorting the data and then looking for the middle value. In real life, we use the median command:

```
median( x = afl.margins )
```

```
## [1] 30.5
```

which outputs the median value of 30.5.

6.1.4 Mean or median? What's the difference?

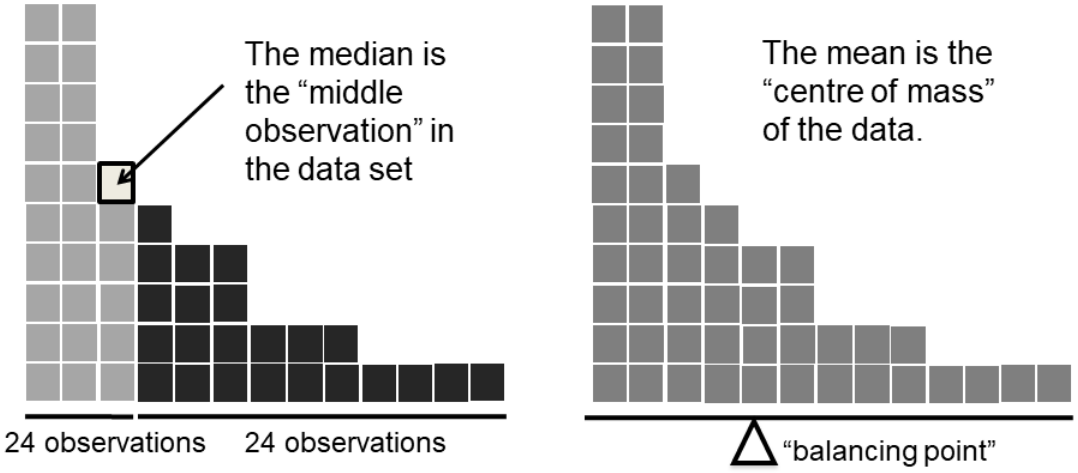


Figure 5.2: An illustration of the difference between how the mean and the median should be interpreted. The mean is basically the "centre of gravity" of the data set: if you imagine that the histogram of the data is a solid object, then the point on which you could balance it (as if on a see-saw) is the mean. In contrast, the median is the middle observation. Half of the observations are smaller, and half of the observations are larger.

Knowing how to calculate means and medians is only a part of the story. You also need to understand what each one is saying about the data, and what that implies for when you should use each one. This is illustrated in Figure 5.2 the mean is kind of like the "centre of gravity" of the data set, whereas the median is the "middle value" in the data. What this implies, as far as which one you should use, depends a little on what type of data you've got and what you're trying to achieve. As a rough guide:

- If your data are nominal scale, you probably shouldn't be using either the mean or the median. Both the mean and the median rely on the idea that the numbers assigned to values are meaningful. If the numbering scheme is arbitrary, then it's probably best to use the mode (Section 5.1.7) instead.
- If your data are ordinal scale, you're more likely to want to use the median than the mean. The median only makes use of the order information in your data (i.e., which numbers are bigger), but doesn't depend on the precise numbers involved. That's exactly the situation that applies when your data are ordinal scale. The mean, on the other hand, makes use of the precise numeric values assigned to the observations, so it's not really appropriate for ordinal data.
- For interval and ratio scale data, either one is generally acceptable. Which one you pick depends a bit on what you're trying to achieve. The mean has the advantage that it uses all the information in the data (which is useful when you don't have a lot of data), but it's very sensitive to extreme values, as we'll see in Section 5.1.6.

Let's expand on that last part a little. One consequence is that there's systematic differences between the mean and the median when the histogram is asymmetric (skewed; see Section 5.3). This is illustrated in Figure 5.2 notice that the median (right hand side) is located closer to the "body" of the histogram, whereas the mean (left hand side) gets dragged towards the "tail" (where the extreme values are). To give a concrete example, suppose Bob (income $50,000), Kate (income $60,000) and Jane (income $65,000) are sitting at a table: the average income at the table is $58,333 and the median income is $60,000. Then Bill sits down with them (income $100,000,000). The average income has now jumped to $25,043,750 but the median rises only to $62,500. If you're interested in looking at the overall income at the table, the mean might be the right answer; but if you're interested in what counts as a typical income at the table, the median would be a better choice here.

6.1.5 real life example

To try to get a sense of why you need to pay attention to the differences between the mean and the median, let's consider a real life example. Since I tend to mock journalists for their poor scientific and statistical knowledge, I should give credit where credit is due. This is from an excellent article on the ABC news website[67] 24 September, 2010:

Senior Commonwealth Bank executives have travelled the world in the past couple of weeks with a presentation showing how Australian house prices, and the key price to income ratios, compare favourably with similar countries. "Housing affordability has

actually been going sideways for the last five to six years," said Craig James, the chief economist of the bank's trading arm, CommSec.

This probably comes as a huge surprise to anyone with a mortgage, or who wants a mortgage, or pays rent, or isn't completely oblivious to what's been going on in the Australian housing market over the last several years. Back to the article:

CBA has waged its war against what it believes are housing doomsayers with graphs, numbers and international comparisons. In its presentation, the bank rejects arguments that Australia's housing is relatively expensive compared to incomes. It says Australia's house price to household income ratio of 5.6 in the major cities, and 4.3 nationwide, is comparable to many other developed nations. It says San Francisco and New York have ratios of 7, Auckland's is 6.7, and Vancouver comes in at 9.3.

More excellent news! Except, the article goes on to make the observation that…

Many analysts say that has led the bank to use misleading figures and comparisons. If you go to page four of CBA's presentation and read the source information at the bottom of the graph and table, you would notice there is an additional source on the international comparison – Demographia. However, if the Commonwealth Bank had also used Demographia's analysis of Australia's house price to income ratio, it would have come up with a figure closer to 9 rather than 5.6 or 4.3

That's, um, a rather serious discrepancy. One group of people say 9, another says 4-5. Should we just split the difference, and say the truth lies somewhere in between? Absolutely not: this is a situation where there is a right answer and a wrong answer. Demographia are correct, and the Commonwealth Bank is incorrect. As the article points out

[An] obvious problem with the Commonwealth Bank's domestic price to income figures is they compare average incomes with median house prices (unlike the Demographia figures that compare median incomes to median prices). The median is the mid-point, effectively cutting out the highs and lows, and that means the average is generally higher when it comes to incomes and asset prices, because it includes the earnings of Australia's wealthiest people. To put it another way: the Commonwealth Bank's figures count Ralph Norris' multi-million dollar pay packet on the income side, but not his (no doubt) very expensive house in the property price figures, thus understating the house price to income ratio for middle-income Australians.

Couldn't have put it better myself. The way that Demographia calculated the ratio is the right thing to do. The way that the Bank did it is incorrect. As for why an extremely quantitatively sophisticated organisation such as a major bank made such an elementary mistake, well… I can't say for sure, since I have no special insight into their thinking, but the article itself does happen to mention the following facts, which may or may not be relevant:

[As] Australia's largest home lender, the Commonwealth Bank has one of the biggest vested interests in house prices rising. It effectively owns a massive swathe of Australian housing as security for its home loans as well as many small business loans.

My, my.

6.1.6 Trimmed mean

One of the fundamental rules of applied statistics is that the data are messy. Real life is never simple, and so the data sets that you obtain are never as straightforward as the statistical theory says.[68] This can have awkward consequences. To illustrate, consider this rather strange looking data set:

$$-100,2,3,4,5,6,7,8,9,10$$

If you were to observe this in a real life data set, you'd probably suspect that something funny was going on with the −100 value. It's probably an ***outlier***, a value that doesn't really belong with the others. You might consider removing it from the data set entirely, and in this particular case I'd probably agree with that course of action. In real life, however, you don't always get such cut-and-dried examples. For instance, you might get this instead:

$$-15,2,3,4,5,6,7,8,9,12$$

The −15 looks a bit suspicious, but not anywhere near as much as that −100 did. In this case, it's a little trickier. It *might* be a legitimate observation, it might not.

When faced with a situation where some of the most extreme-valued observations might not be quite trustworthy, the mean is not necessarily a good measure of central tendency. It is highly sensitive to one or two extreme values, and is thus not considered to be a ***robust*** measure. One remedy that we've seen is to use the median. A more general solution is to use a "trimmed mean". To

calculate a trimmed mean, what you do is "discard" the most extreme examples on both ends (i.e., the largest and the smallest), and then take the mean of everything else. The goal is to preserve the best characteristics of the mean and the median: just like a median, you aren't highly influenced by extreme outliers, but like the mean, you "use" more than one of the observations. Generally, we describe a trimmed mean in terms of the percentage of observation on either side that are discarded. So, for instance, a 10% trimmed mean discards the largest 10% of the observations *and* the smallest 10% of the observations, and then takes the mean of the remaining 80% of the observations. Not surprisingly, the 0% trimmed mean is just the regular mean, and the 50% trimmed mean is the median. In that sense, trimmed means provide a whole family of central tendency measures that span the range from the mean to the median.

For our toy example above, we have 10 observations, and so a 10% trimmed mean is calculated by ignoring the largest value (i.e., `12`) and the smallest value (i.e., `-15`) and taking the mean of the remaining values. First, let's enter the data

```
dataset <- c( -15,2,3,4,5,6,7,8,9,12 )
```

Next, let's calculate means and medians:

```
mean( x = dataset )
```

```
## [1] 4.1
```

```
median( x = dataset )
```

```
## [1] 5.5
```

That's a fairly substantial difference, but I'm tempted to think that the mean is being influenced a bit too much by the extreme values at either end of the data set, especially the −15 one. So let's just try trimming the mean a bit. If I take a 10% trimmed mean, we'll drop the extreme values on either side, and take the mean of the rest:

```
mean( x = dataset, trim = .1)
```

```
## [1] 5.5
```

which in this case gives exactly the same answer as the median. Note that, to get a 10% trimmed mean you write `trim = .1`, not `trim = 10`. In any case, let's finish up by calculating the 5% trimmed mean for the `afl.margins` data,

```
mean( x = afl.margins, trim = .05)
```

```
## [1] 33.75
```

6.1.7 Mode

The mode of a sample is very simple: it is the value that occurs most frequently. To illustrate the mode using the AFL data, let's examine a different aspect to the data set. Who has played in the most finals? The `afl.finalists` variable is a factor that contains the name of every team that played in any AFL final from 1987-2010, so let's have a look at it. To do this we will use the `head()` command. `head()` is useful when you're working with a data.frame with a lot of rows since you can use it to tell you how many rows to return. There have been a lot of finals in this period so printing afl.finalists using `print(afl.finalists)` will just fill us the screen. The command below tells R we just want the first 25 rows of the data.frame.

```
head(afl.finalists, 25)
```

```
##  [1] Hawthorn    Melbourne   Carlton     Melbourne   Hawthorn
##  [6] Carlton     Melbourne   Carlton     Hawthorn    Melbourne
## [11] Melbourne   Hawthorn    Melbourne   Essendon    Hawthorn
## [16] Geelong     Geelong     Hawthorn    Collingwood Melbourne
## [21] Collingwood West Coast  Collingwood Essendon    Collingwood
## 17 Levels: Adelaide Brisbane Carlton Collingwood Essendon ... Western Bulldogs
```

There are actually 400 entries (aren't you glad we didn't print them all?). We *could* read through all 400, and count the number of occasions on which each team name appears in our list of finalists, thereby producing a ***frequency table***. However, that would be mindless and boring: exactly the sort of task that computers are great at. So let's use the `table()` function (discussed in more detail in Section 7.1) to do this task for us:

```
table( afl.finalists )
```

```
## afl.finalists
##         Adelaide         Brisbane          Carlton      Collingwood
##               26               25               26               28
##         Essendon          Fitzroy        Fremantle          Geelong
##               32                0                6               39
##         Hawthorn        Melbourne  North Melbourne    Port Adelaide
##               27               28               28               17
##         Richmond         St Kilda           Sydney       West Coast
##                6               24               26               38
## Western Bulldogs
##               24
```

Now that we have our frequency table, we can just look at it and see that, over the 24 years for which we have data, Geelong has played in more finals than any other team. Thus, the mode of the `finalists` data is `"Geelong"`. The core packages in R don't have a function for calculating the mode[69]. However, I've included a function in the `lsr` package that does this. The function is called `modeOf()`, and here's how you use it:

```
modeOf( x = afl.finalists )
```

```
## [1] "Geelong"
```

There's also a function called `maxFreq()` that tells you what the modal frequency is. If we apply this function to our finalists data, we obtain the following:

```
maxFreq( x = afl.finalists )
```

```
## [1] 39
```

Taken together, we observe that Geelong (39 finals) played in more finals than any other team during the 1987-2010 period.

One last point to make with respect to the mode. While it's generally true that the mode is most often calculated when you have nominal scale data (because means and medians are useless for those sorts of variables), there are some situations in which you

really do want to know the mode of an ordinal, interval or ratio scale variable. For instance, let's go back to thinking about our `afl.margins` variable. This variable is clearly ratio scale (if it's not clear to you, it may help to re-read Section 2.2), and so in most situations the mean or the median is the measure of central tendency that you want. But consider this scenario… a friend of yours is offering a bet. They pick a football game at random, and (without knowing who is playing) you have to guess the *exact* margin. If you guess correctly, you win $50. If you don't, you lose $1. There are no consolation prizes for "almost" getting the right answer. You have to guess exactly the right margin[70] For this bet, the mean and the median are completely useless to you. It is the mode that you should bet on. So, we calculate this modal value

```
modeOf( x = afl.margins )
```

```
## [1] 3
```

```
maxFreq( x = afl.margins )
```

```
## [1] 8
```

So the 2010 data suggest you should bet on a 3 point margin, and since this was observed in 8 of the 176 game (4.5% of games) the odds are firmly in your favour.

This page titled 6.1: Measures of Central Tendency is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **5.1: Measures of Central Tendency** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

6.2: Measures of Variability

The statistics that we've discussed so far all relate to *central tendency*. That is, they all talk about which values are "in the middle" or "popular" in the data. However, central tendency is not the only type of summary statistic that we want to calculate. The second thing that we really want is a measure of the ***variability*** of the data. That is, how "spread out" are the data? How "far" away from the mean or median do the observed values tend to be? For now, let's assume that the data are interval or ratio scale, so we'll continue to use the `afl.margins` data. We'll use this data to discuss several different measures of spread, each with different strengths and weaknesses.

6.2.1 Range

The ***range*** of a variable is very simple: it's the biggest value minus the smallest value. For the AFL winning margins data, the maximum value is 116, and the minimum value is 0. We can calculate these values in R using the `max()` and `min()` functions:

```
max( afl.margins )
```

```
## [1] 116
```

```
min( afl.margins )
```

```
## [1] 0
```

where I've omitted the output because it's not interesting. The other possibility is to use the `range()` function; which outputs both the minimum value and the maximum value in a vector, like this:

```
range( afl.margins )
```

```
## [1]   0 116
```

Although the range is the simplest way to quantify the notion of "variability", it's one of the worst. Recall from our discussion of the mean that we want our summary measure to be robust. If the data set has one or two extremely bad values in it, we'd like our statistics not to be unduly influenced by these cases. If we look once again at our toy example of a data set containing very extreme outliers…

$$-100,2,3,4,5,6,7,8,9,10$$

… it is clear that the range is not robust, since this has a range of 110, but if the outlier were removed we would have a range of only 8.

6.2.2 Interquartile range

The ***interquartile range*** (IQR) is like the range, but instead of calculating the difference between the biggest and smallest value, it calculates the difference between the 25th quantile and the 75th quantile. Probably you already know what a ***quantile*** is (they're more commonly called percentiles), but if not: the 10th percentile of a data set is the smallest number x such that 10% of the data is less than x. In fact, we've already come across the idea: the median of a data set is its 50th quantile / percentile! R actually provides you with a way of calculating quantiles, using the (surprise, surprise) `quantile()` function. Let's use it to calculate the median AFL winning margin:

```
quantile( x = afl.margins, probs = .5)
```

```
##  50%
## 30.5
```

And not surprisingly, this agrees with the answer that we saw earlier with the `median()` function. Now, we can actually input lots of quantiles at once, by specifying a vector for the `probs` argument. So lets do that, and get the 25th and 75th percentile:

```
quantile( x = afl.margins, probs = c(.25,.75) )
```

```
##   25%   75%
## 12.75 50.50
```

And, by noting that 50.5−12.75=37.75, we can see that the interquartile range for the 2010 AFL winning margins data is 37.75. Of course, that seems like too much work to do all that typing, so R has a built in function called `IQR()` that we can use:

```
IQR( x = afl.margins )
```

```
## [1] 37.75
```

While it's obvious how to interpret the range, it's a little less obvious how to interpret the IQR. The simplest way to think about it is like this: the interquartile range is the range spanned by the "middle half" of the data. That is, one quarter of the data falls below the 25th percentile, one quarter of the data is above the 75th percentile, leaving the "middle half" of the data lying in between the two. And the IQR is the range covered by that middle half.

6.2.3 Mean absolute deviation

The two measures we've looked at so far, the range and the interquartile range, both rely on the idea that we can measure the spread of the data by looking at the quantiles of the data. However, this isn't the only way to think about the problem. A different approach is to select a meaningful reference point (usually the mean or the median) and then report the "typical" deviations from that reference point. What do we mean by "typical" deviation? Usually, the mean or median value of these deviations! In practice, this leads to two different measures, the "mean absolute deviation (from the mean)" and the "median absolute deviation (from the median)". From what I've read, the measure based on the median seems to be used in statistics, and does seem to be the better of the two, but to be honest I don't think I've seen it used much in psychology. The measure based on the mean does occasionally show up in psychology though. In this section I'll talk about the first one, and I'll come back to talk about the second one later.

Since the previous paragraph might sound a little abstract, let's go through the ***mean absolute deviation*** from the mean a little more slowly. One useful thing about this measure is that the name actually tells you exactly how to calculate it. Let's think about our AFL winning margins data, and once again we'll start by pretending that there's only 5 games in total, with winning margins of 56, 31, 56, 8 and 32. Since our calculations rely on an examination of the deviation from some reference point (in this case the mean), the first thing we need to calculate is the mean, $\bar{X}$. For these five observations, our mean is $\bar{X}$=36.6. The next step is to convert each of our observations X_i into a deviation score. We do this by calculating the difference between the observation Xi and the mean $\bar{X}$. That is, the deviation score is defined to be $X_i - \bar{X}$. For the first observation in our sample, this is equal to 56−36.6=19.4. Okay, that's simple enough. The next step in the process is to convert these deviations to absolute deviations. As we discussed earlier when talking about the `abs()` function in R (Section 3.5), we do this by converting any negative values to positive ones. Mathematically, we would denote the absolute value of −3 as |−3|, and so we say that |−3|=3. We use the absolute value function here because we don't really care whether the value is higher than the mean or lower than the mean, we're just interested in how *close* it is to the mean. To help make this process as obvious as possible, the table below shows these calculations for all five observations:

the observation	its symbol	the observed value
winning margin, game 2	X2	31 points
winning margin, game 5	X5	32 points
winning margin, game 1	X1	56 points
winning margin, game 3	X3	56 points
winning margin, game 4	X4	8 points

Now that we have calculated the absolute deviation score for every observation in the data set, all that we have to do to calculate the mean of these scores. Let's do that:

$$\frac{19.4 + 5.6 + 19.4 + 28.6 + 4.6}{5} = 15.52$$

And we're done. The mean absolute deviation for these five scores is 15.52.

However, while our calculations for this little example are at an end, we do have a couple of things left to talk about. Firstly, we should really try to write down a proper mathematical formula. But in order do to this I need some mathematical notation to refer to the mean absolute deviation. Irritatingly, "mean absolute deviation" and "median absolute deviation" have the same acronym (MAD), which leads to a certain amount of ambiguity, and since R tends to use MAD to refer to the median absolute deviation, I'd better come up with something different for the mean absolute deviation. Sigh. What I'll do is use AAD instead, short for *average* absolute deviation. Now that we have some unambiguous notation, here's the formula that describes what we just calculated:

$$(X) = \frac{1}{N}\sum_{i=1}^{N}|X_i - \bar{X}|$$

The last thing we need to talk about is how to calculate AAD in R. One possibility would be to do everything using low level commands, laboriously following the same steps that I used when describing the calculations above. However, that's pretty tedious. You'd end up with a series of commands that might look like this:

```
X <- c(56, 31,56,8,32)   # enter the data
X.bar <- mean( X )       # step 1. the mean of the data
AD <- abs( X - X.bar )   # step 2. the absolute deviations from the mean
AAD <- mean( AD )        # step 3. the mean absolute deviations
print( AAD )             # print the results
```

```
## [1] 15.52
```

Each of those commands is pretty simple, but there's just too many of them. And because I find that to be too much typing, the `lsr` package has a very simple function called `aad()` that does the calculations for you. If we apply the `aad()` function to our data, we get this:

```
library(lsr)
aad( X )
```

```
## [1] 15.52
```

No suprises there.

6.2.4 Variance

Although the mean absolute deviation measure has its uses, it's not the best measure of variability to use. From a purely mathematical perspective, there are some solid reasons to prefer squared deviations rather than absolute deviations. If we do that, we obtain a measure is called the ***variance***, which has a lot of really nice statistical properties that I'm going to ignore,[71](X)$ and Var(Y) respectively. Now imagine I want to define a new variable Z that is the sum of the two, Z=X+Y. As it turns out, the variance of Z is equal to Var(X)+Var(Y). This is a *very* useful property, but it's not true of the other measures that I talk about in this section.] and one massive psychological flaw that I'm going to make a big deal out of in a moment. The variance of a data set X is sometimes written as Var(X), but it's more commonly denoted s^2 (the reason for this will become clearer shortly). The formula that we use to calculate the variance of a set of observations is as follows:

$$\mathrm{Var}(X) = \frac{1}{N}\sum_{i=1}^{N}\left(X_i - \bar{X}\right)^2$$

$$\mathrm{Var}(X) = \frac{\sum_{i=1}^{N}\left(X_i - \bar{X}\right)^2}{N}$$

As you can see, it's basically the same formula that we used to calculate the mean absolute deviation, except that instead of using "absolute deviations" we use "squared deviations". It is for this reason that the variance is sometimes referred to as the "mean square deviation".

Now that we've got the basic idea, let's have a look at a concrete example. Once again, let's use the first five AFL games as our data. If we follow the same approach that we took last time, we end up with the following table:

Table 5.1: Basic arithmetic operations in R. These five operators are used very frequently throughout the text, so it's important to be familiar with them at the outset.

Notation [English]	i [which game]	X_i [value]	$X_i - \bar{X}$ [deviation from mean]	$(Xi - \bar{X})^2$ [absolute deviation]
	5	32	-4.6	21.16
	2	31	-5.6	31.36
	1	56	19.4	376.36
	3	56	19.4	376.36
	4	8	-28.6	817.96

That last column contains all of our squared deviations, so all we have to do is average them. If we do that by typing all the numbers into R by hand…

```
( 376.36 + 31.36 + 376.36 + 817.96 + 21.16 ) / 5
```

```
## [1] 324.64
```

… we end up with a variance of 324.64. Exciting, isn't it? For the moment, let's ignore the burning question that you're all probably thinking (i.e., what the heck does a variance of 324.64 actually mean?) and instead talk a bit more about how to do the calculations in R, because this will reveal something very weird.

As always, we want to avoid having to type in a whole lot of numbers ourselves. And as it happens, we have the vector `X` lying around, which we created in the previous section. With this in mind, we can calculate the variance of `X` by using the following command,

```
mean( (X - mean(X) )^2)
```

```
## [1] 324.64
```

and as usual we get the same answer as the one that we got when we did everything by hand. However, I *still* think that this is too much typing. Fortunately, R has a built in function called `var()` which does calculate variances. So we could also do this…

```
var(X)
```

```
## [1] 405.8
```

and you get the same… no, wait… you get a completely *different* answer. That's just weird. Is R broken? Is this a typo? Is Dan an idiot?

As it happens, the answer is no.[72] It's not a typo, and R is not making a mistake. To get a feel for what's happening, let's stop using the tiny data set containing only 5 data points, and switch to the full set of 176 games that we've got stored in our `afl.margins` vector. First, let's calculate the variance by using the formula that I described above:

```
mean( (afl.margins - mean(afl.margins) )^2)
```

```
## [1] 675.9718
```

Now let's use the `var()` function:

```
var( afl.margins )
```

```
## [1] 679.8345
```

Hm. These two numbers are very similar this time. That seems like too much of a coincidence to be a mistake. And of course it isn't a mistake. In fact, it's very simple to explain what R is doing here, but slightly trickier to explain *why* R is doing it. So let's start with the "what". What R is doing is evaluating a slightly different formula to the one I showed you above. Instead of averaging the squared deviations, which requires you to divide by the number of data points N, R has chosen to divide by N−1. In other words, the formula that R is using is this one

$$\frac{1}{N-1}\sum_{i=1}^{N}\left(X_i - \bar{X}\right)^2$$

It's easy enough to verify that this is what's happening, as the following command illustrates:

```
sum( (X-mean(X))^2 ) / 4
```

```
## [1] 405.8
```

This is the same answer that R gave us originally when we calculated `var(X)` originally. So that's the *what*. The real question is *why* R is dividing by N−1 and not by N. After all, the variance is supposed to be the *mean* squared deviation, right? So shouldn't we be dividing by N, the actual number of observations in the sample? Well, yes, we should. However, as we'll discuss in Chapter 10, there's a subtle distinction between "describing a sample" and "making guesses about the population from which the sample came". Up to this point, it's been a distinction without a difference. Regardless of whether you're describing a sample or drawing inferences about the population, the mean is calculated exactly the same way. Not so for the variance, or the standard deviation, or for many other measures besides. What I outlined to you initially (i.e., take the actual average, and thus divide by N) assumes that you literally intend to calculate the variance of the sample. Most of the time, however, you're not terribly interested in the sample

in and of itself. Rather, the sample exists to tell you something about the world. If so, you're actually starting to move away from calculating a "sample statistic", and towards the idea of estimating a "population parameter". However, I'm getting ahead of myself. For now, let's just take it on faith that R knows what it's doing, and we'll revisit the question later on when we talk about estimation in Chapter 10.

Okay, one last thing. This section so far has read a bit like a mystery novel. I've shown you how to calculate the variance, described the weird "N−1" thing that R does and hinted at the reason why it's there, but I haven't mentioned the single most important thing… how do you *interpret* the variance? Descriptive statistics are supposed to describe things, after all, and right now the variance is really just a gibberish number. Unfortunately, the reason why I haven't given you the human-friendly interpretation of the variance is that there really isn't one. This is the most serious problem with the variance. Although it has some elegant mathematical properties that suggest that it really is a fundamental quantity for expressing variation, it's completely useless if you want to communicate with an actual human… variances are completely uninterpretable in terms of the original variable! All the numbers have been squared, and they don't mean anything anymore. This is a huge issue. For instance, according to the table I presented earlier, the margin in game 1 was "376.36 points-squared higher than the average margin". This is *exactly* as stupid as it sounds; and so when we calculate a variance of 324.64, we're in the same situation. I've watched a lot of footy games, and never has anyone referred to "points squared". It's *not* a real unit of measurement, and since the variance is expressed in terms of this gibberish unit, it is totally meaningless to a human.

6.2.5 Standard deviation

Okay, suppose that you like the idea of using the variance because of those nice mathematical properties that I haven't talked about, but – since you're a human and not a robot – you'd like to have a measure that is expressed in the same units as the data itself (i.e., points, not points-squared). What should you do? The solution to the problem is obvious: take the square root of the variance, known as the ***standard deviation***, also called the "root mean squared deviation", or RMSD. This solves out problem fairly neatly: while nobody has a clue what "a variance of 324.68 points-squared" really means, it's much easier to understand "a standard deviation of 18.01 points", since it's expressed in the original units. It is traditional to refer to the standard deviation of a sample of data as s, though "sd" and "std dev." are also used at times. Because the standard deviation is equal to the square root of the variance, you probably won't be surprised to see that the formula is:

$$s = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(X_i - \bar{X}\right)^2}$$

and the R function that we use to calculate it is `sd()`. However, as you might have guessed from our discussion of the variance, what R actually calculates is slightly different to the formula given above. Just like the we saw with the variance, what R calculates is a version that divides by N−1 rather than N. For reasons that will make sense when we return to this topic in Chapter@refch:estimation I'll refer to this new quantity as $\hat{\sigma}$ (read as: "sigma hat"), and the formula for this is

$$\hat{\sigma} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(X_i - \bar{X}\right)^2}$$

With that in mind, calculating standard deviations in R is simple:

```
sd( afl.margins )
```

```
## [1] 26.07364
```

Interpreting standard deviations is slightly more complex. Because the standard deviation is derived from the variance, and the variance is a quantity that has little to no meaning that makes sense to us humans, the standard deviation doesn't have a simple interpretation. As a consequence, most of us just rely on a simple rule of thumb: in general, you should expect 68% of the data to fall within 1 standard deviation of the mean, 95% of the data to fall within 2 standard deviation of the mean, and 99.7% of the data to fall within 3 standard deviations of the mean. This rule tends to work pretty well most of the time, but it's not exact: it's actually calculated based on an *assumption* that the histogram is symmetric and "bell shaped."[73] As you can tell from looking at the AFL

winning margins histogram in Figure 5.1, this isn't exactly true of our data! Even so, the rule is approximately correct. As it turns out, 65.3% of the AFL margins data fall within one standard deviation of the mean. This is shown visually in Figure 5.3.

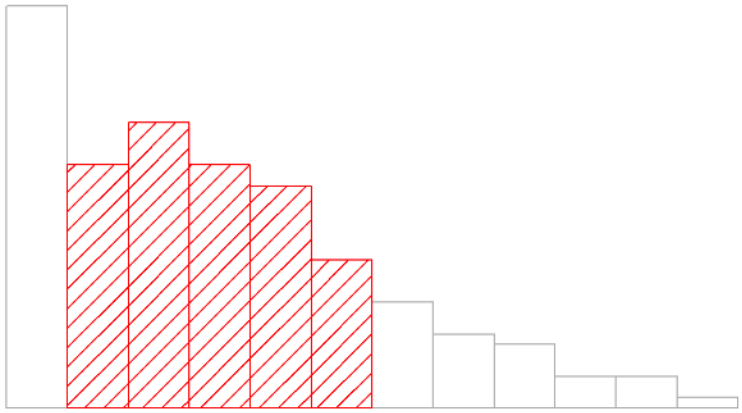


Figure 5.3: An illustration of the standard deviation, applied to the AFL winning margins data. The shaded bars in the histogram show how much of the data fall within one standard deviation of the mean. In this case, 65.3% of the data set lies within this range, which is pretty consistent with the "approximately 68% rule" discussed in the main text.

6.2.6 Median absolute deviation

The last measure of variability that I want to talk about is the ***median absolute deviation*** (MAD). The basic idea behind MAD is very simple, and is pretty much identical to the idea behind the mean absolute deviation (Section 5.2.3). The difference is that you use the median everywhere. If we were to frame this idea as a pair of R commands, they would look like this:

```
# mean absolute deviation from the mean:
mean( abs(afl.margins - mean(afl.margins)) )
```

```
## [1] 21.10124
```

```
# *median* absolute deviation from the *median*:
median( abs(afl.margins - median(afl.margins)) )
```

```
## [1] 19.5
```

This has a straightforward interpretation: every observation in the data set lies some distance away from the typical value (the median). So the MAD is an attempt to describe a *typical deviation from a typical value* in the data set. It wouldn't be unreasonable to interpret the MAD value of 19.5 for our AFL data by saying something like this:

The median winning margin in 2010 was 30.5, indicating that a typical game involved a winning margin of about 30 points. However, there was a fair amount of variation from game to game: the MAD value was 19.5, indicating that a typical winning margin would differ from this median value by about 19-20 points.

As you'd expect, R has a built in function for calculating MAD, and you will be shocked no doubt to hear that it's called `mad()`. However, it's a little bit more complicated than the functions that we've been using previously. If you want to use it to calculate MAD in the exact same way that I have described it above, the command that you need to use specifies two arguments: the data set itself `x`, and a `constant` that I'll explain in a moment. For our purposes, the constant is 1, so our command becomes

```
mad( x = afl.margins, constant = 1 )
```

```
## [1] 19.5
```

Apart from the weirdness of having to type that `constant = 1` part, this is pretty straightforward.

Okay, so what exactly is this `constant = 1` argument? I won't go into all the details here, but here's the gist. Although the "raw" MAD value that I've described above is completely interpretable on its own terms, that's not actually how it's used in a lot of real world contexts. Instead, what happens a lot is that the researcher *actually* wants to calculate the standard deviation. However, in the same way that the mean is very sensitive to extreme values, the standard deviation is vulnerable to the exact same issue. So, in much the same way that people sometimes use the median as a "robust" way of calculating "something that is like the mean", it's not uncommon to use MAD as a method for calculating "something that is like the standard deviation". Unfortunately, the *raw* MAD value doesn't do this. Our raw MAD value is 19.5, and our standard deviation was 26.07. However, what some clever person has shown is that, under certain assumptions[74], you can multiply the raw MAD value by 1.4826 and obtain a number that is directly comparable to the standard deviation. As a consequence, the default value of `constant` is 1.4826, and so when you use the `mad()` command without manually setting a value, here's what you get:

```
mad( afl.margins )
```

```
## [1] 28.9107
```

I should point out, though, that if you want to use this "corrected" MAD value as a robust version of the standard deviation, you really are relying on the assumption that the data are (or at least, are "supposed to be" in some sense) symmetric and basically shaped like a bell curve. That's really *not* true for our `afl.margins` data, so in this case I wouldn't try to use the MAD value this way.

6.2.7 Which measure to use?

We've discussed quite a few measures of spread (range, IQR, MAD, variance and standard deviation), and hinted at their strengths and weaknesses. Here's a quick summary:

- *Range*. Gives you the full spread of the data. It's very vulnerable to outliers, and as a consequence it isn't often used unless you have good reasons to care about the extremes in the data.
- *Interquartile range*. Tells you where the "middle half" of the data sits. It's pretty robust, and complements the median nicely. This is used a lot.
- *Mean absolute deviation*. Tells you how far "on average" the observations are from the mean. It's very interpretable, but has a few minor issues (not discussed here) that make it less attractive to statisticians than the standard deviation. Used sometimes, but not often.
- *Variance*. Tells you the average squared deviation from the mean. It's mathematically elegant, and is probably the "right" way to describe variation around the mean, but it's completely uninterpretable because it doesn't use the same units as the data. Almost never used except as a mathematical tool; but it's buried "under the hood" of a very large number of statistical tools.
- *Standard deviation*. This is the square root of the variance. It's fairly elegant mathematically, and it's expressed in the same units as the data so it can be interpreted pretty well. In situations where the mean is the measure of central tendency, this is the default. This is by far the most popular measure of variation.
- *Median absolute deviation*. The typical (i.e., median) deviation from the median value. In the raw form it's simple and interpretable; in the corrected form it's a robust way to estimate the standard deviation, for some kinds of data sets. Not used very often, but it does get reported sometimes.

In short, the IQR and the standard deviation are easily the two most common measures used to report the variability of the data; but there are situations in which the others are used. I've described all of them in this book because there's a fair chance you'll run into most of these somewhere.

This page titled 6.2: Measures of Variability is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **5.2: Measures of Variability** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

6.3: Skew and Kurtosis

There are two more descriptive statistics that you will sometimes see reported in the psychological literature, known as skew and kurtosis. In practice, neither one is used anywhere near as frequently as the measures of central tendency and variability that we've been talking about. Skew is pretty important, so you do see it mentioned a fair bit; but I've actually never seen kurtosis reported in a scientific article to date.

```
## [1] -0.9174977
```

```
## [1] 0.009023979
```

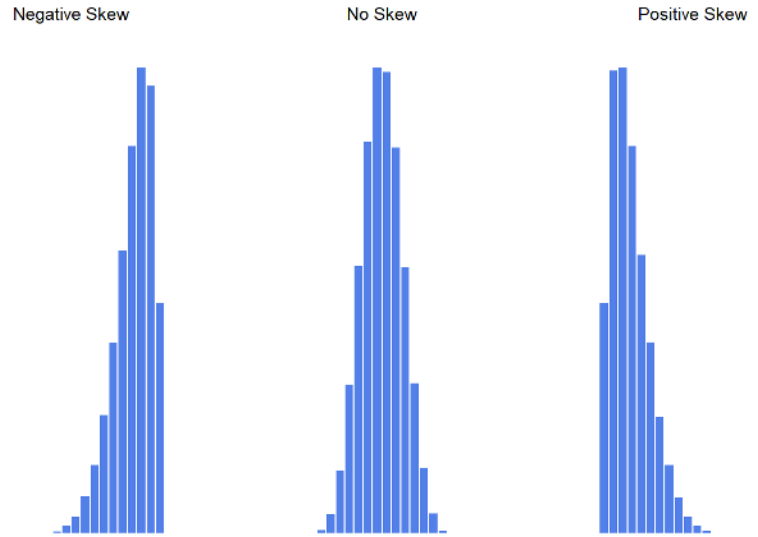


Figure 5.4: An illustration of skewness. On the left we have a negatively skewed data set (skewness =−.93), in the middle we have a data set with no skew (technically, skewness =−.006), and on the right we have a positively skewed data set (skewness =.93).

```
## [1] 0.9250898
```

Since it's the more interesting of the two, let's start by talking about the ***skewness***. Skewness is basically a measure of asymmetry, and the easiest way to explain it is by drawing some pictures. As Figure 5.4 illustrates, if the data tend to have a lot of extreme small values (i.e., the lower tail is "longer" than the upper tail) and not so many extremely large values (left panel), then we say that the data are *negatively skewed*. On the other hand, if there are more extremely large values than extremely small ones (right panel) we say that the data are *positively skewed*. That's the qualitative idea behind skewness. The actual formula for the skewness of a data set is as follows

$$\text{skewness}\ (X) = \frac{1}{N\hat{\sigma}^{\,3}} \sum_{i=1}^{N} \left(X_i - \bar{X}\right)^3 \tag{6.3.1}$$

where N is the number of observations, $\bar{X}$ is the sample mean, and $\hat{\sigma}$ is the standard deviation (the "divide by N−1" version, that is). Perhaps more helpfully, it might be useful to point out that the `psych` package contains a `skew()` function that you can use to calculate skewness. So if we wanted to use this function to calculate the skewness of the `afl.margins` data, we'd first need to load the package

```
library( psych )
```

which now makes it possible to use the following command:

```
skew( x = afl.margins )
```

```
## [1] 0.7671555
```

Not surprisingly, it turns out that the AFL winning margins data is fairly skewed.

The final measure that is sometimes referred to is the **kurtosis** of a data set. Put simply, kurtosis is a measure of the "tailedness", or outlier character, of the data. Historically, it was thought that this statistic measures "pointiness" or "flatness" of a distribution, but this has been shown to be an error of interpretation. See Figure 5.5.

```
## [1] -0.9631805
```

```
## [1] 0.02226287
```

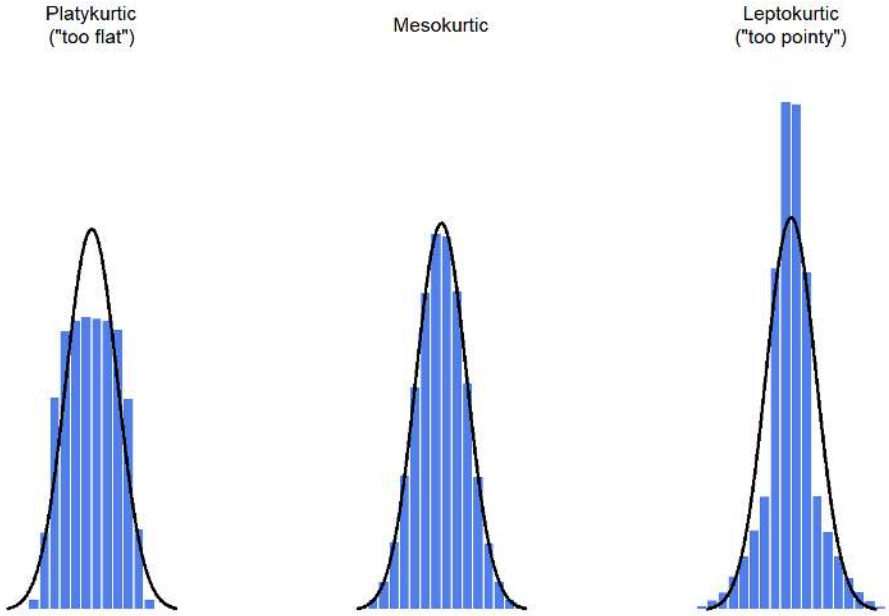


Figure 5.5: An illustration of kurtosis. On the left, we have a "platykurtic" data set (kurtosis = −.25), meaning that the data set has lesser outliers (extreme values) as compared to the standard normal curve (solid line). In the middle we have a "mesokurtic" data set (kurtosis is almost exactly 0), which means that the outlier character of the data set is similar to that of the normal distribution. Finally, on the right, we have a "leptokurtic" data set (kurtosis =6.44) indicating that the data set has more extreme outlier character than the normal distribution. (Note that the outliers are difficult to see in the distribution graphs because the heights at the outliers are so close to zero; a quantile-quantile plot is better to more easily visualize both outliers and kurtosis.)

```
## [1] 1.994329
```

By mathematical calculations, the "normal curve" (black lines) has zero kurtosis, so the outlier character of a data set is assessed relative to this curve. In this Figure, the data on the left are less outlier-prone, so the kurtosis is negative and we call the data *platykurtic*. The data on the right are more outlier-prone, so the kurtosis is positive and we say that the data is *leptokurtic*. But the data in the middle are similar in their outlier character, so we say that it is *mesokurtic* and has kurtosis zero. This is summarised in the table below:

informal term	technical name	kurtosis value
just pointy enough	mesokurtic	zero
too pointy	leptokurtic	positive
too flat	platykurtic	negative

The equation for kurtosis is pretty similar in spirit to the formulas we've seen already for the variance and the skewness (Equation 6.3.1); except that where the variance involved squared deviations and the skewness involved cubed deviations, the kurtosis involves raising the deviations to the fourth power:[75]

$$\text{kurtosis}\,(X) = \frac{1}{N\hat{\sigma}^4}\sum_{i=1}^{N}\left(X_i - \bar{X}\right)^4 - 3 \tag{6.3.2}$$

The `psych` package has a function called `kurtosi()` that you can use to calculate the kurtosis of your data. For instance, if we were to do this for the AFL margins,

```
kurtosi( x = afl.margins )
```

```
## [1] 0.02962633
```

we discover that the AFL winning margins data are just pointy enough.

6.3.1 Contributors

- Danielle Navarro (Associate Professor (Psychology) at University of New South Wales)
- Peter H. Westfall (Paul Whitfield Horn Professor and James and Marguerite Niver Professor, Texas Tech University)

This page titled 6.3: Skew and Kurtosis is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **5.3: Skew and Kurtosis** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

6.4: Getting an Overall Summary of a Variable

Up to this point in the chapter I've explained several different summary statistics that are commonly used when analysing data, along with specific functions that you can use in R to calculate each one. However, it's kind of annoying to have to separately calculate means, medians, standard deviations, skews etc. Wouldn't it be nice if R had some helpful functions that would do all these tedious calculations at once? Something like `summary()` or `describe()`, perhaps? Why yes, yes it would. So much so that both of these functions exist. The `summary()` function is in the `base` package, so it comes with every installation of R. The `describe()` function is part of the `psych` package, which we loaded earlier in the chapter.

6.4.1 "Summarising" a variable

The `summary()` function is an easy thing to use, but a tricky thing to understand in full, since it's a generic function (see Section 4.11. The basic idea behind the `summary()` function is that it prints out some useful information about whatever object (i.e., variable, as far as we're concerned) you specify as the `object` argument. As a consequence, the behaviour of the `summary()` function differs quite dramatically depending on the class of the object that you give it. Let's start by giving it a *numeric* object:

```
summary( object = afl.margins )
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   12.75   30.50   35.30   50.50  116.00
```

For numeric variables, we get a whole bunch of useful descriptive statistics. It gives us the minimum and maximum values (i.e., the range), the first and third quartiles (25th and 75th percentiles; i.e., the IQR), the mean and the median. In other words, it gives us a pretty good collection of descriptive statistics related to the central tendency and the spread of the data.

Okay, what about if we feed it a logical vector instead? Let's say I want to know something about how many "blowouts" there were in the 2010 AFL season. I operationalise the concept of a blowout (see Chapter 2) as a game in which the winning margin exceeds 50 points. Let's create a logical variable `blowouts` in which the i-th element is `TRUE` if that game was a blowout according to my definition,

```
blowouts <-  afl.margins > 50
blowouts
```

```
##   [1]  TRUE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
##  [12]  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
##  [23] FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE FALSE FALSE
##  [34]  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [45] FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE
##  [56]  TRUE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
##  [67]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE
##  [78] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [89] FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE
## [100] FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE FALSE
## [111] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE
## [122]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE
## [133] FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE  TRUE
## [144]  TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE
## [155]  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE
## [166] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

So that's what the `blowouts` variable looks like. Now let's ask R for a `summary()`

```
summary( object = blowouts )
```

```
##    Mode   FALSE    TRUE
## logical     132      44
```

In this context, the `summary()` function gives us a count of the number of `TRUE` values, the number of `FALSE` values, and the number of missing values (i.e., the `NA` s). Pretty reasonable behaviour.

Next, let's try to give it a factor. If you recall, I've defined the `afl.finalists` vector as a factor, so let's use that:

```
summary( object = afl.finalists )
```

```
##         Adelaide         Brisbane          Carlton      Collingwood
##               26               25               26               28
##         Essendon          Fitzroy        Fremantle          Geelong
##               32                0                6               39
##         Hawthorn        Melbourne  North Melbourne    Port Adelaide
##               27               28               28               17
##         Richmond         St Kilda           Sydney       West Coast
##                6               24               26               38
## Western Bulldogs
##               24
```

For factors, we get a frequency table, just like we got when we used the `table()` function. Interestingly, however, if we convert this to a character vector using the `as.character()` function (see Section 7.10, we don't get the same results:

```
f2 <- as.character( afl.finalists )
summary( object = f2 )
```

```
##    Length     Class      Mode
##       400 character character
```

This is one of those situations I was referring to in Section 4.7, in which it is helpful to declare your nominal scale variable as a factor rather than a character vector. Because I've defined `afl.finalists` as a factor, R *knows* that it should treat it as a nominal scale variable, and so it gives you a much more detailed (and helpful) summary than it would have if I'd left it as a character vector.

6.4.2 "Summarising" a data frame

Okay what about data frames? When you pass a data frame to the `summary()` function, it produces a slightly condensed summary of each variable inside the data frame. To give you a sense of how this can be useful, let's try this for a new data set, one that you've never seen before. The data is stored in the `clinicaltrial.Rdata` file, and we'll use it a lot in Chapter 14 (you can find a complete description of the data at the start of that chapter). Let's load it, and see what we've got:

```
load( "./data/clinicaltrial.Rdata" )
who(TRUE)
```

```
##    -- Name --     -- Class --    -- Size --
##    clin.trial     data.frame     18 x 3
##     $drug         factor         18
##     $therapy      factor         18
##     $mood.gain    numeric        18
```

There's a single data frame called `clin.trial` which contains three variables, `drug`, `therapy` and `mood.gain`. Presumably then, this data is from a clinical trial of some kind, in which people were administered different drugs; and the researchers looked to see what the drugs did to their mood. Let's see if the `summary()` function sheds a little more light on this situation:

```
summary( clin.trial )
```

```
##        drug         therapy    mood.gain
##  placebo :6   no.therapy:9   Min.   :0.1000
##  anxifree:6   CBT       :9   1st Qu.:0.4250
##  joyzepam:6                  Median :0.8500
##                              Mean   :0.8833
##                              3rd Qu.:1.3000
##                              Max.   :1.8000
```

Evidently there were three drugs: a placebo, something called "anxifree" and something called "joyzepam"; and there were 6 people administered each drug. There were 9 people treated using cognitive behavioural therapy (CBT) and 9 people who received no psychological treatment. And we can see from looking at the summary of the `mood.gain` variable that most people did show a mood gain (mean =.88), though without knowing what the scale is here it's hard to say much more than that. Still, that's not too bad. Overall, I feel that I learned something from that.

6.4.3 "Describing" a data frame

The `describe()` function (in the `psych` package) is a little different, and it's really only intended to be useful when your data are interval or ratio scale. Unlike the `summary()` function, it calculates the same descriptive statistics for any type of variable you give it. By default, these are:

- `var` . This is just an index: 1 for the first variable, 2 for the second variable, and so on.
- `n` . This is the sample size: more precisely, it's the number of non-missing values.
- `mean` . This is the sample mean (Section 5.1.1).
- `sd` . This is the (bias corrected) standard deviation (Section 5.2.5).
- `median` . The median (Section 5.1.3).
- `trimmed` . This is trimmed mean. By default it's the 10% trimmed mean (Section 5.1.6).
- `mad` . The median absolute deviation (Section 5.2.6).
- `min` . The minimum value.
- `max` . The maximum value.
- `range` . The range spanned by the data (Section 5.2.1).
- `skew` . The skewness (Section 5.3).
- `kurtosis` . The kurtosis (Section 5.3).
- `se` . The standard error of the mean (Chapter 10).

Notice that these descriptive statistics generally only make sense for data that are interval or ratio scale (usually encoded as numeric vectors). For nominal or ordinal variables (usually encoded as factors), most of these descriptive statistics are not all that useful. What the `describe()` function does is convert factors and logical variables to numeric vectors in order to do the

calculations. These variables are marked with `*` and most of the time, the descriptive statistics for those variables won't make much sense. If you try to feed it a data frame that includes a character vector as a variable, it produces an error.

With those caveats in mind, let's use the `describe()` function to have a look at the `clin.trial` data frame. Here's what we get:

```
describe( x = clin.trial )
```

```
##           vars  n mean   sd median trimmed  mad min max range skew
## drug*        1 18 2.00 0.84   2.00    2.00 1.48 1.0 3.0   2.0 0.00
## therapy*     2 18 1.50 0.51   1.50    1.50 0.74 1.0 2.0   1.0 0.00
## mood.gain    3 18 0.88 0.53   0.85    0.88 0.67 0.1 1.8   1.7 0.13
##           kurtosis   se
## drug*        -1.66 0.20
## therapy*     -2.11 0.12
## mood.gain    -1.44 0.13
```

As you can see, the output for the asterisked variables is pretty meaningless, and should be ignored. However, for the `mood.gain` variable, there's a lot of useful information.

This page titled 6.4: Getting an Overall Summary of a Variable is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **5.4: Getting an Overall Summary of a Variable** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

6.5: Descriptive Statistics Separately for each Group

It is very commonly the case that you find yourself needing to look at descriptive statistics, broken down by some grouping variable. This is pretty easy to do in R, and there are three functions in particular that are worth knowing about: `by()`, `describeBy()` and `aggregate()`. Let's start with the `describeBy()` function, which is part of the `psych` package. The `describeBy()` function is very similar to the `describe()` function, except that it has an additional argument called `group` which specifies a grouping variable. For instance, let's say, I want to look at the descriptive statistics for the `clin.trial` data, broken down separately by `therapy` type. The command I would use here is:

```
describeBy( x=clin.trial, group=clin.trial$therapy )
```

```
##
##  Descriptive statistics by group
## group: no.therapy
##           vars n mean   sd median trimmed  mad min max range skew kurtosis
## drug*        1 9 2.00 0.87    2.0    2.00 1.48 1.0 3.0   2.0 0.00    -1.81
## therapy*     2 9 1.00 0.00    1.0    1.00 0.00 1.0 1.0   0.0  NaN      NaN
## mood.gain    3 9 0.72 0.59    0.5    0.72 0.44 0.1 1.7   1.6 0.51    -1.59
##              se
## drug*     0.29
## therapy*  0.00
## mood.gain 0.20
## ------------------------------------------------------------
## group: CBT
##           vars n mean   sd median trimmed  mad min max range  skew
## drug*        1 9 2.00 0.87    2.0    2.00 1.48 1.0 3.0   2.0  0.00
## therapy*     2 9 2.00 0.00    2.0    2.00 0.00 2.0 2.0   0.0   NaN
## mood.gain    3 9 1.04 0.45    1.1    1.04 0.44 0.3 1.8   1.5 -0.03
##           kurtosis   se
## drug*        -1.81 0.29
## therapy*       NaN 0.00
## mood.gain    -1.12 0.15
```

As you can see, the output is essentially identical to the output that the `describe()` function produce, except that the output now gives you means, standard deviations etc separately for the `CBT` group and the `no.therapy` group. Notice that, as before, the output displays asterisks for factor variables, in order to draw your attention to the fact that the descriptive statistics that it has calculated won't be very meaningful for those variables. Nevertheless, this command has given us some really useful descriptive statistics `mood.gain` variable, broken down as a function of `therapy`.

A somewhat more general solution is offered by the `by()` function. There are three arguments that you need to specify when using this function: the `data` argument specifies the data set, the `INDICES` argument specifies the grouping variable, and the `FUN` argument specifies the name of a function that you want to apply separately to each group. To give a sense of how powerful this is, you can reproduce the `describeBy()` function by using a command like this:

```
by( data=clin.trial, INDICES=clin.trial$therapy, FUN=describe )
```

```
## clin.trial$therapy: no.therapy
##           vars n mean   sd median trimmed  mad min max range skew kurtosis
## drug*        1 9 2.00 0.87    2.0    2.00 1.48 1.0 3.0   2.0 0.00    -1.81
## therapy*     2 9 1.00 0.00    1.0    1.00 0.00 1.0 1.0   0.0  NaN      NaN
## mood.gain    3 9 0.72 0.59    0.5    0.72 0.44 0.1 1.7   1.6 0.51    -1.59
##             se
## drug*     0.29
## therapy*  0.00
## mood.gain 0.20
## ------------------------------------------------------------
## clin.trial$therapy: CBT
##           vars n mean   sd median trimmed  mad min max range  skew
## drug*        1 9 2.00 0.87    2.0    2.00 1.48 1.0 3.0   2.0  0.00
## therapy*     2 9 2.00 0.00    2.0    2.00 0.00 2.0 2.0   0.0   NaN
## mood.gain    3 9 1.04 0.45    1.1    1.04 0.44 0.3 1.8   1.5 -0.03
##           kurtosis   se
## drug*        -1.81 0.29
## therapy*       NaN 0.00
## mood.gain    -1.12 0.15
```

This will produce the exact same output as the command shown earlier. However, there's nothing special about the `describe()` function. You could just as easily use the `by()` function in conjunction with the `summary()` function. For example:

```
by( data=clin.trial, INDICES=clin.trial$therapy, FUN=summary )
```

```
## clin.trial$therapy: no.therapy
##        drug         therapy    mood.gain
##  placebo :3   no.therapy:9   Min.   :0.1000
##  anxifree:3   CBT        :0   1st Qu.:0.3000
##  joyzepam:3                   Median :0.5000
##                               Mean   :0.7222
##                               3rd Qu.:1.3000
##                               Max.   :1.7000
## ------------------------------------------------------------
## clin.trial$therapy: CBT
##        drug         therapy    mood.gain
##  placebo :3   no.therapy:0   Min.   :0.300
##  anxifree:3   CBT        :9   1st Qu.:0.800
##  joyzepam:3                   Median :1.100
##                               Mean   :1.044
##                               3rd Qu.:1.300
##                               Max.   :1.800
```

Again, this output is pretty easy to interpret. It's the output of the `summary()` function, applied separately to `CBT` group and the `no.therapy` group. For the two factors (`drug` and `therapy`) it prints out a frequency table, whereas for the numeric variable (`mood.gain`) it prints out the range, interquartile range, mean and median.

What if you have multiple grouping variables? Suppose, for example, you would like to look at the average mood gain separately for all possible combinations of drug and therapy. It is actually possible to do this using the `by()` and `describeBy()`

functions, but I usually find it more convenient to use the `aggregate()` function in this situation. There are again three arguments that you need to specify. The `formula` argument is used to indicate which variable you want to analyse, and which variables are used to specify the groups. For instance, if you want to look at `mood.gain` separately for each possible combination of `drug` and `therapy`, the formula you want is `mood.gain ~ drug + therapy`. The `data` argument is used to specify the data frame containing all the data, and the `FUN` argument is used to indicate what function you want to calculate for each group (e.g., the `mean`). So, to obtain group means, use this command:

```
aggregate( formula = mood.gain ~ drug + therapy,  # mood.gain by drug/therapy combina
           data = clin.trial,                      # data is in the clin.trial data f
           FUN = mean                              # print out group means
 )
```

```
##       drug    therapy mood.gain
## 1  placebo no.therapy  0.300000
## 2 anxifree no.therapy  0.400000
## 3 joyzepam no.therapy  1.466667
## 4  placebo        CBT  0.600000
## 5 anxifree        CBT  1.033333
## 6 joyzepam        CBT  1.500000
```

or, alternatively, if you want to calculate the standard deviations for each group, you would use the following command (argument names omitted this time):

```
aggregate( mood.gain ~ drug + therapy, clin.trial, sd )
```

```
##       drug    therapy mood.gain
## 1  placebo no.therapy 0.2000000
## 2 anxifree no.therapy 0.2000000
## 3 joyzepam no.therapy 0.2081666
## 4  placebo        CBT 0.3000000
## 5 anxifree        CBT 0.2081666
## 6 joyzepam        CBT 0.2645751
```

This page titled 6.5: Descriptive Statistics Separately for each Group is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **5.5: Descriptive Statistics Separately for each Group** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

6.6: Standard Scores

Suppose my friend is putting together a new questionnaire intended to measure "grumpiness". The survey has 50 questions, which you can answer in a grumpy way or not. Across a big sample (hypothetically, let's imagine a million people or so!) the data are fairly normally distributed, with the mean grumpiness score being 17 out of 50 questions answered in a grumpy way, and the standard deviation is 5. In contrast, when I take the questionnaire, I answer 35 out of 50 questions in a grumpy way. So, how grumpy am I? One way to think about would be to say that I have grumpiness of 35/50, so you might say that I'm 70% grumpy. But that's a bit weird, when you think about it. If my friend had phrased her questions a bit differently, people might have answered them in a different way, so the overall distribution of answers could easily move up or down depending on the precise way in which the questions were asked. So, I'm only 70% grumpy *with respect to this set of survey questions*. Even if it's a very good questionnaire, this isn't very a informative statement.

A simpler way around this is to describe my grumpiness by comparing me to other people. Shockingly, out of my friend's sample of 1,000,000 people, only 159 people were as grumpy as me (that's not at all unrealistic, frankly), suggesting that I'm in the top 0.016% of people for grumpiness. This makes much more sense than trying to interpret the raw data. This idea – that we should describe my grumpiness in terms of the overall distribution of the grumpiness of humans – is the qualitative idea that standardisation attempts to get at. One way to do this is to do exactly what I just did, and describe everything in terms of percentiles. However, the problem with doing this is that "it's lonely at the top". Suppose that my friend had only collected a sample of 1000 people (still a pretty big sample for the purposes of testing a new questionnaire, I'd like to add), and this time gotten a mean of 16 out of 50 with a standard deviation of 5, let's say. The problem is that almost certainly, not a single person in that sample would be as grumpy as me.

However, all is not lost. A different approach is to convert my grumpiness score into a ***standard score***, also referred to as a z-score. The standard score is defined as the number of standard deviations above the mean that my grumpiness score lies. To phrase it in "pseudo-maths" the standard score is calculated like this:

$$\text{standard score} = \frac{\text{raw score} - \text{mean}}{\text{standard deviation}}$$

In actual maths, the equation for the z-score is

$$z_i = \frac{X_i - \bar{X}}{\hat{\sigma}}$$

So, going back to the grumpiness data, we can now transform Dan's raw grumpiness into a standardised grumpiness score.[76] If the mean is 17 and the standard deviation is 5 then my standardised grumpiness score would be[77]

$$z = \frac{35 - 17}{5} = 3.6$$

To interpret this value, recall the rough heuristic that I provided in Section 5.2.5, in which I noted that 99.7% of values are expected to lie within 3 standard deviations of the mean. So the fact that my grumpiness corresponds to a z score of 3.6 indicates that I'm very grumpy indeed. Later on, in Section 9.5, I'll introduce a function called `pnorm()` that allows us to be a bit more precise than this. Specifically, it allows us to calculate a theoretical percentile rank for my grumpiness, as follows:

```
pnorm( 3.6 )
```

```
## [1] 0.9998409
```

At this stage, this command doesn't make too much sense, but don't worry too much about it. It's not important for now. But the output is fairly straightforward: it suggests that I'm grumpier than 99.98% of people. Sounds about right.

In addition to allowing you to interpret a raw score in relation to a larger population (and thereby allowing you to make sense of variables that lie on arbitrary scales), standard scores serve a second useful function. Standard scores can be compared to one another in situations where the raw scores can't. Suppose, for instance, my friend also had another questionnaire that measured extraversion using a 24 items questionnaire. The overall mean for this measure turns out to be 13 with standard deviation 4; and I scored a 2. As you can imagine, it doesn't make a lot of sense to try to compare my raw score of 2 on the extraversion questionnaire

to my raw score of 35 on the grumpiness questionnaire. The raw scores for the two variables are "about" fundamentally different things, so this would be like comparing apples to oranges.

What about the standard scores? Well, this is a little different. If we calculate the standard scores, we get z=(35−17)/5=3.6 for grumpiness and z=(2−13)/4=−2.75 for extraversion. These two numbers *can* be compared to each other.[78] I'm much less extraverted than most people (z=−2.75) and much grumpier than most people (z=3.6): but the extent of my unusualness is much more extreme for grumpiness (since 3.6 is a bigger number than 2.75). Because each standardised score is a statement about where an observation falls *relative to its own population*, it *is* possible to compare standardised scores across completely different variables.

This page titled 6.6: Standard Scores is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **5.6: Standard Scores** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

6.7: Epilogue- Good Descriptive Statistics Are Descriptive!

> *The death of one man is a tragedy. The death of millions is a statistic.*
>
> *– Josef Stalin, Potsdam 1945*

> *950,000 – 1,200,000*
>
> *– Estimate of Soviet repression deaths, 1937-1938 (Ellman 2002)*

Stalin's infamous quote about the statistical character death of millions is worth giving some thought. The clear intent of his statement is that the death of an individual touches us personally and its force cannot be denied, but that the deaths of a multitude are incomprehensible, and as a consequence mere statistics, more easily ignored. I'd argue that Stalin was half right. A statistic is an abstraction, a description of events beyond our personal experience, and so hard to visualise. Few if any of us can imagine what the deaths of millions is "really" like, but we can imagine one death, and this gives the lone death its feeling of immediate tragedy, a feeling that is missing from Ellman's cold statistical description.

Yet it is not so simple: without numbers, without counts, without a description of what happened, we have *no chance* of understanding what really happened, no opportunity event to try to summon the missing feeling. And in truth, as I write this, sitting in comfort on a Saturday morning, half a world and a whole lifetime away from the Gulags, when I put the Ellman estimate next to the Stalin quote a dull dread settles in my stomach and a chill settles over me. The Stalinist repression is something truly beyond my experience, but with a combination of statistical data and those recorded personal histories that have come down to us, it is not entirely beyond my comprehension. Because what Ellman's numbers tell us is this: over a two year period, Stalinist repression wiped out the equivalent of every man, woman and child currently alive in the city where I live. Each one of those deaths had it's own story, was it's own tragedy, and only some of those are known to us now. Even so, with a few carefully chosen statistics, the scale of the atrocity starts to come into focus.

Thus it is no small thing to say that the first task of the statistician and the scientist is to summarise the data, to find some collection of numbers that can convey to an audience a sense of what has happened. This is the job of descriptive statistics, but it's not a job that can be told solely using the numbers. You are a data analyst, not a statistical software package. Part of your job is to take these *statistics* and turn them into a *description*. When you analyse data, it is not sufficient to list off a collection of numbers. Always remember that what you're really trying to do is communicate with a human audience. The numbers are important, but they need to be put together into a meaningful story that your audience can interpret. That means you need to think about framing. You need to think about context. And you need to think about the individual events that your statistics are summarising.

References

Ellman, Michael. 2002. "Soviet Repression Statistics: Some Comments." *Europe-Asia Studies* 54 (7). Taylor & Francis: 1151–72.

---

64. Note for non-Australians: the AFL is an Australian rules football competition. You don't need to know anything about Australian rules in order to follow this section.
65. The choice to use Σ to denote summation isn't arbitrary: it's the Greek upper case letter sigma, which is the analogue of the letter S in that alphabet. Similarly, there's an equivalent symbol used to denote the multiplication of lots of numbers: because multiplications are also called "products", we use the Π symbol for this; the Greek upper case pi, which is the analogue of the letter P.
66. Note that, just as we saw with the combine function `c()` and the remove function `rm()`, the `sum()` function has unnamed arguments. I'll talk about unnamed arguments later in Section 8.4.1, but for now let's just ignore this detail.
67. www.abc.net.au/news/stories/2010/09/24/3021480.htm
68. Or at least, the basic statistical theory – these days there is a whole subfield of statistics called *robust statistics* that tries to grapple with the messiness of real data and develop theory that can cope with it.
69. As we saw earlier, it *does* have a function called `mode()`, but it does something completely different.
70. This is called a "0-1 loss function", meaning that you either win (1) or you lose (0), with no middle ground.

71. Well, I will very briefly mention the one that I think is coolest, for a very particular definition of “cool”, that is. Variances are *additive*. Here’s what that means: suppose I have two variables X and Y, whose variances are $
72. With the possible exception of the third question.
73. Strictly, the assumption is that the data are *normally* distributed, which is an important concept that we’ll discuss more in Chapter 9, and will turn up over and over again later in the book.
74. The assumption again being that the data are normally-distributed!
75. The “−3” part is something that statisticians tack on to ensure that the normal curve has kurtosis zero. It looks a bit stupid, just sticking a “-3” at the end of the formula, but there are good mathematical reasons for doing this.
76. I haven’t discussed how to compute z-scores, explicitly, but you can probably guess. For a variable `X`, the simplest way is to use a command like `(X - mean(X)) / sd(X)`. There’s also a fancier function called `scale()` that you can use, but it relies on somewhat more complicated R concepts that I haven’t explained yet.
77. Technically, because I’m calculating means and standard deviations from a sample of data, but want to talk about my grumpiness relative to a population, what I’m actually doing is *estimating* a z score. However, since we haven’t talked about estimation yet (see Chapter 10) I think it’s best to ignore this subtlety, especially as it makes very little difference to our calculations.
78. Though some caution is usually warranted. It’s not always the case that one standard deviation on variable A corresponds to the same “kind” of thing as one standard deviation on variable B. Use common sense when trying to determine whether or not the z scores of two variables can be meaningfully compared.
79. Actually, even that table is more than I’d bother with. In practice most people pick *one* measure of central tendency, and *one* measure of variability only.
80. Just like we saw with the variance and the standard deviation, in practice we divide by N−1 rather than N.
81. This is an oversimplification, but it’ll do for our purposes.
82. If you are reading this after having already completed Chapter 11 you might be wondering about hypothesis tests for correlations. R has a function called `cor.test()` that runs a hypothesis test for a single correlation, and the `psych` package contains a version called `corr.test()` that can run tests for every correlation in a correlation matrix; hypothesis tests for correlations are discussed in more detail in Section 15.6.
83. An alternative usage of `cor()` is to correlate one set of variables with another subset of variables. If `X` and `Y` are both data frames with the same number of rows, then `cor(x = X, y = Y)` will produce a correlation matrix that correlates all variables in `X` with all variables in `Y`.
84. It’s worth noting that, even though we have missing data for each of these variables, the output doesn’t contain any `NA` values. This is because, while `describe()` also has an `na.rm` argument, the default value for this function is `na.rm = TRUE`.
85. The technical term here is “missing completely at random” (often written MCAR for short). Makes sense, I suppose, but it does sound ungrammatical to me.

This page titled 6.7: Epilogue- Good Descriptive Statistics Are Descriptive! is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **5.10: Epilogue- Good Descriptive Statistics Are Descriptive!** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

CHAPTER OVERVIEW

7: Introduction to Probability

> *[God] has afforded us only the twilight … of Probability.*
>
> *– John Locke*

Up to this point in the book, we've discussed some of the key ideas in experimental design, and we've talked a little about how you can summarise a data set. To a lot of people, this is all there is to statistics: it's about calculating averages, collecting all the numbers, drawing pictures, and putting them all in a report somewhere. Kind of like stamp collecting, but with numbers. However, statistics covers much more than that. In fact, descriptive statistics is one of the smallest parts of statistics, and one of the least powerful. The bigger and more useful part of statistics is that it provides that let you make inferences about data.

Once you start thinking about statistics in these terms – that statistics is there to help us draw inferences from data – you start seeing examples of it everywhere. For instance, here's a tiny extract from a newspaper article in the Sydney Morning Herald (30 Oct 2010):

"I have a tough job," the Premier said in response to a poll which found her government is now the most unpopular Labor administration in polling history, with a primary vote of just 23 per cent.

This kind of remark is entirely unremarkable in the papers or in everyday life, but let's have a think about what it entails. A polling company has conducted a survey, usually a pretty big one because they can afford it. I'm too lazy to track down the original survey, so let's just imagine that they called 1000 NSW voters at random, and 230 (23%) of those claimed that they intended to vote for the ALP. For the 2010 Federal election, the Australian Electoral Commission reported 4,610,795 enrolled voters in NSW; so the opinions of the remaining 4,609,795 voters (about 99.98% of voters) remain unknown to us. Even assuming that no-one lied to the polling company the only thing we can say with 100% confidence is that the true ALP primary vote is somewhere between 230/4610795 (about 0.005%) and 4610025/4610795 (about 99.83%). So, on what basis is it legitimate for the polling company, the newspaper, and the readership to conclude that the ALP primary vote is only about 23%?

The answer to the question is pretty obvious: if I call 1000 people at random, and 230 of them say they intend to vote for the ALP, then it seems very unlikely that these are the *only* 230 people out of the entire voting public who actually intend to do so. In other words, we assume that the data collected by the polling company is pretty representative of the population at large. But how representative? Would we be surprised to discover that the true ALP primary vote is actually 24%? 29%? 37%? At this point everyday intuition starts to break down a bit. No-one would be surprised by 24%, and everybody would be surprised by 37%, but it's a bit hard to say whether 29% is plausible. We need some more powerful tools than just looking at the numbers and guessing.

Inferential statistics provides the tools that we need to answer these sorts of questions, and since these kinds of questions lie at the heart of the scientific enterprise, they take up the lions share of every introductory course on statistics and research methods. However, the theory of statistical inference is built on top of ***probability theory***. And it is to probability theory that we must now turn. This discussion of probability theory is basically background: there's not a lot of statistics per se in this chapter, and you don't need to understand this material in as much depth as the other chapters in this part of the book. Nevertheless, because probability theory does underpin so much of statistics, it's worth covering some of the basics.

7.1: How are Probability and Statistics Different?
7.2: What Does Probability Mean?
7.3: Basic Probability Theory
7.4: The Binomial Distribution
7.5: The Normal Distribution
7.6: Other Useful Distributions
7.7: Summary
7.8: Statistical Literacy

7.E: Probability (Exercises)

This page titled 7: Introduction to Probability is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

7.1: How are Probability and Statistics Different?

Before we start talking about probability theory, it's helpful to spend a moment thinking about the relationship between probability and statistics. The two disciplines are closely related but they're not identical. Probability theory is "the doctrine of chances". It's a branch of mathematics that tells you how often different kinds of events will happen. For example, all of these questions are things you can answer using probability theory:

- What are the chances of a fair coin coming up heads 10 times in a row?
- If I roll two six sided dice, how likely is it that I'll roll two sixes?
- How likely is it that five cards drawn from a perfectly shuffled deck will all be hearts?
- What are the chances that I'll win the lottery?

Notice that all of these questions have something in common. In each case the "truth of the world" is known, and my question relates to the "what kind of events" will happen. In the first question I *know* that the coin is fair, so there's a 50% chance that any individual coin flip will come up heads. In the second question, I *know* that the chance of rolling a 6 on a single die is 1 in 6. In the third question I *know* that the deck is shuffled properly. And in the fourth question, I *know* that the lottery follows specific rules. You get the idea. The critical point is that probabilistic questions start with a known ***model*** of the world, and we use that model to do some calculations. The underlying model can be quite simple. For instance, in the coin flipping example, we can write down the model like this:

$$P(\text{heads})=0.5$$

which you can read as "the probability of heads is 0.5". As we'll see later, in the same way that percentages are numbers that range from 0% to 100%, probabilities are just numbers that range from 0 to 1. When using this probability model to answer the first question, I don't actually know exactly what's going to happen. Maybe I'll get 10 heads, like the question says. But maybe I'll get three heads. That's the key thing: in probability theory, the *model* is known, but the *data* are not.

So that's probability. What about statistics? Statistical questions work the other way around. In statistics, we *do not* know the truth about the world. All we have is the data, and it is from the data that we want to *learn* the truth about the world. Statistical questions tend to look more like these:

- If my friend flips a coin 10 times and gets 10 heads, are they playing a trick on me?
- If five cards off the top of the deck are all hearts, how likely is it that the deck was shuffled? - If the lottery commissioner's spouse wins the lottery, how likely is it that the lottery was rigged?

This time around, the only thing we have are data. What I *know* is that I saw my friend flip the coin 10 times and it came up heads every time. And what I want to ***infer*** is whether or not I should conclude that what I just saw was actually a fair coin being flipped 10 times in a row, or whether I should suspect that my friend is playing a trick on me. The data I have look like this:

```
H H H H H H H H H H
```

and what I'm trying to do is work out which "model of the world" I should put my trust in. If the coin is fair, then the model I should adopt is one that says that the probability of heads is 0.5; that is, P(heads)=0.5. If the coin is not fair, then I should conclude that the probability of heads is *not* 0.5, which we would write as P(heads)≠0.5. In other words, the statistical inference problem is to figure out which of these probability models is right. Clearly, the statistical question isn't the same as the probability question, but they're deeply connected to one another. Because of this, a good introduction to statistical theory will start with a discussion of what probability is and how it works.

This page titled 7.1: How are Probability and Statistics Different? is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **9.1: How are Probability and Statistics Different?** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

7.2: What Does Probability Mean?

Let's start with the first of these questions. What is "probability"? It might seem surprising to you, but while statisticians and mathematicians (mostly) agree on what the *rules* of probability are, there's much less of a consensus on what the word really *means*. It seems weird because we're all very comfortable using words like "chance", "likely", "possible" and "probable", and it doesn't seem like it should be a very difficult question to answer. If you had to explain "probability" to a five year old, you could do a pretty good job. But if you've ever had that experience in real life, you might walk away from the conversation feeling like you didn't quite get it right, and that (like many everyday concepts) it turns out that you don't *really* know what it's all about.

So I'll have a go at it. Let's suppose I want to bet on a soccer game between two teams of robots, *Arduino Arsenal* and *C Milan*. After thinking about it, I decide that there is an 80% probability that *Arduino Arsenal* winning. What do I mean by that? Here are three possibilities…

- They're robot teams, so I can make them play over and over again, and if I did that, *Arduino Arsenal* would win 8 out of every 10 games on average.
- For any given game, I would only agree that betting on this game is only "fair" if a $1 bet on *C Milan* gives a $5 payoff (i.e. I get my $1 back plus a $4 reward for being correct), as would a $4 bet on *Arduino Arsenal* (i.e., my $4 bet plus a $1 reward).
- My subjective "belief" or "confidence" in an *Arduino Arsenal* victory is four times as strong as my belief in a *C Milan* victory.

Each of these seems sensible. However they're not identical, and not every statistician would endorse all of them. The reason is that there are different statistical ideologies (yes, really!) and depending on which one you subscribe to, you might say that some of those statements are meaningless or irrelevant. In this section, I give a brief introduction the two main approaches that exist in the literature. These are by no means the only approaches, but they're the two big ones.

7.2.1 frequentist view

The first of the two major approaches to probability, and the more dominant one in statistics, is referred to as the ***frequentist view***, and it defines probability as a ***long-run frequency***. Suppose we were to try flipping a fair coin, over and over again. By definition, this is a coin that has P(H)=0.5. What might we observe? One possibility is that the first 20 flips might look like this:

```
T,H,H,H,H,T,T,H,H,H,H,T,H,H,T,T,T,T,T,H
```

In this case 11 of these 20 coin flips (55%) came up heads. Now suppose that I'd been keeping a running tally of the number of heads (which I'll call N_H) that I've seen, across the first N flips, and calculate the proportion of heads N_H/N every time. Here's what I'd get (I did literally flip coins to produce this!):

number.of.flips	number.of.heads	proportion
1	0	0.00
2	1	0.50
3	2	0.67
4	3	0.75
5	4	0.80
6	4	0.67
7	4	0.57
8	5	0.63
9	6	0.67
10	7	0.70
11	8	0.73
12	8	0.67
13	9	0.69
14	10	0.71
15	10	0.67
16	10	0.63
17	10	0.59
18	10	0.56
19	10	0.53
20	11	0.55

Notice that at the start of the sequence, the *proportion* of heads fluctuates wildly, starting at .00 and rising as high as .80. Later on, one gets the impression that it dampens out a bit, with more and more of the values actually being pretty close to the "right" answer of .50. This is the frequentist definition of probability in a nutshell: flip a fair coin over and over again, and as N grows large (approaches infinity, denoted $N \rightarrow \infty$), the proportion of heads will converge to 50%. There are some subtle technicalities that the mathematicians care about, but qualitatively speaking, that's how the frequentists define probability. Unfortunately, I don't have an infinite number of coins, or the infinite patience required to flip a coin an infinite number of times. However, I do have a computer, and computers excel at mindless repetitive tasks. So I asked my computer to simulate flipping a coin 1000 times, and then drew a picture of what happens to the proportion N_H/N as N increases. Actually, I did it four times, just to make sure it wasn't a fluke. The results are shown in Figure 9.1. As you can see, the *proportion of observed heads* eventually stops fluctuating, and settles down; when it does, the number at which it finally settles is the true probability of heads.

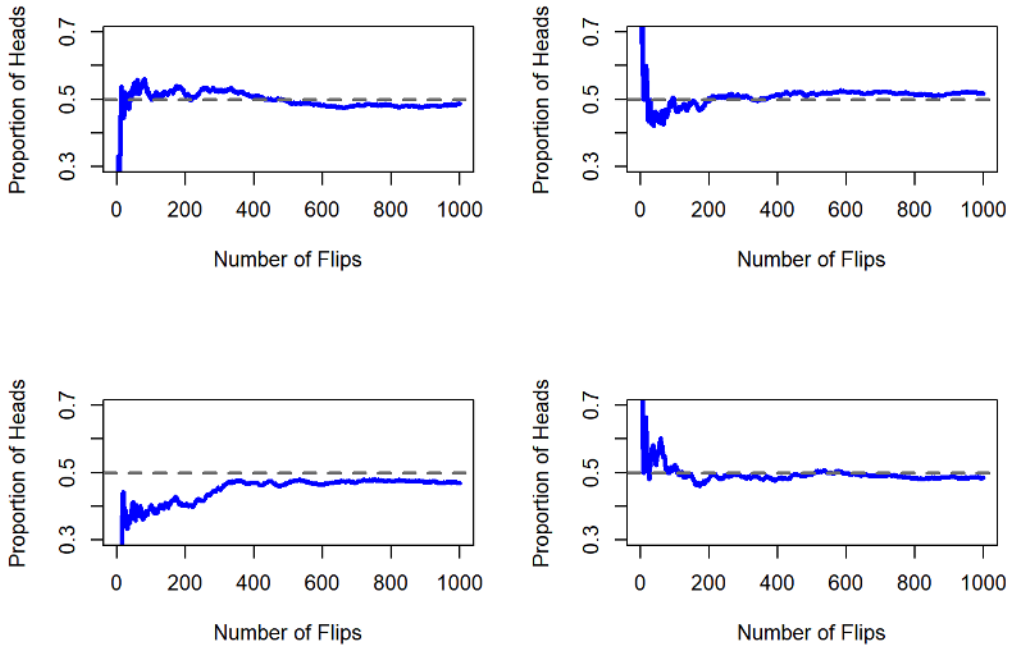


Figure 9.1: An illustration of how frequentist probability works. If you flip a fair coin over and over again, the proportion of heads that you've seen eventually settles down, and converges to the true probability of 0.5. Each panel shows four different simulated experiments: in each case, we pretend we flipped a coin 1000 times, and kept track of the proportion of flips that were heads as we went along. Although none of these sequences actually ended up with an exact value of .5, if we'd extended the experiment for an infinite number of coin flips they would have.

The frequentist definition of probability has some desirable characteristics. Firstly, it is objective: the probability of an event is *necessarily* grounded in the world. The only way that probability statements can make sense is if they refer to (a sequence of) events that occur in the physical universe.[142] Secondly, it is unambiguous: any two people watching the same sequence of events unfold, trying to calculate the probability of an event, must inevitably come up with the same answer. However, it also has undesirable characteristics. Firstly, infinite sequences don't exist in the physical world. Suppose you picked up a coin from your pocket and started to flip it. Every time it lands, it impacts on the ground. Each impact wears the coin down a bit; eventually, the coin will be destroyed. So, one might ask whether it really makes sense to pretend that an "infinite" sequence of coin flips is even a meaningful concept, or an objective one. We can't say that an "infinite sequence" of events is a real thing in the physical universe, because the physical universe doesn't allow infinite anything. More seriously, the frequentist definition has a narrow scope. There are lots of things out there that human beings are happy to assign probability to in everyday language, but cannot (even in theory) be mapped onto a hypothetical sequence of events. For instance, if a meteorologist comes on TV and says, "the probability of rain in Adelaide on 2 November 2048 is 60%" we humans are happy to accept this. But it's not clear how to define this in frequentist terms. There's only one city of Adelaide, and only 2 November 2048. There's no infinite sequence of events here, just a once-off thing. Frequentist probability genuinely *forbids* us from making probability statements about a single event. From the frequentist perspective, it will either rain tomorrow or it will not; there is no "probability" that attaches to a single non-repeatable event. Now, it should be said that there are some very clever tricks that frequentists can use to get around this. One possibility is that what the meteorologist means is something like this: "There is a category of days for which I predict a 60% chance of rain; if we look only across those days for which I make this prediction, then on 60% of those days it will actually rain". It's very weird and counterintuitive to think of it this way, but you do see frequentists do this sometimes. And it *will* come up later in this book (see Section 10.5).

7.2.2 Bayesian view

The ***Bayesian view*** of probability is often called the subjectivist view, and it is a minority view among statisticians, but one that has been steadily gaining traction for the last several decades. There are many flavours of Bayesianism, making hard to say exactly what "the" Bayesian view is. The most common way of thinking about subjective probability is to define the probability of an event as the ***degree of belief*** that an intelligent and rational agent assigns to that truth of that event. From that perspective, probabilities don't exist in the world, but rather in the thoughts and assumptions of people and other intelligent beings. However, in

order for this approach to work, we need some way of operationalising "degree of belief". One way that you can do this is to formalise it in terms of "rational gambling", though there are many other ways. Suppose that I believe that there's a 60% probability of rain tomorrow. If someone offers me a bet: if it rains tomorrow, then I win $5, but if it doesn't rain then I lose $5. Clearly, from my perspective, this is a pretty good bet. On the other hand, if I think that the probability of rain is only 40%, then it's a bad bet to take. Thus, we can operationalise the notion of a "subjective probability" in terms of what bets I'm willing to accept.

What are the advantages and disadvantages to the Bayesian approach? The main advantage is that it allows you to assign probabilities to any event you want to. You don't need to be limited to those events that are repeatable. The main disadvantage (to many people) is that we can't be purely objective – specifying a probability requires us to specify an entity that has the relevant degree of belief. This entity might be a human, an alien, a robot, or even a statistician, but there has to be an intelligent agent out there that believes in things. To many people this is uncomfortable: it seems to make probability arbitrary. While the Bayesian approach does require that the agent in question be rational (i.e., obey the rules of probability), it does allow everyone to have their own beliefs; I can believe the coin is fair and you don't have to, even though we're both rational. The frequentist view doesn't allow any two observers to attribute different probabilities to the same event: when that happens, then at least one of them must be wrong. The Bayesian view does not prevent this from occurring. Two observers with different background knowledge can legitimately hold different beliefs about the same event. In short, where the frequentist view is sometimes considered to be too narrow (forbids lots of things that that we want to assign probabilities to), the Bayesian view is sometimes thought to be too broad (allows too many differences between observers).

7.2.3 What's the difference? And who is right?

Now that you've seen each of these two views independently, it's useful to make sure you can compare the two. Go back to the hypothetical robot soccer game at the start of the section. What do you think a frequentist and a Bayesian would say about these three statements? Which statement would a frequentist say is the correct definition of probability? Which one would a Bayesian do? Would some of these statements be meaningless to a frequentist or a Bayesian? If you've understood the two perspectives, you should have some sense of how to answer those questions.

Okay, assuming you understand the different, you might be wondering which of them is *right*? Honestly, I don't know that there is a right answer. As far as I can tell there's nothing mathematically incorrect about the way frequentists think about sequences of events, and there's nothing mathematically incorrect about the way that Bayesians define the beliefs of a rational agent. In fact, when you dig down into the details, Bayesians and frequentists actually agree about a lot of things. Many frequentist methods lead to decisions that Bayesians agree a rational agent would make. Many Bayesian methods have very good frequentist properties.

For the most part, I'm a pragmatist so I'll use any statistical method that I trust. As it turns out, that makes me prefer Bayesian methods, for reasons I'll explain towards the end of the book, but I'm not fundamentally opposed to frequentist methods. Not everyone is quite so relaxed. For instance, consider Sir Ronald Fisher, one of the towering figures of 20th century statistics and a vehement opponent to all things Bayesian, whose paper on the mathematical foundations of statistics referred to Bayesian probability as "an impenetrable jungle [that] arrests progress towards precision of statistical concepts" Fisher (1922b). Or the psychologist Paul Meehl, who suggests that relying on frequentist methods could turn you into "a potent but sterile intellectual rake who leaves in his merry path a long train of ravished maidens but no viable scientific offspring" Meehl (1967). The history of statistics, as you might gather, is not devoid of entertainment.

In any case, while I personally prefer the Bayesian view, the majority of statistical analyses are based on the frequentist approach. My reasoning is pragmatic: the goal of this book is to cover roughly the same territory as a typical undergraduate stats class in psychology, and if you want to understand the statistical tools used by most psychologists, you'll need a good grasp of frequentist methods. I promise you that this isn't wasted effort. Even if you end up wanting to switch to the Bayesian perspective, you really should read through at least one book on the "orthodox" frequentist view. And since R is the most widely used statistical language for Bayesians, you might as well read a book that uses R. Besides, I won't completely ignore the Bayesian perspective. Every now and then I'll add some commentary from a Bayesian point of view, and I'll revisit the topic in more depth in Chapter 17.

This page titled 7.2: What Does Probability Mean? is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **9.2: What Does Probability Mean?** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

https://stats.libretexts.org/@go/page/36106

7.3: Basic Probability Theory

Ideological arguments between Bayesians and frequentists notwithstanding, it turns out that people mostly agree on the rules that probabilities should obey. There are lots of different ways of arriving at these rules. The most commonly used approach is based on the work of Andrey Kolmogorov, one of the great Soviet mathematicians of the 20th century. I won't go into a lot of detail, but I'll try to give you a bit of a sense of how it works. And in order to do so, I'm going to have to talk about my pants.

7.3.1 Introducing probability distributions

One of the disturbing truths about my life is that I only own 5 pairs of pants: three pairs of jeans, the bottom half of a suit, and a pair of tracksuit pants. Even sadder, I've given them names: I call them X_1, X_2, X_3, X_4 and X_5. I really do: that's why they call me Mister Imaginative. Now, on any given day, I pick out exactly one of pair of pants to wear. Not even I'm so stupid as to try to wear two pairs of pants, and thanks to years of training I never go outside without wearing pants anymore. If I were to describe this situation using the language of probability theory, I would refer to each pair of pants (i.e., each X) as an ***elementary event***. The key characteristic of elementary events is that every time we make an observation (e.g., every time I put on a pair of pants), then the outcome will be one and only one of these events. Like I said, these days I always wear exactly one pair of pants, so my pants satisfy this constraint. Similarly, the set of all possible events is called a ***sample space***. Granted, some people would call it a "wardrobe", but that's because they're refusing to think about my pants in probabilistic terms. Sad.

Okay, now that we have a sample space (a wardrobe), which is built from lots of possible elementary events (pants), what we want to do is assign a ***probability*** of one of these elementary events. For an event X, the probability of that event P(X) is a number that lies between 0 and 1. The bigger the value of P(X), the more likely the event is to occur. So, for example, if P(X)=0, it means the event X is impossible (i.e., I never wear those pants). On the other hand, if P(X)=1 it means that event X is certain to occur (i.e., I always wear those pants). For probability values in the middle, it means that I sometimes wear those pants. For instance, if P(X)=0.5 it means that I wear those pants half of the time.

At this point, we're almost done. The last thing we need to recognise is that "something always happens". Every time I put on pants, I really do end up wearing pants (crazy, right?). What this somewhat trite statement means, in probabilistic terms, is that the probabilities of the elementary events need to add up to 1. This is known as the ***law of total probability***, not that any of us really care. More importantly, if these requirements are satisfied, then what we have is a ***probability distribution***. For example, this is an example of a probability distribution

Which.pants	Blue.jeans	Grey.jeans	Black.jeans	Black.suit	Blue.tracksuit
Label	X_1	X_2	X_3	X_4	X_5
Probability	$P(X_1)=.5$	$P(X_2)=.3$	$P(X_3)=.1$	$P(X_4)=0$	$P(X_5)=.1$

Each of the events has a probability that lies between 0 and 1, and if we add up the probability of all events, they sum to 1. Awesome. We can even draw a nice bar graph (see Section 6.7) to visualise this distribution, as shown in Figure **??**. And at this point, we've all achieved something. You've learned what a probability distribution is, and I've finally managed to find a way to create a graph that focuses entirely on my pants. Everyone wins!

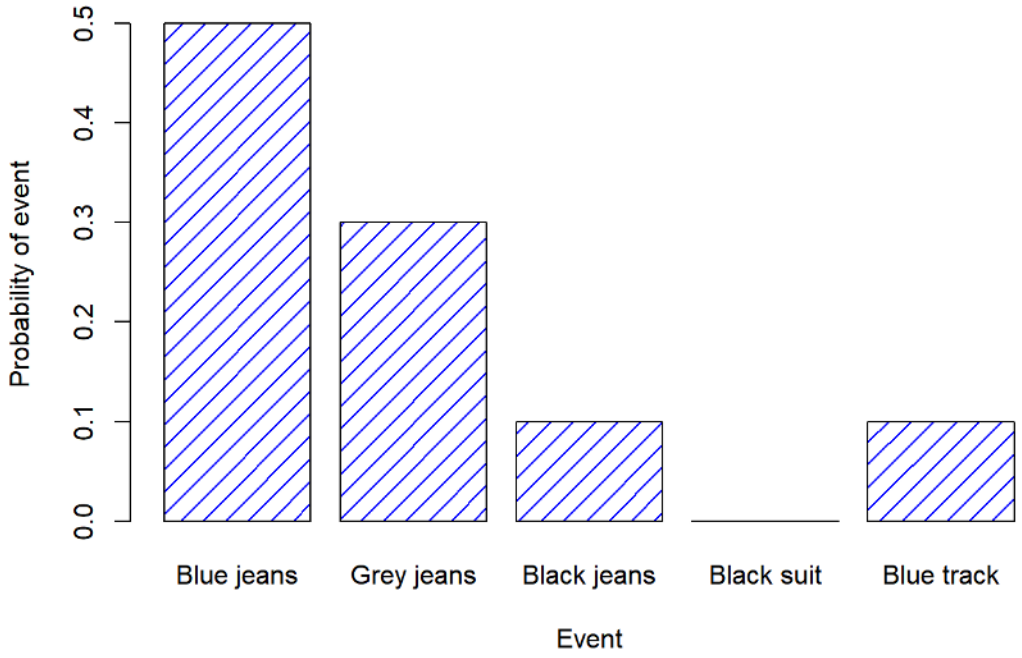


Figure 9.2: A visual depiction of the "pants" probability distribution. There are five "elementary events", corresponding to the five pairs of pants that I own. Each event has some probability of occurring: this probability is a number between 0 to 1. The sum of these probabilities is 1.

The only other thing that I need to point out is that probability theory allows you to talk about ***non elementary events*** as well as elementary ones. The easiest way to illustrate the concept is with an example. In the pants example, it's perfectly legitimate to refer to the probability that I wear jeans. In this scenario, the "Dan wears jeans" event said to have happened as long as the elementary event that actually did occur is one of the appropriate ones; in this case "blue jeans", "black jeans" or "grey jeans". In mathematical terms, we defined the "jeans" event E to correspond to the set of elementary events (X_1,X_2,X_3). If any of these elementary events occurs, then E is also said to have occurred. Having decided to write down the definition of the E this way, it's pretty straightforward to state what the probability P(E) is: we just add everything up. In this particular case

$$P(E)=P(X_1)+P(X_2)+P(X_3)$$

and, since the probabilities of blue, grey and black jeans respectively are .5, .3 and .1, the probability that I wear jeans is equal to .9.

At this point you might be thinking that this is all terribly obvious and simple and you'd be right. All we've really done is wrap some basic mathematics around a few common sense intuitions. However, from these simple beginnings it's possible to construct some extremely powerful mathematical tools. I'm definitely not going to go into the details in this book, but what I will do is list – in Table 9.1 – some of the other rules that probabilities satisfy. These rules can be derived from the simple assumptions that I've outlined above, but since we don't actually use these rules for anything in this book, I won't do so here.

Table 9.1: Some basic rules that probabilities must satisfy. You don't really need to know these rules in order to understand the analyses that we'll talk about later in the book, but they are important if you want to understand probability theory a bit more deeply.

English	Notation	NANA	Formula
Not A	$P(\neg A)$	=	$1-P(A)$
A or B	$P(A \cup B)$	=	$P(A)+P(B)-P(A \cap B)$
A and B	$P(A \cap B)$	=	$P(A\|B)P(B)$

This page titled 7.3: Basic Probability Theory is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **9.3: Basic Probability Theory** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

7.4: The Binomial Distribution

As you might imagine, probability distributions vary enormously, and there's an enormous range of distributions out there. However, they aren't all equally important. In fact, the vast majority of the content in this book relies on one of five distributions: the binomial distribution, the normal distribution, the t distribution, the χ^2 ("chi-square") distribution and the F distribution. Given this, what I'll do over the next few sections is provide a brief introduction to all five of these, paying special attention to the binomial and the normal. I'll start with the binomial distribution, since it's the simplest of the five.

7.4.1 Introducing the binomial

The theory of probability originated in the attempt to describe how games of chance work, so it seems fitting that our discussion of the ***binomial distribution*** should involve a discussion of rolling dice and flipping coins. Let's imagine a simple "experiment": in my hot little hand I'm holding 20 identical six-sided dice. On one face of each die there's a picture of a skull; the other five faces are all blank. If I proceed to roll all 20 dice, what's the probability that I'll get exactly 4 skulls? Assuming that the dice are fair, we know that the chance of any one die coming up skulls is 1 in 6; to say this another way, the skull probability for a single die is approximately .167. This is enough information to answer our question, so let's have a look at how it's done.

As usual, we'll want to introduce some names and some notation. We'll let N denote the number of dice rolls in our experiment; which is often referred to as the ***size parameter*** of our binomial distribution. We'll also use θ to refer to the the probability that a single die comes up skulls, a quantity that is usually called the ***success probability*** of the binomial.[143] Finally, we'll use X to refer to the results of our experiment, namely the number of skulls I get when I roll the dice. Since the actual value of X is due to chance, we refer to it as a ***random variable***. In any case, now that we have all this terminology and notation, we can use it to state the problem a little more precisely. The quantity that we want to calculate is the probability that X=4 given that we know that θ=.167 and N=20. The general "form" of the thing I'm interested in calculating could be written as

$$P(X \mid \theta, N)$$

and we're interested in the special case where X=4, θ=.167 and N=20. There's only one more piece of notation I want to refer to before moving on to discuss the solution to the problem. If I want to say that X is generated randomly from a binomial distribution with parameters θ and N, the notation I would use is as follows:

$$X \sim \text{Binomial}(\theta, N)$$

Yeah, yeah. I know what you're thinking: notation, notation, notation. Really, who cares? Very few readers of this book are here for the notation, so I should probably move on and talk about how to use the binomial distribution. I've included the formula for the binomial distribution in Table 9.2, since some readers may want to play with it themselves, but since most people probably don't care that much and because we don't need the formula in this book, I won't talk about it in any detail. Instead, I just want to show you what the binomial distribution looks like. To that end, Figure 9.3 plots the binomial probabilities for all possible values of X for our dice rolling experiment, from X=0 (no skulls) all the way up to X=20 (all skulls). Note that this is basically a bar chart, and is no different to the "pants probability" plot I drew in Figure 9.2. On the horizontal axis we have all the possible events, and on the vertical axis we can read off the probability of each of those events. So, the probability of rolling 4 skulls out of 20 times is about 0.20 (the actual answer is 0.2022036, as we'll see in a moment). In other words, you'd expect that to happen about 20% of the times you repeated this experiment.

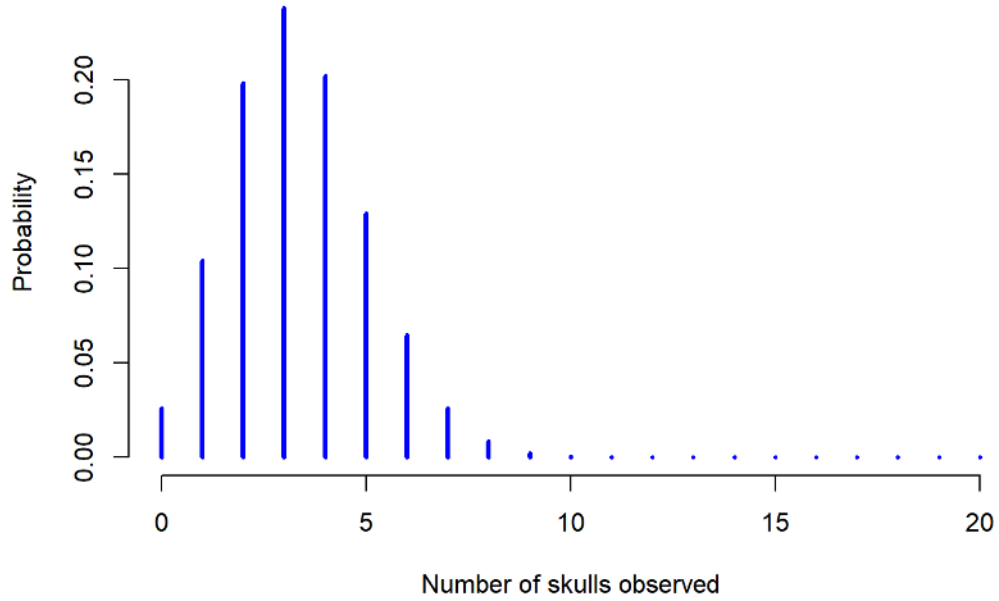


Figure 9.3: The binomial distribution with size parameter of N=20 and an underlying success probability of theta=1/6. Each vertical bar depicts the probability of one specific outcome (i.e., one possible value of X). Because this is a probability distribution, each of the probabilities must be a number between 0 and 1, and the heights of the bars must sum to 1 as well.

7.4.2 Working with the binomial distribution in R

Although some people find it handy to know the formulas in Table 9.2, most people just want to know how to use the distributions without worrying too much about the maths. To that end, R has a function called `dbinom()` that calculates binomial probabilities for us. The main arguments to the function are

- `x` . This is a number, or vector of numbers, specifying the outcomes whose probability you're trying to calculate.
- `size` . This is a number telling R the size of the experiment.
- `prob` . This is the success probability for any one trial in the experiment.

So, in order to calculate the probability of getting `x = 4` skulls, from an experiment of `size = 20` trials, in which the probability of getting a skull on any one trial is `prob = 1/6` … well, the command I would use is simply this:

```
dbinom( x = 4, size = 20, prob = 1/6 )
```

```
## [1] 0.2022036
```

To give you a feel for how the binomial distribution changes when we alter the values of θ and N, let's suppose that instead of rolling dice, I'm actually flipping coins. This time around, my experiment involves flipping a fair coin repeatedly, and the outcome that I'm interested in is the number of heads that I observe. In this scenario, the success probability is now θ=1/2. Suppose I were to flip the coin N=20 times. In this example, I've changed the success probability, but kept the size of the experiment the same. What does this do to our binomial distribution? Well, as Figure 9.4 shows, the main effect of this is to shift the whole distribution, as you'd expect. Okay, what if we flipped a coin N=100 times? Well, in that case, we get Figure 9.5. The distribution stays roughly in the middle, but there's a bit more variability in the possible outcomes.

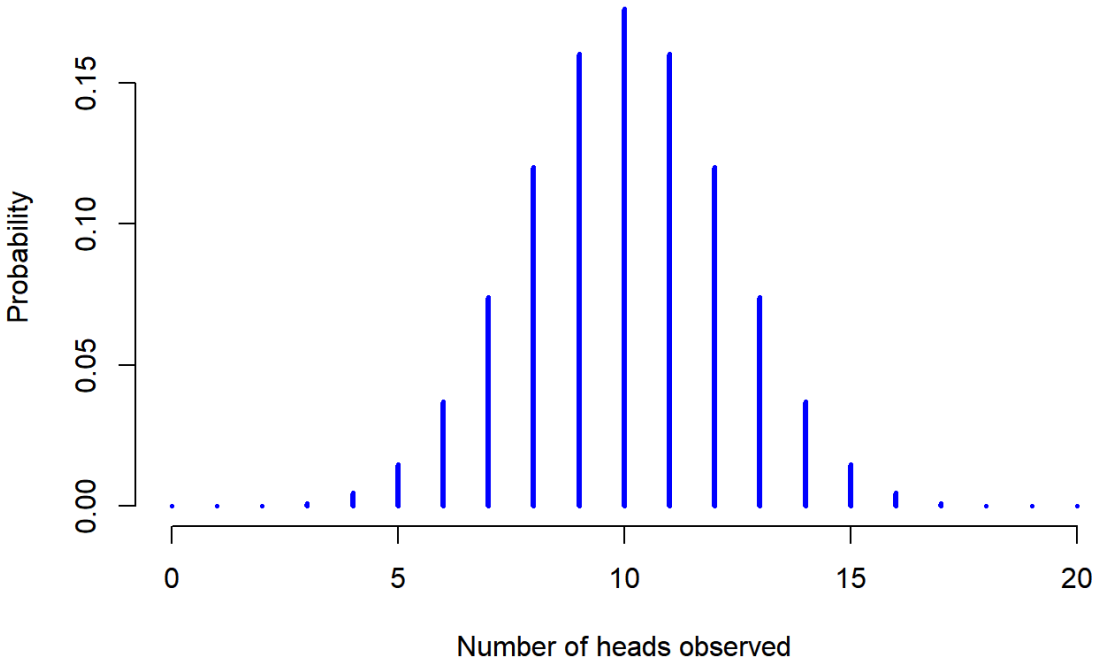


Figure 9.4: Two binomial distributions, involving a scenario in which I'm flipping a fair coin, so the underlying success probability is theta=1/2. Here we assume I'm flipping the coin N=20 times.

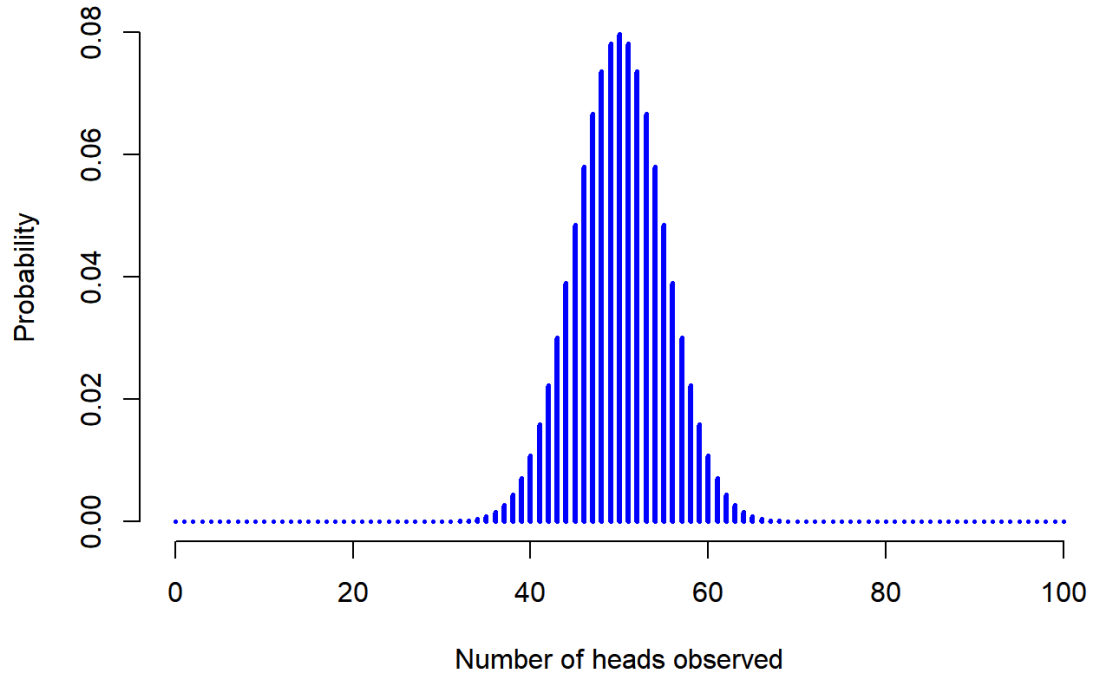


Figure 9.5: Two binomial distributions, involving a scenario in which I'm flipping a fair coin, so the underlying success probability is theta=1/2. Here we assume that the coin is flipped N=100 times.

```
knitr::kable(data.frame(stringsAsFactors=FALSE, Binomial = c("$P(X | \\theta, N) = \\
\\theta^X (1-\\theta)^{N-X}$"),
Normal = c("$p(X | \\mu, \\sigma) = \\displaystyle\\dfrac{1}{\\sqrt{2\\pi}\\sigma} \\
{2\\sigma^2} \\right)$ ")), caption = "Formulas for the binomial and normal distribut
formulas for anything in this book, but they're pretty important for more advanced wo
to put them here in a table, where they can't get in the way of the text. In the equa
factorial function (i.e., multiply all whole numbers from 1 to $X$), and for the norm
the exponential function, which we discussed in the Chapter on Data Handling. If thes
sense to you, don't worry too much about them.")
```

Table 9.2: Formulas for the binomial and normal distributions. We don't really use these formulas for anything in this book, but they're pretty important for more advanced work, so I thought it might be best to put them here in a table, where they can't get in the way of the text. In the equation for the binomial, X! is the factorial function (i.e., multiply all whole numbers from 1 to X), and for the normal distribution "exp" refers to the exponential function, which we discussed in the Chapter on Data Handling. If these equations don't make a lot of sense to you, don't worry too much about them.

Binomial	Normal
$P(X\|\theta, N) = \frac{N!}{X!(N-X)!}\theta^X(1-\theta)^{N-X}$	$p(X\|\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(X-\mu)^2}{2\sigma^2}\right)$

Table 9.3: The naming system for R probability distribution functions. Every probability distribution implemented in R is actually associated with four separate functions, and there is a pretty standardised way for naming these functions.

What.it.does	Prefix	Normal.distribution	Binomial.distribution
probability (density) of	d	dnorm()	dbinom()
cumulative probability of	p	dnorm()	pbinom()
generate random number from	r	rnorm()	rbinom()
q qnorm() qbinom()	q	qnorm()	qbinom(

At this point, I should probably explain the name of the `dbinom()` function. Obviously, the "binom" part comes from the fact that we're working with the binomial distribution, but the "d" prefix is probably a bit of a mystery. In this section I'll give a partial explanation: specifically, I'll explain why there is a prefix. As for why it's a "d" specifically, you'll have to wait until the next section. What's going on here is that R actually provides *four* functions in relation to the binomial distribution. These four functions are `dbinom()`, `pbinom()`, `rbinom()` and `qbinom()`, and each one calculates a different quantity of interest. Not only that, R does the same thing for *every* probability distribution that it implements. No matter what distribution you're talking about, there's a `d` function, a `p` function, a `q` function and a `r` function. This is illustrated in Table 9.3, using the binomial distribution and the normal distribution as examples.

Let's have a look at what all four functions do. Firstly, all four versions of the function require you to specify the `size` and `prob` arguments: no matter what you're trying to get R to calculate, it needs to know what the parameters are. However, they differ in terms of what the other argument is, and what the output is. So let's look at them one at a time.

- The `d` form we've already seen: you specify a particular outcome `x`, and the output is the probability of obtaining exactly that outcome. (the "d" is short for ***density***, but ignore that for now).
- The `p` form calculates the ***cumulative probability***. You specify a particular quantile `q`, and it tells you the probability of obtaining an outcome *smaller than or equal to* `q`.
- The `q` form calculates the ***quantiles*** of the distribution. You specify a probability value `p`, and gives you the corresponding percentile. That is, the value of the variable for which there's a probability `p` of obtaining an outcome lower than that value.

- The `r` form is a ***random number generator***: specifically, it generates `n` random outcomes from the distribution.

This is a little abstract, so let's look at some concrete examples. Again, we've already covered `dbinom()` so let's focus on the other three versions. We'll start with `pbinom()`, and we'll go back to the skull-dice example. Again, I'm rolling 20 dice, and each die has a 1 in 6 chance of coming up skulls. Suppose, however, that I want to know the probability of rolling 4 *or fewer* skulls. If I wanted to, I could use the `dbinom()` function to calculate the exact probability of rolling 0 skulls, 1 skull, 2 skulls, 3 skulls and 4 skulls and then add these up, but there's a faster way. Instead, I can calculate this using the `pbinom()` function. Here's the command:

```
pbinom( q= 4, size = 20, prob = 1/6)
```

```
## [1] 0.7687492
```

In other words, there is a 76.9% chance that I will roll 4 or fewer skulls. Or, to put it another way, R is telling us that a value of 4 is actually the 76.9th percentile of this binomial distribution.

Next, let's consider the `qbinom()` function. Let's say I want to calculate the 75th percentile of the binomial distribution. If we're sticking with our skulls example, I would use the following command to do this:

```
qbinom( p = 0.75, size = 20, prob = 1/6)
```

```
## [1] 4
```

Hm. There's something odd going on here. Let's think this through. What the `qbinom()` function appears to be telling us is that the 75th percentile of the binomial distribution is 4, even though we saw from the `pbinom()` function that 4 is *actually* the 76.9th percentile. And it's definitely the `pbinom()` function that is correct. I promise. The weirdness here comes from the fact that our binomial distribution doesn't really *have* a 75th percentile. Not really. Why not? Well, there's a 56.7% chance of rolling 3 or fewer skulls (you can type `pbinom(3, 20, 1/6)` to confirm this if you want), and a 76.9% chance of rolling 4 or fewer skulls. So there's a sense in which the 75th percentile should lie "in between" 3 and 4 skulls. But that makes no sense at all! You can't roll 20 dice and get 3.9 of them come up skulls. This issue can be handled in different ways: you could report an in between value (or *interpolated* value, to use the technical name) like 3.9, you could round down (to 3) or you could round up (to 4). The `qbinom()` function rounds upwards: if you ask for a percentile that doesn't actually exist (like the 75th in this example), R finds the smallest value for which the the percentile rank is *at least* what you asked for. In this case, since the "true" 75th percentile (whatever that would mean) lies somewhere between 3 and 4 skulls, R rounds up and gives you an answer of 4. This subtlety is tedious, I admit, but thankfully it's only an issue for discrete distributions like the binomial (see Section 2.2.5 for a discussion of continuous versus discrete). The other distributions that I'll talk about (normal, t, χ^2 and F) are all continuous, and so R can always return an exact quantile whenever you ask for it.

Finally, we have the random number generator. To use the `rbinom()` function, you specify how many times R should "simulate" the experiment using the `n` argument, and it will generate random outcomes from the binomial distribution. So, for instance, suppose I were to repeat my die rolling experiment 100 times. I could get R to simulate the results of these experiments by using the following command:

```
rbinom( n = 100, size = 20, prob = 1/6 )
```

```
##   [1] 3 3 3 2 3 3 3 2 3 3 6 2 5 3 1 1 4 7 5 3 3 6 3 4 3 4 5 3 3 3 7 4 5 1 2
##  [36] 1 2 4 2 5 5 4 4 3 1 3 0 2 3 2 2 2 2 1 3 4 5 0 3 2 5 1 2 3 1 5 2 4 3 2
##  [71] 1 2 1 5 2 3 3 2 3 3 4 2 1 2 6 2 3 2 3 3 6 2 1 1 3 3 1 5 4 3
```

As you can see, these numbers are pretty much what you'd expect given the distribution shown in Figure 9.3. Most of the time I roll somewhere between 1 to 5 skulls. There are a lot of subtleties associated with random number generation using a computer,[144] but for the purposes of this book we don't need to worry too much about them.

This page titled 7.4: The Binomial Distribution is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **9.4: The Binomial Distribution** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

7.5: The Normal Distribution

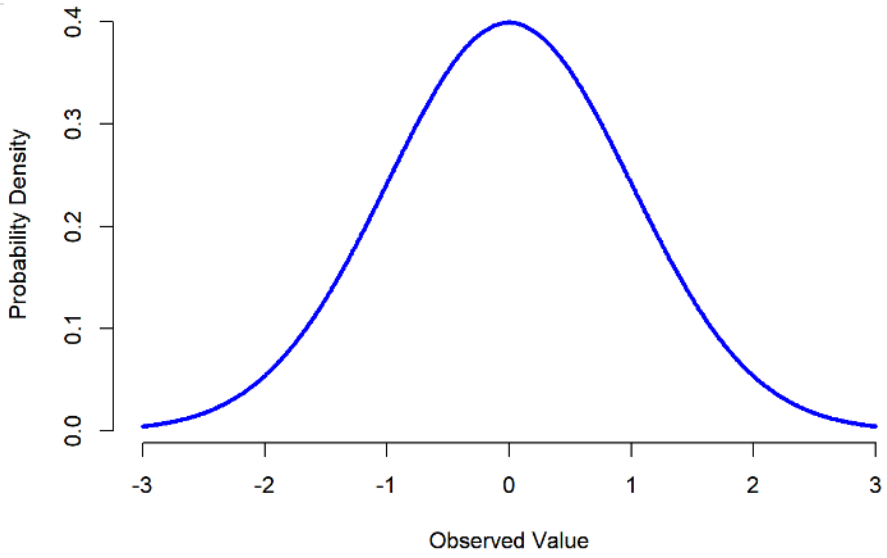


Figure 9.6: {The normal distribution with mean mu=0 and standard deviation sigma=1. The x-axis corresponds to the value of some variable, and the y-axis tells us something about how likely we are to observe that value. However, notice that the y-axis is labelled "Probability Density" and not "Probability". There is a subtle and somewhat frustrating characteristic of continuous distributions that makes the y axis behave a bit oddly: the height of the curve here isn't actually the probability of observing a particular x value. On the other hand, it *is* true that the heights of the curve tells you which x values are more likely (the higher ones!).

While the binomial distribution is conceptually the simplest distribution to understand, it's not the most important one. That particular honour goes to the ***normal distribution***, which is also referred to as "the bell curve" or a "Gaussian distribution". A normal distribution is described using two parameters, the mean of the distribution μ and the standard deviation of the distribution σ. The notation that we sometimes use to say that a variable X is normally distributed is as follows:

$$X \sim \text{Normal}(\mu, \sigma)$$

Of course, that's just notation. It doesn't tell us anything interesting about the normal distribution itself. As was the case with the binomial distribution, I have included the formula for the normal distribution in this book, because I think it's important enough that everyone who learns statistics should at least look at it, but since this is an introductory text I don't want to focus on it, so I've tucked it away in Table 9.2. Similarly, the R functions for the normal distribution are `dnorm()`, `pnorm()`, `qnorm()` and `rnorm()`. However, they behave in pretty much exactly the same way as the corresponding functions for the binomial distribution, so there's not a lot that you need to know. The only thing that I should point out is that the argument names for the parameters are `mean` and `sd`. In pretty much every other respect, there's nothing else to add.

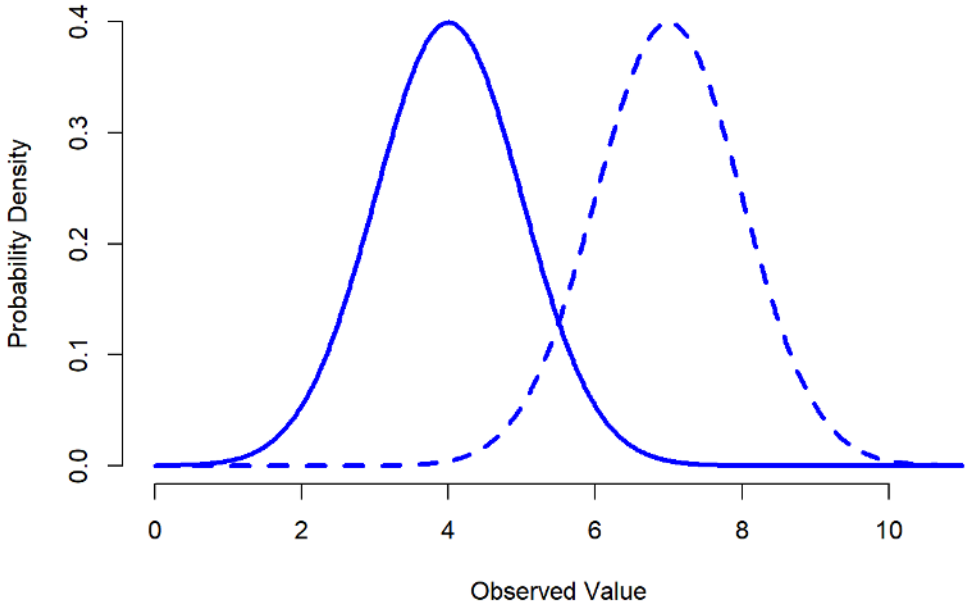


Figure 9.7: An illustration of what happens when you change the mean of a normal distribution. The solid line depicts a normal distribution with a mean of mu=4. The dashed line shows a normal distribution with a mean of mu=7. In both cases, the standard deviation is sigma=1. Not surprisingly, the two distributions have the same shape, but the dashed line is shifted to the right.

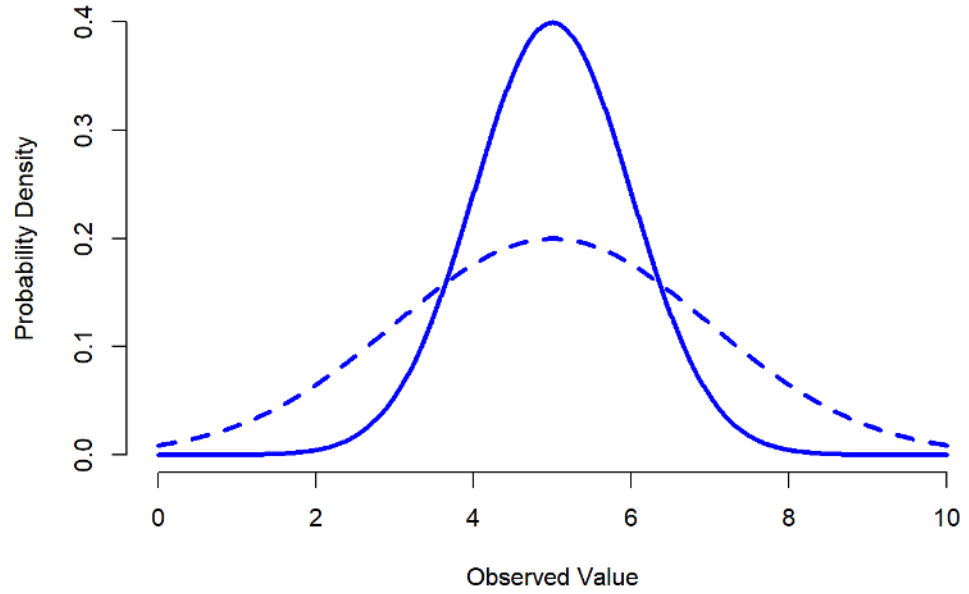


Figure 9.8: An illustration of what happens when you change the the standard deviation of a normal distribution. Both distributions plotted in this figure have a mean of mu=5, but they have different standard deviations. The solid line plots a distribution with standard deviation sigma=1, and the dashed line shows a distribution with standard deviation sigma=2. As a consequence, both distributions are "centred" on the same spot, but the dashed line is wider than the solid one.

Instead of focusing on the maths, let's try to get a sense for what it means for a variable to be normally distributed. To that end, have a look at Figure 9.6, which plots a normal distribution with mean $\mu=0$ and standard deviation $\sigma=1$. You can see where the name "bell curve" comes from: it looks a bit like a bell. Notice that, unlike the plots that I drew to illustrate the binomial distribution, the picture of the normal distribution in Figure 9.6 shows a smooth curve instead of "histogram-like" bars. This isn't an arbitrary choice: the normal distribution is continuous, whereas the binomial is discrete. For instance, in the die rolling example from the last section, it was possible to get 3 skulls or 4 skulls, but impossible to get 3.9 skulls. The figures that I drew in the previous section reflected this fact: in Figure 9.3, for instance, there's a bar located at X=3 and another one at X=4, but there's nothing in between. Continuous quantities don't have this constraint. For instance, suppose we're talking about the weather. The temperature on a pleasant Spring day could be 23 degrees, 24 degrees, 23.9 degrees, or anything in between since temperature is a continuous variable, and so a normal distribution might be quite appropriate for describing Spring temperatures.[145]

With this in mind, let's see if we can't get an intuition for how the normal distribution works. Firstly, let's have a look at what happens when we play around with the parameters of the distribution. To that end, Figure 9.7 plots normal distributions that have different means, but have the same standard deviation. As you might expect, all of these distributions have the same "width". The only difference between them is that they've been shifted to the left or to the right. In every other respect they're identical. In contrast, if we increase the standard deviation while keeping the mean constant, the peak of the distribution stays in the same place, but the distribution gets wider, as you can see in Figure 9.8. Notice, though, that when we widen the distribution, the height of the peak shrinks. This has to happen: in the same way that the heights of the bars that we used to draw a discrete binomial distribution have to *sum* to 1, the total *area under the curve* for the normal distribution must equal 1. Before moving on, I want to point out one important characteristic of the normal distribution. Irrespective of what the actual mean and standard deviation are, 68.3% of the area falls within 1 standard deviation of the mean. Similarly, 95.4% of the distribution falls within 2 standard deviations of the mean, and 99.7% of the distribution is within 3 standard deviations. This idea is illustrated in Figure **??**.

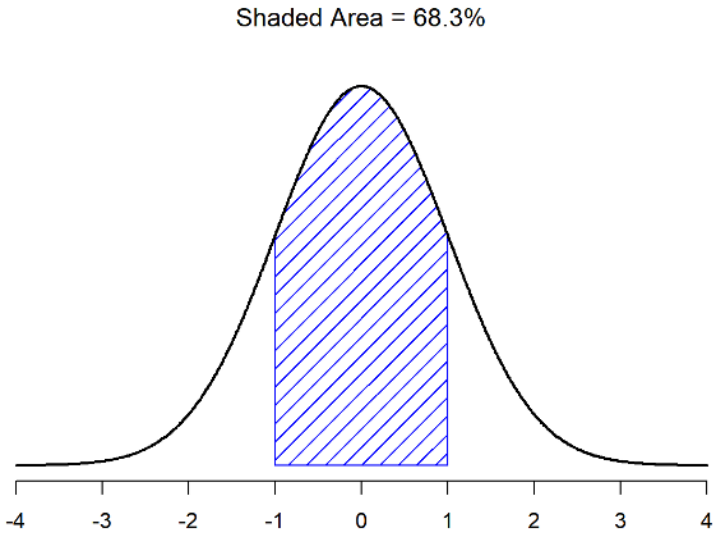


Figure 9.9: The area under the curve tells you the probability that an observation falls within a particular range. The solid lines plot normal distributions with mean mu=0 and standard deviation sigma=1. The shaded areas illustrate "areas under the curve" for two important cases. Here we can see that there is a 68.3% chance that an observation will fall within one standard deviation of the mean

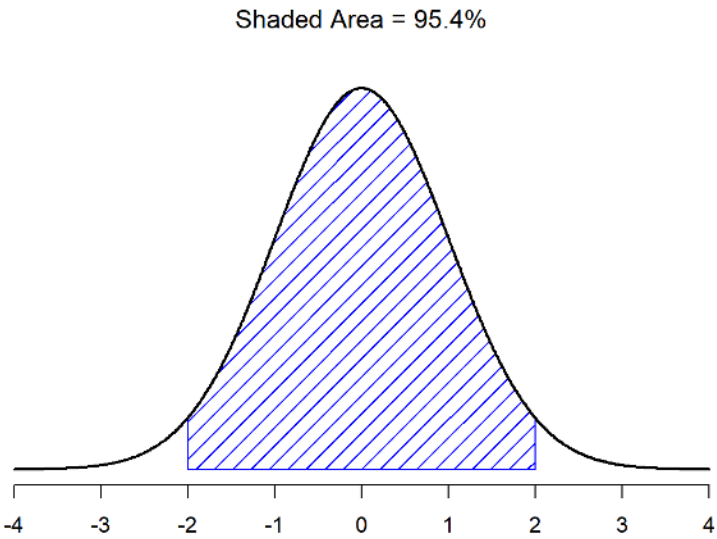


Figure 9.10: The area under the curve tells you the probability that an observation falls within a particular range. The solid lines plot normal distributions with mean mu=0 and standard deviation sigma=1. The shaded areas illustrate "areas under the curve" for two important cases. Here we see that there is a 95.4% chance that an observation will fall within two standard deviations of the mean.

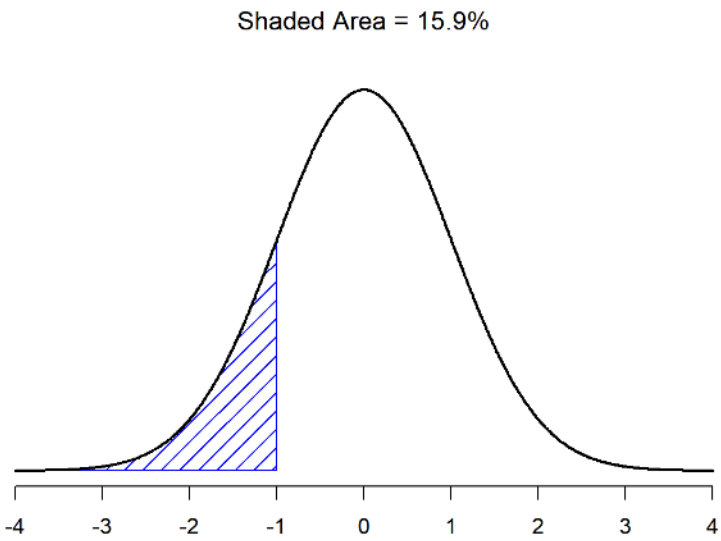


Figure 9.11: Two more examples of the “area under the curve idea”. There is a 15.9% chance that an observation is one standard deviation below the mean or smaller

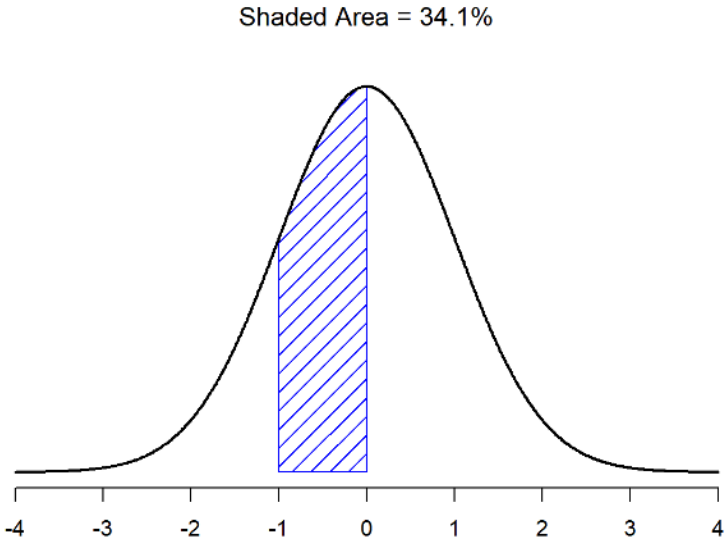


Figure 9.12: There is a 34.1% chance that the observation is greater than one standard deviation below the mean but still below the mean. Notice that if you add these two numbers together you get 15.9+34.1=50. For normally distributed data, there is a 50% chance that an observation falls below the mean. And of course that also implies that there is a 50% chance that it falls above the mean.

7.5.1 Probability density

There’s something I’ve been trying to hide throughout my discussion of the normal distribution, something that some introductory textbooks omit completely. They might be right to do so: this “thing” that I’m hiding is weird and counterintuitive even by the admittedly distorted standards that apply in statistics. Fortunately, it’s not something that you need to understand at a deep level in order to do basic statistics: rather, it’s something that starts to become important later on when you move beyond the basics. So, if it doesn’t make complete sense, don’t worry: try to make sure that you follow the gist of it.

Throughout my discussion of the normal distribution, there’s been one or two things that don’t quite make sense. Perhaps you noticed that the y-axis in these figures is labelled “Probability Density” rather than density. Maybe you noticed that I used p(X) instead of P(X) when giving the formula for the normal distribution. Maybe you’re wondering why R uses the “d” prefix for functions like `dnorm()` . And maybe, just maybe, you’ve been playing around with the `dnorm()` function, and you accidentally typed in a command like this:

```
dnorm( x = 1, mean = 1, sd = 0.1 )
```

```
## [1] 3.989423
```

And if you've done the last part, you're probably very confused. I've asked R to calculate the probability that `x = 1`, for a normally distributed variable with `mean = 1` and standard deviation `sd = 0.1`; and it tells me that the probability is 3.99. But, as we discussed earlier, probabilities *can't* be larger than 1. So either I've made a mistake, or that's not a probability.

As it turns out, the second answer is correct. What we've calculated here isn't actually a probability: it's something else. To understand what that something is, you have to spend a little time thinking about what it really *means* to say that X is a continuous variable. Let's say we're talking about the temperature outside. The thermometer tells me it's 23 degrees, but I know that's not really true. It's not *exactly* 23 degrees. Maybe it's 23.1 degrees, I think to myself. But I know that that's not really true either, because it might actually be 23.09 degrees. But, I know that… well, you get the idea. The tricky thing with genuinely continuous quantities is that you never really know exactly what they are.

Now think about what this implies when we talk about probabilities. Suppose that tomorrow's maximum temperature is sampled from a normal distribution with mean 23 and standard deviation 1. What's the probability that the temperature will be *exactly* 23 degrees? The answer is "zero", or possibly, "a number so close to zero that it might as well be zero". Why is this? It's like trying to throw a dart at an infinitely small dart board: no matter how good your aim, you'll never hit it. In real life you'll never get a value of exactly 23. It'll always be something like 23.1 or 22.99998 or something. In other words, it's completely meaningless to talk about the probability that the temperature is exactly 23 degrees. However, in everyday language, if I told you that it was 23 degrees outside and it turned out to be 22.9998 degrees, you probably wouldn't call me a liar. Because in everyday language, "23 degrees" usually means something like "somewhere between 22.5 and 23.5 degrees". And while it doesn't feel very meaningful to ask about the probability that the temperature is exactly 23 degrees, it does seem sensible to ask about the probability that the temperature lies between 22.5 and 23.5, or between 20 and 30, or any other range of temperatures.

The point of this discussion is to make clear that, when we're talking about continuous distributions, it's not meaningful to talk about the probability of a specific value. However, what we *can* talk about is the probability that the value lies within a particular range of values. To find out the probability associated with a particular range, what you need to do is calculate the "area under the curve". We've seen this concept already: in Figures 9.9 and (fig:sdnorm1b), the shaded areas shown depict genuine probabilities (e.g., in Figure 9.9 it shows the probability of observing a value that falls within 1 standard deviation of the mean).

Okay, so that explains part of the story. I've explained a little bit about how continuous probability distributions should be interpreted (i.e., area under the curve is the key thing), but I haven't actually explained what the `dnorm()` function actually calculates. Equivalently, what does the formula for p(x) that I described earlier actually mean? Obviously, p(x) doesn't describe a probability, but what is it? The name for this quantity p(x) is a ***probability density***, and in terms of the plots we've been drawing, it corresponds to the *height* of the curve. The densities themselves aren't meaningful in and of themselves: but they're "rigged" to ensure that the *area* under the curve is always interpretable as genuine probabilities. To be honest, that's about as much as you really need to know for now.[146]

This page titled 7.5: The Normal Distribution is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **9.5: The Normal Distribution** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

7.6: Other Useful Distributions

The normal distribution is the distribution that statistics makes most use of (for reasons to be discussed shortly), and the binomial distribution is a very useful one for lots of purposes. But the world of statistics is filled with probability distributions, some of which we'll run into in passing. In particular, the three that will appear in this book are the t distribution, the χ^2 distribution and the F distribution. I won't give formulas for any of these, or talk about them in too much detail, but I will show you some pictures.

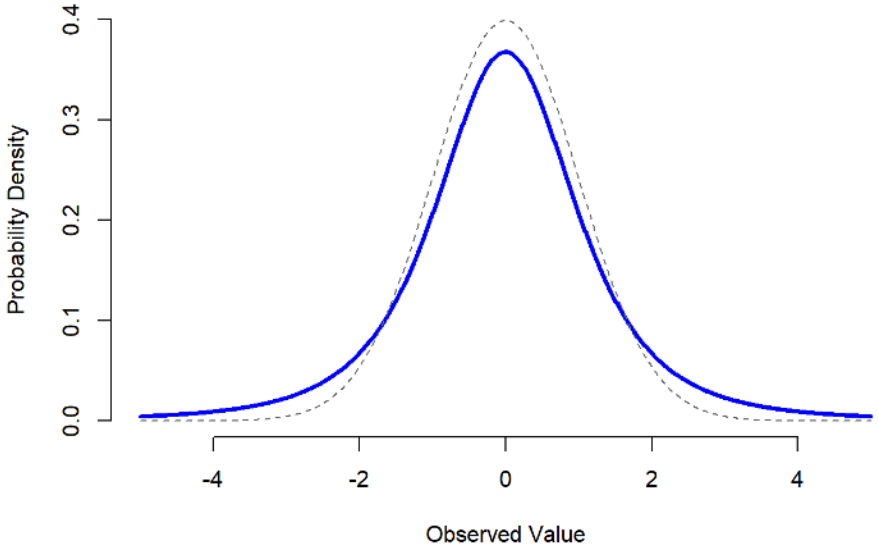


Figure 9.13: A t distribution with 3 degrees of freedom (solid line). It looks similar to a normal distribution, but it's not quite the same. For comparison purposes, I've plotted a standard normal distribution as the dashed line. Note that the "tails" of the t distribution are "heavier" (i.e., extend further outwards) than the tails of the normal distribution? That's the important difference between the two.

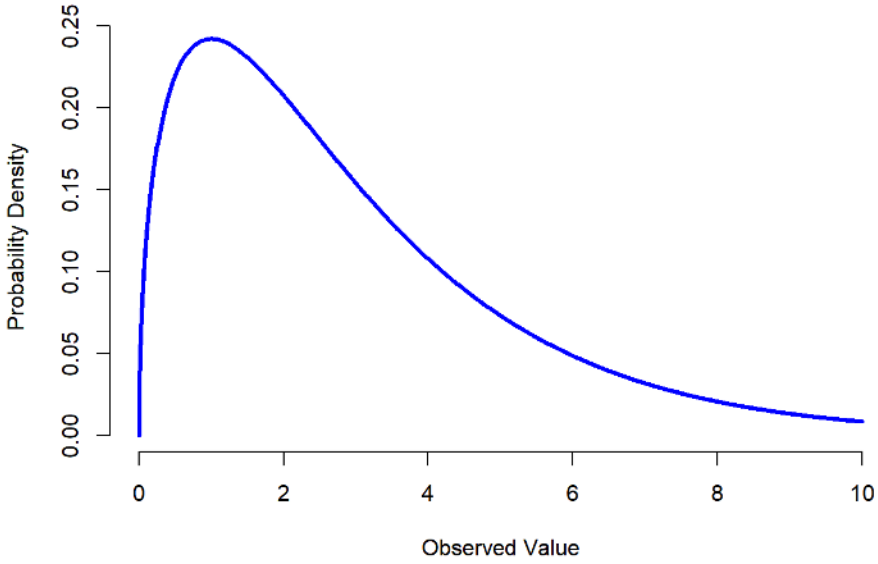


Figure 9.14: A chi^2 distribution with 3 degrees of freedom. Notice that the observed values must always be greater than zero, and that the distribution is pretty skewed. These are the key features of a chi-square distribution.

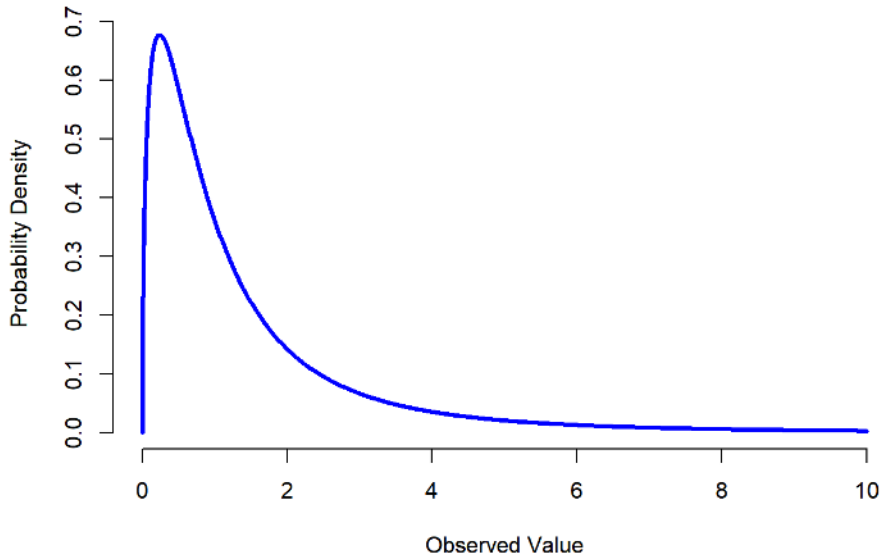


Figure 9.15: An F distribution with 3 and 5 degrees of freedom. Qualitatively speaking, it looks pretty similar to a chi-square distribution, but they're not quite the same in general.

- The ***t distribution*** is a continuous distribution that looks very similar to a normal distribution, but has heavier tails: see Figure 9.13. This distribution tends to arise in situations where you think that the data actually follow a normal distribution, but you don't know the mean or standard deviation. As you might expect, the relevant R functions are `dt()`, `pt()`, `qt()` and `rt()`, and we'll run into this distribution again in Chapter 13.
- The ***χ^2 distribution*** is another distribution that turns up in lots of different places. The situation in which we'll see it is when doing categorical data analysis (Chapter 12), but it's one of those things that actually pops up all over the place. When you dig into the maths (and who doesn't love doing that?), it turns out that the main reason why the $\chi 2$ distribution turns up all over the place is that, if you have a bunch of variables that are normally distributed, square their values and then add them up (a procedure referred to as taking a "sum of squares"), this sum has a χ^2 distribution. You'd be amazed how often this fact turns out to be useful. Anyway, here's what a $\chi 2$ distribution looks like: Figure 9.14. Once again, the R commands for this one are pretty predictable: `dchisq()`, `pchisq()`, `qchisq()`, `rchisq()`.
- The ***F distribution*** looks a bit like a χ^2 distribution, and it arises whenever you need to compare two χ^2 distributions to one another. Admittedly, this doesn't exactly sound like something that any sane person would want to do, but it turns out to be very important in real world data analysis. Remember when I said that χ^2 turns out to be the key distribution when we're taking a "sum of squares"? Well, what that means is if you want to compare two different "sums of squares", you're probably talking about something that has an F distribution. Of course, as yet I still haven't given you an example of anything that involves a sum of squares, but I will… in Chapter 14. And that's where we'll run into the F distribution. Oh, and here's a picture: Figure 9.15. And of course we can get R to do things with F distributions just by using the commands `df()`, `pf()`, `qf()` and `rf()`.

Because these distributions are all tightly related to the normal distribution and to each other, and because they are will turn out to be the important distributions when doing inferential statistics later in this book, I think it's useful to do a little demonstration using R, just to "convince ourselves" that these distributions really are related to each other in the way that they're supposed to be. First, we'll use the `rnorm()` function to generate 1000 normally-distributed observations:

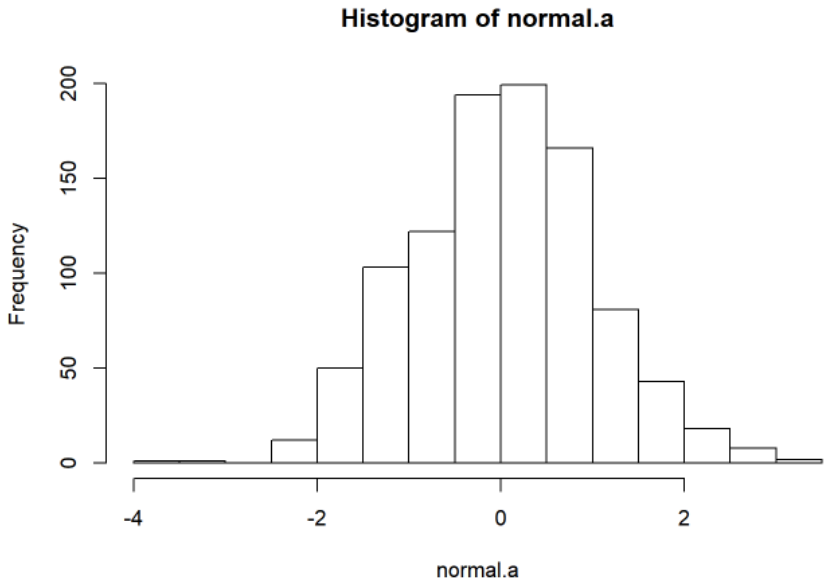
```
normal.a <- rnorm( n=1000, mean=0, sd=1 )
print(head(normal.a))
```

```
## [1] -0.4728528 -0.4483396 -0.5134192  2.1540478 -0.5104661  0.3013308
```

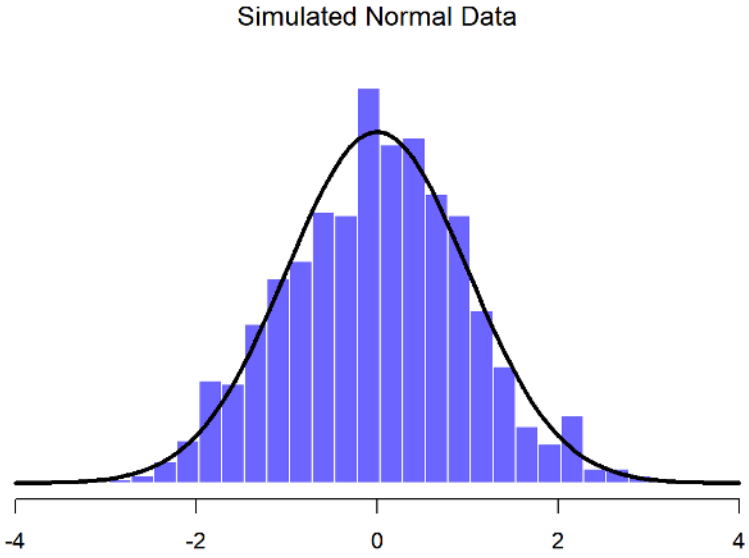
So the `normal.a` variable contains 1000 numbers that are normally distributed, and have mean 0 and standard deviation 1, and the actual print out of these numbers goes on for rather a long time. Note that, because the default parameters of the `rnorm()`

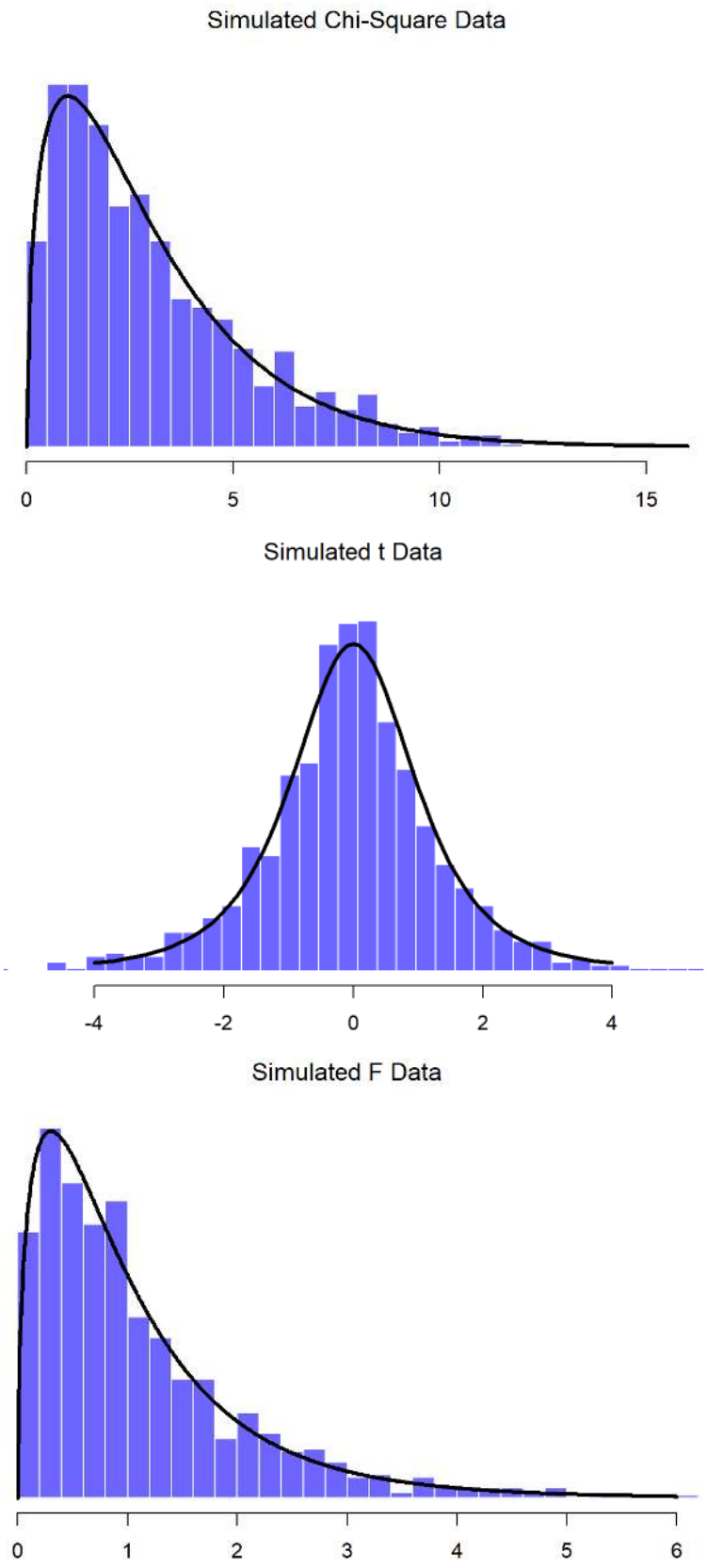
function are `mean=0` and `sd=1` , I could have shortened the command to `rnorm( n=1000 )` . In any case, what we can do is use the `hist()` function to draw a histogram of the data, like so:

```
hist( normal.a )
```



If you do this, you should see something similar to Figure **??**. Your plot won't look quite as pretty as the one in the figure, of course, because I've played around with all the formatting (see Chapter 6), and I've also plotted the true distribution of the data as a solid black line (i.e., a normal distribution with mean 0 and standard deviation 1) so that you can compare the data that we just generated to the true distribution.





In the previous example all I did was generate lots of normally distributed observations using `rnorm()` and then compared those to the true probability distribution in the figure (using `dnorm()` to generate the black line in the figure, but I didn't show the commmands for that). Now let's try something trickier. We'll try to generate some observations that follow a chi-square

distribution with 3 degrees of freedom, but instead of using `rchisq()`, we'll start with variables that are normally distributed, and see if we can exploit the known relationships between normal and chi-square distributions to do the work. As I mentioned earlier, a chi-square distribution with k degrees of freedom is what you get when you take k normally-distributed variables (with mean 0 and standard deviation 1), square them, and add them up. Since we want a chi-square distribution with 3 degrees of freedom, we'll need to supplement our `normal.a` data with two more sets of normally-distributed observations, imaginatively named `normal.b` and `normal.c`:

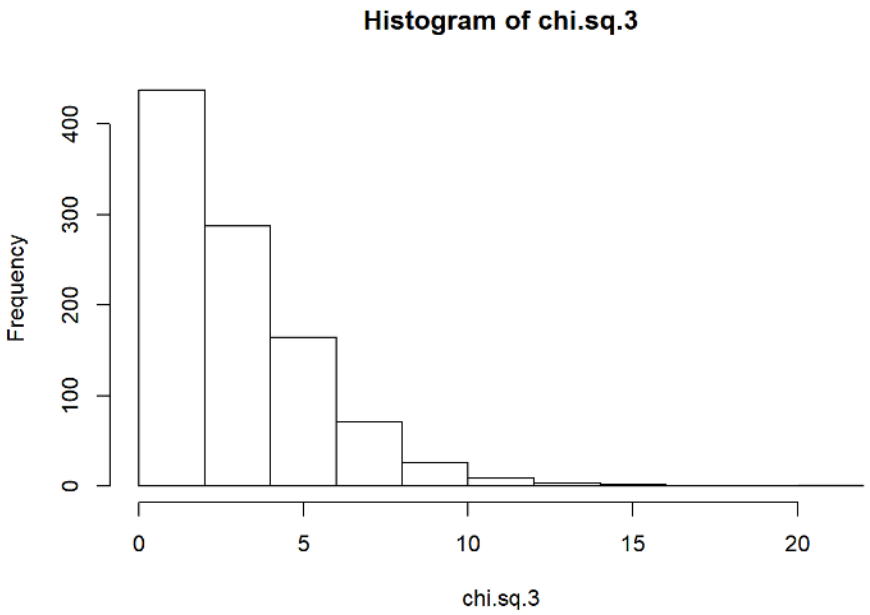
```
normal.b <- rnorm( n=1000 )  # another set of normally distributed data
normal.c <- rnorm( n=1000 )  # and another!
```

Now that we've done that, the theory says we should square these and add them together, like this

```
chi.sq.3 <- (normal.a)^2 + (normal.b)^2 + (normal.c)^2
```

and the resulting `chi.sq.3` variable should contain 1000 observations that follow a chi-square distribution with 3 degrees of freedom. You can use the `hist()` function to have a look at these observations yourself, using a command like this,

```
hist( chi.sq.3 )
```



and you should obtain a result that looks pretty similar to the chi-square plot in Figure **??**. Once again, the plot that I've drawn is a little fancier: in addition to the histogram of `chi.sq.3`, I've also plotted a chi-square distribution with 3 degrees of freedom. It's pretty clear that – even though I used `rnorm()` to do all the work rather than `rchisq()` – the observations stored in the `chi.sq.3` variable really do follow a chi-square distribution. Admittedly, this probably doesn't seem all that interesting right now, but later on when we start encountering the chi-square distribution in Chapter 12, it will be useful to understand the fact that these distributions are related to one another.

We can extend this demonstration to the t distribution and the F distribution. Earlier, I implied that the t distribution is related to the normal distribution when the standard deviation is unknown. That's certainly true, and that's the what we'll see later on in Chapter 13, but there's a somewhat more precise relationship between the normal, chi-square and t distributions. Suppose we "scale" our chi-square data by dividing it by the degrees of freedom, like so

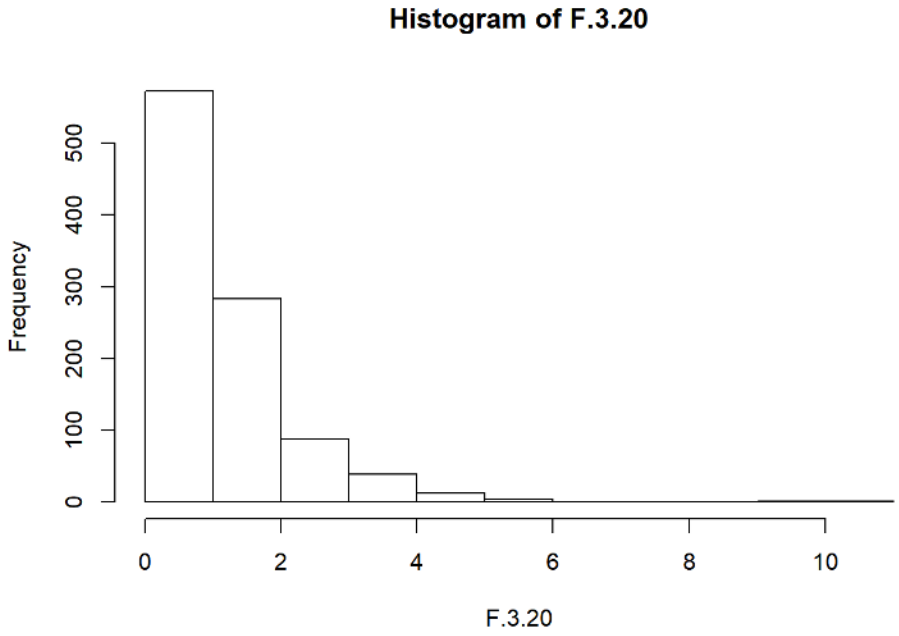
```
scaled.chi.sq.3 <- chi.sq.3 / 3
```

We then take a set of normally distributed variables and divide them by (the square root of) our scaled chi-square variable which had df=3, and the result is a t distribution with 3 degrees of freedom:

```
normal.d <- rnorm( n=1000 )  # yet another set of normally distributed data
t.3 <- normal.d / sqrt( scaled.chi.sq.3 )  # divide by square root of scaled chi-squa
```

If we plot the histogram of `t.3`, we end up with something that looks very similar to the t distribution in Figure **??**. Similarly, we can obtain an F distribution by taking the ratio between two scaled chi-square distributions. Suppose, for instance, we wanted to generate data from an F distribution with 3 and 20 degrees of freedom. We could do this using `df()`, but we could also do the same thing by generating two chi-square variables, one with 3 degrees of freedom, and the other with 20 degrees of freedom. As the example with `chi.sq.3` illustrates, we can actually do this using `rnorm()` if we really want to, but this time I'll take a short cut:

```
chi.sq.20 <- rchisq( 1000, 20)  # generate chi square data with df = 20...
scaled.chi.sq.20 <- chi.sq.20 / 20  # scale the chi square variable...
F.3.20 <-  scaled.chi.sq.3  / scaled.chi.sq.20 # take the ratio of the two chi square
hist( F.3.20 ) # ... and draw a picture
```



The resulting `F.3.20` variable does in fact store variables that follow an F distribution with 3 and 20 degrees of freedom. This is illustrated in Figure **??**, which plots the histgram of the observations stored in `F.3.20` against the true F distribution with df1=3 and df2=20. Again, they match.

Okay, time to wrap this section up. We've seen three new distributions: χ^2, t and F. They're all continuous distributions, and they're all closely related to the normal distribution. I've talked a little bit about the precise nature of this relationship, and shown you some R commands that illustrate this relationship. The key thing for our purposes, however, is not that you have a deep understanding of all these different distributions, nor that you remember the precise relationships between them. The main thing is that you grasp the basic idea that these distributions are all deeply related to one another, and to the normal distribution. Later on in this book, we're going to run into data that are normally distributed, or at least assumed to be normally distributed. What I want you to understand right now is that, if you make the assumption that your data are normally distributed, you shouldn't be surprised to see χ^2, t and F distributions popping up all over the place when you start trying to do your data analysis.

This page titled 7.6: Other Useful Distributions is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **9.6: Other Useful Distributions** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

7.7: Summary

In this chapter we've talked about probability. We've talked what probability means, and why statisticians can't agree on what it means. We talked about the rules that probabilities have to obey. And we introduced the idea of a probability distribution, and spent a good chunk of the chapter talking about some of the more important probability distributions that statisticians work with. The section by section breakdown looks like this:

- Probability theory versus statistics (Section 9.1)
- Frequentist versus Bayesian views of probability (Section 9.2)
- Basics of probability theory (Section 9.3)
- Binomial distribution (Section 9.4), normal distribution (Section 9.5), and others (Section 9.6)

As you'd expect, my coverage is by no means exhaustive. Probability theory is a large branch of mathematics in its own right, entirely separate from its application to statistics and data analysis. As such, there are thousands of books written on the subject and universities generally offer multiple classes devoted entirely to probability theory. Even the "simpler" task of documenting standard probability distributions is a big topic. I've described five standard probability distributions in this chapter, but sitting on my bookshelf I have a 45-chapter book called "Statistical Distributions" Evans, Hastings, and Peacock (2011) that lists a *lot* more than that. Fortunately for you, very little of this is necessary. You're unlikely to need to know dozens of statistical distributions when you go out and do real world data analysis, and you definitely won't need them for this book, but it never hurts to know that there's other possibilities out there.

Picking up on that last point, there's a sense in which this whole chapter is something of a digression. Many undergraduate psychology classes on statistics skim over this content very quickly (I know mine did), and even the more advanced classes will often "forget" to revisit the basic foundations of the field. Most academic psychologists would not know the difference between probability and density, and until recently very few would have been aware of the difference between Bayesian and frequentist probability. However, I think it's important to understand these things before moving onto the applications. For example, there are a lot of rules about what you're "allowed" to say when doing statistical inference, and many of these can seem arbitrary and weird. However, they start to make sense if you understand that there is this Bayesian/frequentist distinction. Similarly, in Chapter 13 we're going to talk about something called the t-test, and if you really want to have a grasp of the mechanics of the t-test it really helps to have a sense of what a t-distribution actually looks like. You get the idea, I hope.

References

Fisher, R. 1922b. "On the Mathematical Foundation of Theoretical Statistics." *Philosophical Transactions of the Royal Society A* 222: 309–68.

Meehl, P. H. 1967. "Theory Testing in Psychology and Physics: A Methodological Paradox." *Philosophy of Science* 34: 103–15.

Evans, M., N. Hastings, and B. Peacock. 2011. *Statistical Distributions (3rd Ed)*. Wiley.

142. This doesn't mean that frequentists can't make hypothetical statements, of course; it's just that if you want to make a statement about probability, then it must be possible to redescribe that statement in terms of a sequence of potentially observable events, and the relative frequencies of different outcomes that appear within that sequence.
143. Note that the term "success" is pretty arbitrary, and doesn't actually imply that the outcome is something to be desired. If θ referred to the probability that any one passenger gets injured in a bus crash, I'd still call it the success probability, but that doesn't mean I want people to get hurt in bus crashes!
144. Since computers are deterministic machines, they can't actually produce truly random behaviour. Instead, what they do is take advantage of various mathematical functions that share a lot of similarities with true randomness. What this means is that any random numbers generated on a computer are *pseudorandom*, and the quality of those numbers depends on the specific method used. By default R uses the "Mersenne twister" method. In any case, you can find out more by typing `?Random` , but as usual the R help files are fairly dense.
145. In practice, the normal distribution is so handy that people tend to use it even when the variable isn't actually continuous. As long as there are enough categories (e.g., Likert scale responses to a questionnaire), it's pretty standard practice to use the normal distribution as an approximation. This works out much better in practice than you'd think.

146. For those readers who know a little calculus, I'll give a slightly more precise explanation. In the same way that probabilities are non-negative numbers that must sum to 1, probability densities are non-negative numbers that must integrate to 1 (where the integral is taken across all possible values of X). To calculate the probability that X falls between a and b we calculate the definite integral of the density function over the corresponding range, $\int_a^b p(x)dx$. If you don't remember or never learned calculus, don't worry about this. It's not needed for this book.

This page titled 7.7: Summary is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **9.7: Summary** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

7.8: Statistical Literacy

Learning Objectives

- Learning to use the Base Rate information to compute probability

School shooting: The warning signs

This webpage gives the FBI list of warning signs for school shooters.

Example 7.8.1: What do you think?

Do you think it is likely that someone showing a majority of these signs would actually shoot people in school?

Solution

Fortunately the vast majority of students do not become shooters. It is necessary to take this base rate information into account in order to compute the probability that any given student will be a shooter. The warning signs are unlikely to be sufficiently predictive to warrant the conclusion that a student will become a shooter. If an action is taken on the basis of these warning signs, it is likely that the student involved would never have become a shooter even without the action.

This page titled 7.8: Statistical Literacy is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **5.15: Statistical Literacy** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

7.E: Probability (Exercises)

You may want to use the Binomial Calculator for some of these exercises.

General Questions

Q1

a. What is the probability of rolling a pair of dice and obtaining a total score of 9 or more?
b. What is the probability of rolling a pair of dice and obtaining a total score of 7? (relevant section)

Q2

A box contains four black pieces of cloth, two striped pieces, and six dotted pieces. A piece is selected randomly and then placed back in the box. A second piece is selected randomly. What is the probability that:

a. both pieces are dotted?
b. the first piece is black and the second piece is dotted?
c. one piece is black and one piece is striped?
(relevant section)

Q3

A card is drawn at random from a deck.

a. What is the probability that it is an ace or a king?
b. What is the probability that it is either a red card or a black card? (relevant section)

Q4

The probability that you will win a game is 0.45.

a. If you play the game 80 times, what is the most likely number of wins?
b. What are the mean and standard deviation of a binomial distribution with $\pi = 0.45$ and $N = 80$? (relevant section)

Q5

A fair coin is flipped 9 times. What is the probability of getting exactly 6 heads? (relevant section)

Q6

When Susan and Jessica play a card game, Susan wins 60% of the time. If they play 9 games, what is the probability that Jessica will have won more games than Susan? (relevant section)

Q7

You flip a coin three times.

a. What is the probability of getting heads on only one of your flips?
b. What is the probability of getting heads on at least one flip? (relevant section & relevant section)

Q8

A test correctly identifies a disease in 95% of people who have it. It correctly identifies no disease in 94% of people who do not have it. In the population, 3% of the people have the disease. What is the probability that you have the disease if you tested positive? (relevant section)

Q9

A jar contains 10 blue marbles, 5 red marbles, 4 green marbles, and 1 yellow marble. Two marbles are chosen (without replacement).

a. What is the probability that one will be green and the other red?
b. What is the probability that one will be blue and the other yellow? (relevant section)

Q10

You roll a fair die five times, and you get a 6 each time. What is the probability that you get a 6 on the next roll? (relevant section)

Q11

You win a game if you roll a die and get a 2 or a 5. You play this game 60 times.

a. What is the probability that you win between 5 and 10 times (inclusive)?
b. What is the probability that you will win the game at least 15 times?
c. What is the probability that you will win the game at least 40 times?
d. What is the most likely number of wins.
e. What is the probability of obtaining the number of wins in d?
(relevant section)

Q12

In a baseball game, Tommy gets a hit 30% of the time when facing this pitcher. Joey gets a hit 25% of the time. They are both coming up to bat this inning.

a. What is the probability that Joey or Tommy (but not both) will get a hit?
b. What is the probability that neither player gets a hit?
c. What is the probability that they both get a hit? (relevant section)

Q13

An unfair coin has a probability of coming up heads of 0.65. The coin is flipped 50 times. What is the probability it will come up heads 25 or fewer times? (Give answer to at least 3 decimal places). (relevant section)

Q14

You draw two cards from a deck, what is the probability that

a. both of them are face cards (king, queen, or jack)?
b. What is the probability that you draw two cards from a deck and both of them are hearts? (relevant section)

Q15

True/False: You are more likely to get a pattern of $HTHHHTHTTH$ than $HHHHHHHHTT$ when you flip a coin 10 times. (relevant section)

Q16

True/False: Suppose that at your regular physical exam you test positive for a relatively rare disease. You will need to start taking medicine if you have the disease, so you ask your doctor about the accuracy of the test. It turns out that the test is 98% accurate. The probability that you have Disease X is therefore 0.98 and the probability that you do not have it is 0.02. (relevant section)

Questions from Case Studies

The following questions are from the Diet and Health (DH) case study.

Q17

(DH#1)

a. What percentage of people on the AHA diet had some sort of illness or death?
b. What is the probability that if you randomly selected a person on the AHA diet, he or she would have some sort of illness or death? (relevant section)
c. If 3 people on the AHA diet are chosen at random, what is the probability that they will all be healthy? (relevant section)

Q18

(DH#2)

a. What percentage of people on the Mediterranean diet had some sort of illness or death?
b. What is the probability that if you randomly selected a person on the Mediterranean diet, he or she would have some sort of illness or death? (relevant section)
c. What is the probability that if you randomly selected a person on the Mediterranean diet, he or she would have cancer? (relevant section)
d. If you randomly select five people from the Mediterranean diet, what is the probability that they would all be healthy? (relevant section)

The following questions are from (reproduced with permission)



Visit the site

Q19

Five faces of a fair die are painted black, and one face is painted white. The die is rolled six times. Which of the following results is more likely?
a. Black side up on five of the rolls; white side up on the other roll
b. Black side up on all six rolls
c. a and b are equally likely

Q20

One of the items on the student survey for an introductory statistics course was "Rate your intelligence on a scale of 1 to 10." The distribution of this variable for the 100 women in the class is presented below. What is the probability of randomly selecting a woman from the class who has an intelligence rating that is LESS than seven (7)?

Intelligence Rating	Count
5	12
6	24
7	38
8	23
9	2
10	1

a. $(12+24)/100=0.36$

b. $(12+24+38)/100=0.74$

c. $38/100=0.38$

d. $(23+2+1)/100=0.26$

e. None of the above.

Q21

You roll 2 fair six-sided dice. Which of the following outcomes is most likely to occur on the next roll?

a. Getting double 3.
b. Getting a 3 and a 4.

c. They are equally likely. Explain your choice.

Q22

If Tahnee flips a coin 10 times, and records the results (Heads or Tails), which outcome below is more likely to occur, A or B? Explain your choice.

Throw Number	1	2	3	4	5	6	7	8	9	10
A	H	T	T	H	T	H	H	T	T	T
B	H	T	H	T	H	T	H	T	H	T

Q23

A bowl has 100 wrapped hard candies in it. 20 are yellow, 50 are red, and 30 are blue. They are well mixed up in the bowl. Jenny pulls out a handful of 10 candies, counts the number of reds, and tells her teacher. The teacher writes the number of red candies on a list. Then, Jenny puts the candies back into the bowl, and mixes them all up again. Four of Jenny's classmates, Jack, Julie, Jason, and Jerry do the same thing. They each pick ten candies, count the reds, and the teacher writes down the number of reds. Then they put the candies back and mix them up again each time. The teacher's list for the number of reds is most likely to be (please select one):

a. $8, 9, 7, 10, 9$
b. $3, 7, 5, 8, 5$
c. $5, 5, 5, 5, 5$
d. $2, 4, 3, 4, 3$
e. $3, 0, 9, 2, 8$

Q24

An insurance company writes policies for a large number of newly-licensed drivers each year. Suppose 40% of these are low-risk drivers, 40% are moderate risk, and 20% are high risk. The company has no way to know which group any individual driver falls in when it writes the policies. None of the low-risk drivers will have an at-fault accident in the next year, but 10% of the moderate-risk and 20% of the high-risk drivers will have such an accident. If a driver has an at-fault accident in the next year, what is the probability that he or she is high-risk?

Q25

You are to participate in an exam for which you had no chance to study, and for that reason cannot do anything but guess for each question (all questions being of the multiple choice type, so the chance of guessing the correct answer for each question is $1/d$, d being the number of options per question; so in case of a 4-choice question, your chance is 0.25). Your instructor offers you the opportunity to choose amongst the following exam formats:

a. 6 questions of the 4-choice type; you pass when 5 or more answers are correct
b. 5 questions of the 5-choice type; you pass when 4 or more answers are correct
c. 4 questions of the 10-choice type; you pass when 3 or more answers are correct.

Rank the three exam formats according to their attractiveness. It should be clear that the format with the highest probability to pass is the most attractive format. Which would you choose and why?

Q26

Consider the question of whether the home team wins more than half of its games in the National Basketball Association. Suppose that you study a simple random sample of 80 professional basketball games and find that 52 of them are won by the home team.

a. Assuming that there is no home court advantage and that the home team therefore wins 50% of its games in the long run, determine the probability that the home team would win 65% or more of its games in a simple random sample of 80 games.

b. Does the sample information (that 52 of a random sample of 80 games are won by the home team) provide strong evidence that the home team wins more than half of its games in the long run? Explain.

Q27

A refrigerator contains 6 apples, 5 oranges, 10 bananas, 3 pears, 7 peaches, 11 plums, and 2 mangos.

a. Imagine you stick your hand in this refrigerator and pull out a piece of fruit at random. What is the probability that you will pull out a pear?
b. Imagine now that you put your hand in the refrigerator and pull out a piece of fruit. You decide you do not want to eat that fruit so you put it back into the refrigerator and pull out another piece of fruit. What is the probability that the first piece of fruit you pull out is a banana and the second piece you pull out is an apple?
c. What is the probability that you stick your hand in the refrigerator one time and pull out a mango or an orange?

Select Answers

S1

a. $5/18 = 0.278$

S2

b. $1/6 = 0.167$

S3

a. $2/13 = 0.154$

S4

b. mean = 36; SD = 4.24

S5

0.164

S7

b. $7/8 = 0.875$

S9

a. $2/19 = 0.105$

S11

b. 0.937

S12

c. 0.075

S14

b. $1/17 = 0.0588$

S17

c. 0.493

S18

b. 0.10

This page titled 7.E: Probability (Exercises) is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **5.E: Probability (Exercises)** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

CHAPTER OVERVIEW

8: Estimating Unknown Quantities from a Sample

At the start of the last chapter I highlighted the critical distinction between *descriptive statistics* and *inferential statistics*. As discussed in Chapter 5, the role of descriptive statistics is to concisely summarise what we *do* know. In contrast, the purpose of inferential statistics is to "learn what we do not know from what we do". Now that we have a foundation in probability theory, we are in a good position to think about the problem of statistical inference. What kinds of things would we like to learn about? And how do we learn them? These are the questions that lie at the heart of inferential statistics, and they are traditionally divided into two "big ideas": estimation and hypothesis testing. The goal in this chapter is to introduce the first of these big ideas, estimation theory, but I'm going to witter on about sampling theory first because estimation theory doesn't make sense until you understand sampling. As a consequence, this chapter divides naturally into two parts Sections 10.1 through 10.3 are focused on sampling theory, and Sections 10.4 and 10.5 make use of sampling theory to discuss how statisticians think about estimation.

8.1: Samples, Populations and Sampling

8.2: The Law of Large Numbers

8.3: Sampling Distributions and the Central Limit Theorem

8.4: Estimating Population Parameters

8.5: Estimating a Confidence Interval

8.6: Summary

8.7: Statistical Literacy

8.E: Estimation (Exercises)

This page titled 8: Estimating Unknown Quantities from a Sample is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

8.1: Samples, Populations and Sampling

In the prelude to Part I discussed the riddle of induction, and highlighted the fact that *all* learning requires you to make assumptions. Accepting that this is true, our first task to come up with some fairly general assumptions about data that make sense. This is where ***sampling theory*** comes in. If probability theory is the foundations upon which all statistical theory builds, sampling theory is the frame around which you can build the rest of the house. Sampling theory plays a huge role in specifying the assumptions upon which your statistical inferences rely. And in order to talk about "making inferences" the way statisticians think about it, we need to be a bit more explicit about what it is that we're drawing inferences *from* (the sample) and what it is that we're drawing inferences *about* (the population).

In almost every situation of interest, what we have available to us as researchers is a ***sample*** of data. We might have run experiment with some number of participants; a polling company might have phoned some number of people to ask questions about voting intentions; etc. Regardless: the data set available to us is finite, and incomplete. We can't possibly get every person in the world to do our experiment; a polling company doesn't have the time or the money to ring up every voter in the country etc. In our earlier discussion of descriptive statistics (Chapter 5, this sample was the only thing we were interested in. Our only goal was to find ways of describing, summarising and graphing that sample. This is about to change.

8.1.1 Defining a population

A sample is a concrete thing. You can open up a data file, and there's the data from your sample. A ***population***, on the other hand, is a more abstract idea. It refers to the set of all possible people, or all possible observations, that you want to draw conclusions about, and is generally *much* bigger than the sample. In an ideal world, the researcher would begin the study with a clear idea of what the population of interest is, since the process of designing a study and testing hypotheses about the data that it produces does depend on the population about which you want to make statements. However, that doesn't always happen in practice: usually the researcher has a fairly vague idea of what the population is and designs the study as best he/she can on that basis.

Sometimes it's easy to state the population of interest. For instance, in the "polling company" example that opened the chapter, the population consisted of all voters enrolled at the a time of the study – millions of people. The sample was a set of 1000 people who all belong to that population. In most situations the situation is much less simple. In a typical a psychological experiment, determining the population of interest is a bit more complicated. Suppose I run an experiment using 100 undergraduate students as my participants. My goal, as a cognitive scientist, is to try to learn something about how the mind works. So, which of the following would count as "the population":

- All of the undergraduate psychology students at the University of Adelaide?
- Undergraduate psychology students in general, anywhere in the world?
- Australians currently living?
- Australians of similar ages to my sample?
- Anyone currently alive?
- Any human being, past, present or future?
- Any biological organism with a sufficient degree of intelligence operating in a terrestrial environment?
- Any intelligent being?

Each of these defines a real group of mind-possessing entities, all of which might be of interest to me as a cognitive scientist, and it's not at all clear which one ought to be the true population of interest. As another example, consider the Wellesley-Croker game that we discussed in the prelude. The sample here is a specific sequence of 12 wins and 0 losses for Wellesley. What is the population?

- All outcomes until Wellesley and Croker arrived at their destination?
- All outcomes if Wellesley and Croker had played the game for the rest of their lives?
- All outcomes if Wellseley and Croker lived forever and played the game until the world ran out of hills?
- All outcomes if we created an infinite set of parallel universes and the Wellesely/Croker pair made guesses about the same 12 hills in each universe?

Again, it's not obvious what the population is.

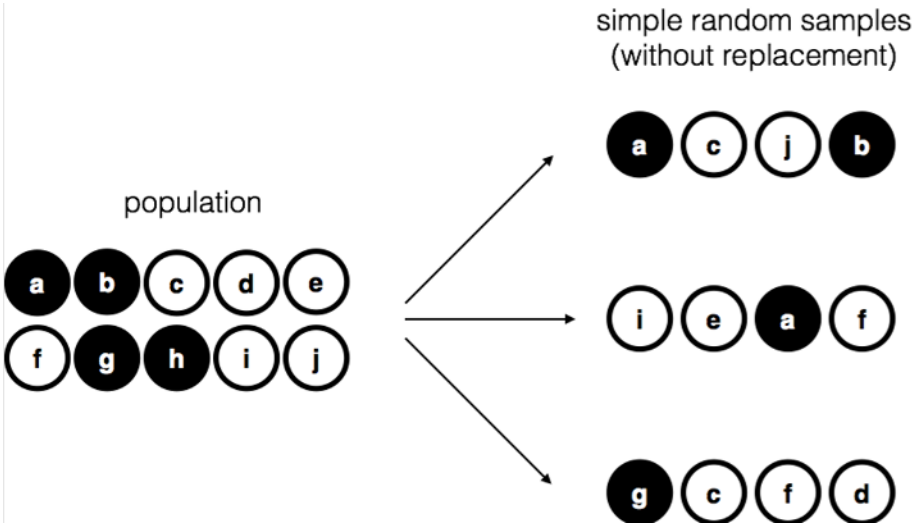


Figure 10.1: Simple random sampling without replacement from a finite population

Irrespective of how I define the population, the critical point is that the sample is a subset of the population, and our goal is to use our knowledge of the sample to draw inferences about the properties of the population. The relationship between the two depends on the *procedure* by which the sample was selected. This procedure is referred to as a ***sampling method***, and it is important to understand why it matters.

To keep things simple, let's imagine that we have a bag containing 10 chips. Each chip has a unique letter printed on it, so we can distinguish between the 10 chips. The chips come in two colours, black and white. This set of chips is the population of interest, and it is depicted graphically on the left of Figure 10.1. As you can see from looking at the picture, there are 4 black chips and 6 white chips, but of course in real life we wouldn't know that unless we looked in the bag. Now imagine you run the following "experiment": you shake up the bag, close your eyes, and pull out 4 chips without putting any of them back into the bag. First out comes the a chip (black), then the c chip (white), then j (white) and then finally b (black). If you wanted, you could then put all the chips back in the bag and repeat the experiment, as depicted on the right hand side of Figure 10.1. Each time you get different results, but the procedure is identical in each case. The fact that the same procedure can lead to different results each time, we refer to it as a *random* process.[147] However, because we shook the bag before pulling any chips out, it seems reasonable to think that every chip has the same chance of being selected. A procedure in which every member of the population has the same chance of being selected is called a ***simple random sample***. The fact that we did *not* put the chips back in the bag after pulling them out means that you can't observe the same thing twice, and in such cases the observations are said to have been sampled ***without replacement***.

To help make sure you understand the importance of the sampling procedure, consider an alternative way in which the experiment could have been run. Suppose that my 5-year old son had opened the bag, and decided to pull out four black chips without putting any of them back in the bag. This *biased* sampling scheme is depicted in Figure 10.2. Now consider the evidentiary value of seeing 4 black chips and 0 white chips. Clearly, it depends on the sampling scheme, does it not? If you know that the sampling scheme is biased to select only black chips, then a sample that consists of only black chips doesn't tell you very much about the population! For this reason, statisticians really like it when a data set can be considered a simple random sample, because it makes the data analysis *much* easier.

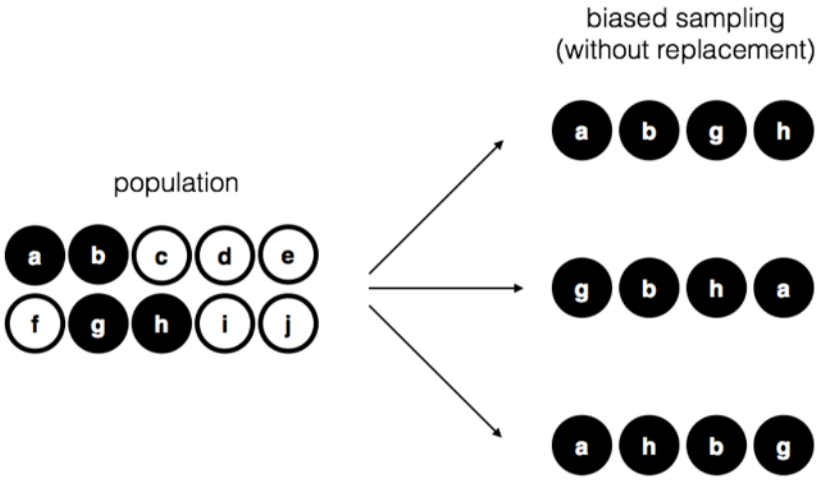


Figure 10.2: Biased sampling without replacement from a finite population

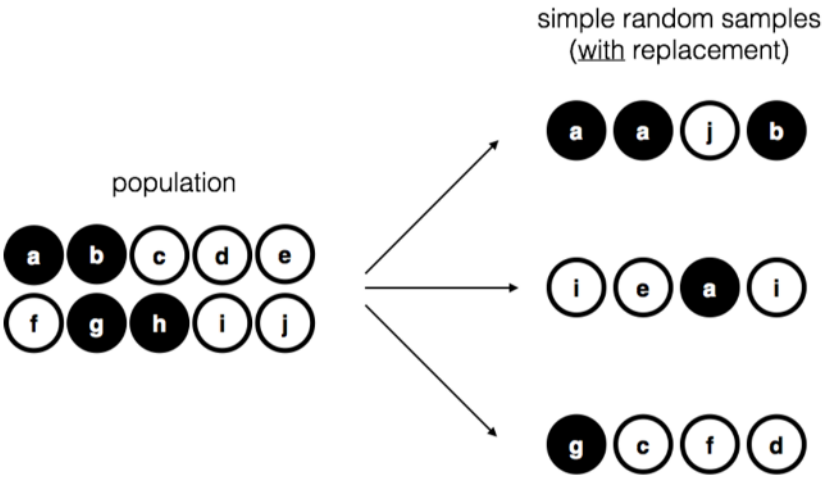


Figure 10.3: Simple random sampling *with* replacement from a finite population

A third procedure is worth mentioning. This time around we close our eyes, shake the bag, and pull out a chip. This time, however, we record the observation and then put the chip back in the bag. Again we close our eyes, shake the bag, and pull out a chip. We then repeat this procedure until we have 4 chips. Data sets generated in this way are still simple random samples, but because we put the chips back in the bag immediately after drawing them it is referred to as a sample ***with replacement***. The difference between this situation and the first one is that it is possible to observe the same population member multiple times, as illustrated in Figure 10.3.

In my experience, most psychology experiments tend to be sampling without replacement, because the same person is not allowed to participate in the experiment twice. However, most statistical theory is based on the assumption that the data arise from a simple random sample *with* replacement. In real life, this very rarely matters. If the population of interest is large (e.g., has more than 10 entities!) the difference between sampling with- and without- replacement is too small to be concerned with. The difference between simple random samples and biased samples, on the other hand, is not such an easy thing to dismiss.

8.1.2 Most samples are not simple random samples

As you can see from looking at the list of possible populations that I showed above, it is almost impossible to obtain a simple random sample from most populations of interest. When I run experiments, I'd consider it a minor miracle if my participants turned out to be a random sampling of the undergraduate psychology students at Adelaide university, even though this is by far the narrowest population that I might want to generalise to. A thorough discussion of other types of sampling schemes is beyond the scope of this book, but to give you a sense of what's out there I'll list a few of the more important ones:

- *Stratified sampling*. Suppose your population is (or can be) divided into several different subpopulations, or *strata*. Perhaps you're running a study at several different sites, for example. Instead of trying to sample randomly from the population as a whole, you instead try to collect a separate random sample from each of the strata. Stratified sampling is sometimes easier to do than simple random sampling, especially when the population is already divided into the distinct strata. It can also be more efficient that simple random sampling, especially when some of the subpopulations are rare. For instance, when studying schizophrenia it would be much better to divide the population into two[148] strata (schizophrenic and not-schizophrenic), and then sample an equal number of people from each group. If you selected people randomly, you would get so few schizophrenic people in the sample that your study would be useless. This specific kind of of stratified sampling is referred to as *oversampling* because it makes a deliberate attempt to over-represent rare groups.
- *Snowball sampling* is a technique that is especially useful when sampling from a "hidden" or hard to access population, and is especially common in social sciences. For instance, suppose the researchers want to conduct an opinion poll among transgender people. The research team might only have contact details for a few trans folks, so the survey starts by asking them to participate (stage 1). At the end of the survey, the participants are asked to provide contact details for other people who might want to participate. In stage 2, those new contacts are surveyed. The process continues until the researchers have sufficient data. The big advantage to snowball sampling is that it gets you data in situations that might otherwise be impossible to get any. On the statistical side, the main disadvantage is that the sample is highly non-random, and non-random in ways that are difficult to address. On the real life side, the disadvantage is that the procedure can be unethical if not handled well, because hidden populations are often hidden for a reason. I chose transgender people as an example here to highlight this: if you weren't careful you might end up outing people who don't want to be outed (very, very bad form), and even if you don't make that mistake it can still be intrusive to use people's social networks to study them. It's certainly very hard to get people's informed consent *before* contacting them, yet in many cases the simple act of contacting them and saying "hey we want to study you" can be hurtful. Social networks are complex things, and just because you can use them to get data doesn't always mean you should.
- *Convenience sampling* is more or less what it sounds like. The samples are chosen in a way that is convenient to the researcher, and not selected at random from the population of interest. Snowball sampling is one type of convenience sampling, but there are many others. A common example in psychology are studies that rely on undergraduate psychology students. These samples are generally non-random in two respects: firstly, reliance on undergraduate psychology students automatically means that your data are restricted to a single subpopulation. Secondly, the students usually get to pick which studies they participate in, so the sample is a self selected subset of psychology students not a randomly selected subset. In real life, most studies are convenience samples of one form or another. This is sometimes a severe limitation, but not always.

8.1.3 much does it matter if you don't have a simple random sample?

Okay, so real world data collection tends not to involve nice simple random samples. Does that matter? A little thought should make it clear to you that it *can* matter if your data are not a simple random sample: just think about the difference between Figures 10.1 and 10.2. However, it's not quite as bad as it sounds. Some types of biased samples are entirely unproblematic. For instance, when using a stratified sampling technique you actually *know* what the bias is because you created it deliberately, often to *increase* the effectiveness of your study, and there are statistical techniques that you can use to adjust for the biases you've introduced (not covered in this book!). So in those situations it's not a problem.

More generally though, it's important to remember that random sampling is a means to an end, not the end in itself. Let's assume you've relied on a convenience sample, and as such you can assume it's biased. A bias in your sampling method is only a problem if it causes you to draw the wrong conclusions. When viewed from that perspective, I'd argue that we don't need the sample to be randomly generated in *every* respect: we only need it to be random with respect to the psychologically-relevant phenomenon of interest. Suppose I'm doing a study looking at working memory capacity. In study 1, I actually have the ability to sample randomly from all human beings currently alive, with one exception: I can only sample people born on a Monday. In study 2, I am able to sample randomly from the Australian population. I want to generalise my results to the population of all living humans. Which study is better? The answer, obviously, is study 1. Why? Because we have no reason to think that being "born on a Monday" has any interesting relationship to working memory capacity. In contrast, I can think of several reasons why "being Australian" might matter. Australia is a wealthy, industrialised country with a very well-developed education system. People growing up in that system will have had life experiences much more similar to the experiences of the people who designed the tests for working memory capacity. This shared experience might easily translate into similar beliefs about how to "take a test", a shared assumption

about how psychological experimentation works, and so on. These things might actually matter. For instance, "test taking" style might have taught the Australian participants how to direct their attention exclusively on fairly abstract test materials relative to people that haven't grown up in a similar environment; leading to a misleading picture of what working memory capacity is.

There are two points hidden in this discussion. Firstly, when designing your own studies, it's important to think about what population you care about, and try hard to sample in a way that is appropriate to that population. In practice, you're usually forced to put up with a "sample of convenience" (e.g., psychology lecturers sample psychology students because that's the least expensive way to collect data, and our coffers aren't exactly overflowing with gold), but if so you should at least spend some time thinking about what the dangers of this practice might be.

Secondly, if you're going to criticise someone else's study because they've used a sample of convenience rather than laboriously sampling randomly from the entire human population, at least have the courtesy to offer a specific theory as to *how* this might have distorted the results. Remember, everyone in science is aware of this issue, and does what they can to alleviate it. Merely pointing out that "the study only included people from group BLAH" is entirely unhelpful, and borders on being insulting to the researchers, who are *of course* aware of the issue. They just don't happen to be in possession of the infinite supply of time and money required to construct the perfect sample. In short, if you want to offer a responsible critique of the sampling process, then be *helpful*. Rehashing the blindingly obvious truisms that I've been rambling on about in this section isn't helpful.

8.1.4 Population parameters and sample statistics

Okay. Setting aside the thorny methodological issues associated with obtaining a random sample and my rather unfortunate tendency to rant about lazy methodological criticism, let's consider a slightly different issue. Up to this point we have been talking about populations the way a scientist might. To a psychologist, a population might be a group of people. To an ecologist, a population might be a group of bears. In most cases the populations that scientists care about are concrete things that actually exist in the real world. Statisticians, however, are a funny lot. On the one hand, they *are* interested in real world data and real science in the same way that scientists are. On the other hand, they also operate in the realm of pure abstraction in the way that mathematicians do. As a consequence, statistical theory tends to be a bit abstract in how a population is defined. In much the same way that psychological researchers operationalise our abstract theoretical ideas in terms of concrete measurements (Section 2.1, statisticians operationalise the concept of a "population" in terms of mathematical objects that they know how to work with. You've already come across these objects in Chapter 9: they're called probability distributions.

The idea is quite simple. Let's say we're talking about IQ scores. To a psychologist, the population of interest is a group of actual humans who have IQ scores. A statistician "simplifies" this by operationally defining the population as the probability distribution depicted in Figure **??**. IQ tests are designed so that the average IQ is 100, the standard deviation of IQ scores is 15, and the distribution of IQ scores is normal. These values are referred to as the ***population parameters*** because they are characteristics of the entire population. That is, we say that the population mean μ is 100, and the population standard deviation σ is 15.

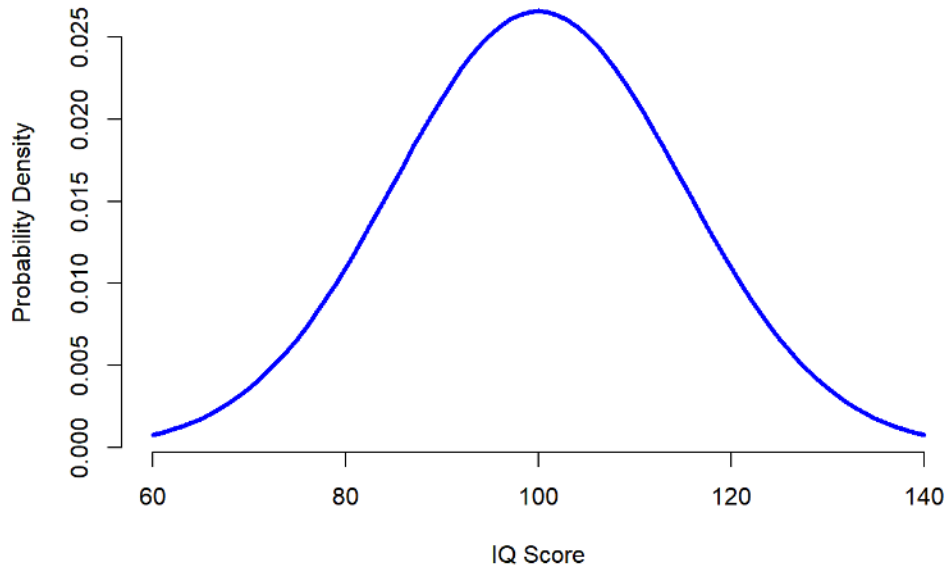


Figure 10.4: The population distribution of IQ scores (panel a) and two samples drawn randomly from it. In panel b we have a sample of 100 observations, and panel c we have a sample of 10,000 observations.

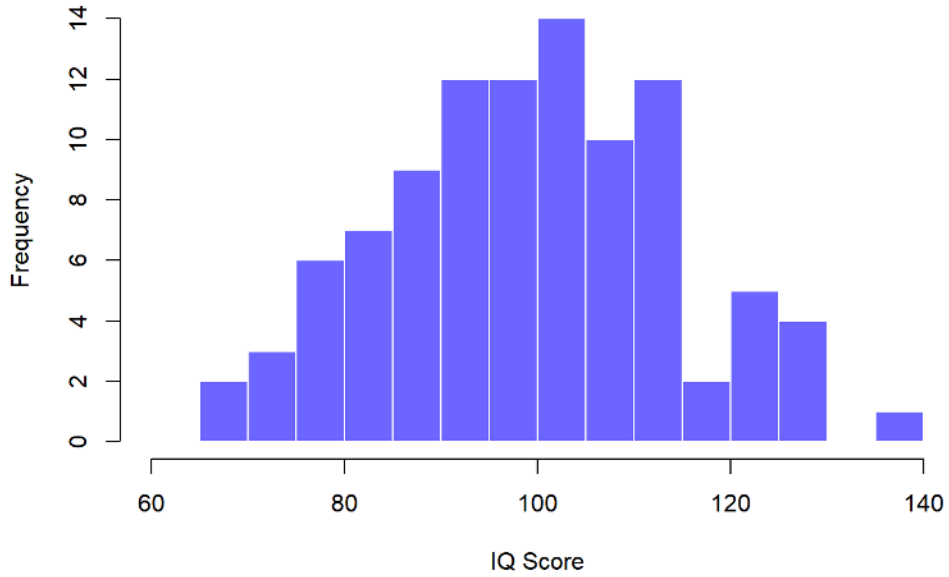


Figure 10.4: The population distribution of IQ scores (panel a) and two samples drawn randomly from it. In panel b we have a sample of 100 observations, and panel c we have a sample of 10,000 observations.

```
## [1] "n= 100 mean= 99.6064025956605 sd= 16.0047604703873"
```

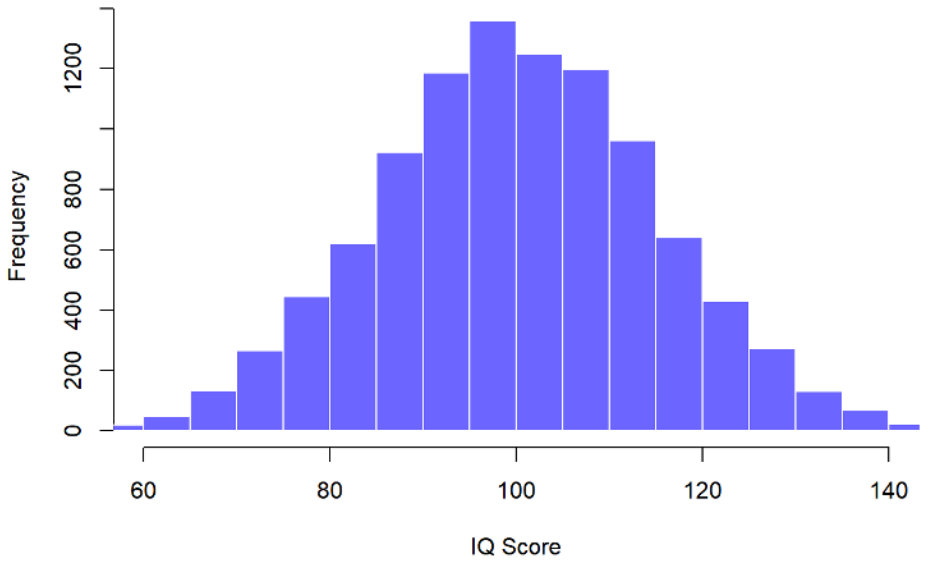


Figure 10.4: The population distribution of IQ scores (panel a) and two samples drawn randomly from it. In panel b we have a sample of 100 observations, and panel c we have a sample of 10,000 observations.

```
## [1] "n= 10000 mean= 100.096924966188 sd= 14.9554812898374"
```

Now suppose I run an experiment. I select 100 people at random and administer an IQ test, giving me a simple random sample from the population. My sample would consist of a collection of numbers like this:

```
  106 101 98 80 74 ... 107 72 100
```

Each of these IQ scores is sampled from a normal distribution with mean 100 and standard deviation 15. So if I plot a histogram of the sample, I get something like the one shown in Figure 10.4b. As you can see, the histogram is *roughly* the right shape, but it's a very crude approximation to the true population distribution shown in Figure 10.4a. When I calculate the mean of my sample, I get a number that is fairly close to the population mean 100 but not identical. In this case, it turns out that the people in my sample have a mean IQ of 98.5, and the standard deviation of their IQ scores is 15.9. These ***sample statistics*** are properties of my data set, and although they are fairly similar to the true population values, they are not the same. In general, sample statistics are the things you can calculate from your data set, and the population parameters are the things you want to learn about. Later on in this chapter I'll talk about how you can estimate population parameters using your sample statistics (Section 10.4 and how to work out how confident you are in your estimates (Section 10.5 but before we get to that there's a few more ideas in sampling theory that you need to know about.

This page titled 8.1: Samples, Populations and Sampling is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **10.1: Samples, Populations and Sampling** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

8.2: The Law of Large Numbers

In the previous section I showed you the results of one fictitious IQ experiment with a sample size of N=100. The results were somewhat encouraging: the true population mean is 100, and the sample mean of 98.5 is a pretty reasonable approximation to it. In many scientific studies that level of precision is perfectly acceptable, but in other situations you need to be a lot more precise. If we want our sample statistics to be much closer to the population parameters, what can we do about it?

The obvious answer is to collect more data. Suppose that we ran a much larger experiment, this time measuring the IQs of 10,000 people. We can simulate the results of this experiment using R. In Section 9.5 I introduced the `rnorm()` function, which generates random numbers sampled from a normal distribution. For an experiment with a sample size of `n = 10000`, and a population with `mean = 100` and `sd = 15`, R produces our fake IQ data using these commands:

```
IQ <- rnorm(n = 10000, mean = 100, sd = 15) # generate IQ scores
IQ <- round(IQ) # IQs are whole numbers!
print(head(IQ))
```

```
## [1]  82  91 123 129 104  96
```

I can compute the mean IQ using the command `mean(IQ)` and the standard deviation using the command `sd(IQ)`, and I can draw a histgram using `hist()`. The histogram of this much larger sample is shown in Figure 10.4c. Even a moment's inspections makes clear that the larger sample is a much better approximation to the true population distribution than the smaller one. This is reflected in the sample statistics: the mean IQ for the larger sample turns out to be 99.9, and the standard deviation is 15.1. These values are now very close to the true population.

I feel a bit silly saying this, but the thing I want you to take away from this is that large samples generally give you better information. I feel silly saying it because it's so bloody obvious that it shouldn't need to be said. In fact, it's such an obvious point that when Jacob Bernoulli – one of the founders of probability theory – formalised this idea back in 1713, he was kind of a jerk about it. Here's how he described the fact that we all share this intuition:

For even the most stupid of men, by some instinct of nature, by himself and without any instruction (which is a remarkable thing), is convinced that the more observations have been made, the less danger there is of wandering from one's goal Stigler (1986)

Okay, so the passage comes across as a bit condescending (not to mention sexist), but his main point is correct: it really does feel obvious that more data will give you better answers. The question is, why is this so? Not surprisingly, this intuition that we all share turns out to be correct, and statisticians refer to it as the ***law of large numbers***. The law of large numbers is a mathematical law that applies to many different sample statistics, but the simplest way to think about it is as a law about averages. The sample mean is the most obvious example of a statistic that relies on averaging (because that's what the mean is… an average), so let's look at that. When applied to the sample mean, what the law of large numbers states is that as the sample gets larger, the sample mean tends to get closer to the true population mean. Or, to say it a little bit more precisely, as the sample size "approaches" infinity (written as $N \rightarrow \infty$) the sample mean approaches the population mean ($\bar{X} \rightarrow \mu$).[149]

I don't intend to subject you to a proof that the law of large numbers is true, but it's one of the most important tools for statistical theory. The law of large numbers is the thing we can use to justify our belief that collecting more and more data will eventually lead us to the truth. For any particular data set, the sample statistics that we calculate from it will be wrong, but the law of large numbers tells us that if we keep collecting more data those sample statistics will tend to get closer and closer to the true population parameters.

This page titled 8.2: The Law of Large Numbers is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **10.2: The Law of Large Numbers** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

8.3: Sampling Distributions and the Central Limit Theorem

The law of large numbers is a very powerful tool, but it's not going to be good enough to answer all our questions. Among other things, all it gives us is a "long run guarantee". In the long run, if we were somehow able to collect an infinite amount of data, then the law of large numbers guarantees that our sample statistics will be correct. But as John Maynard Keynes famously argued in economics, a long run guarantee is of little use in real life:

[The] long run is a misleading guide to current affairs. In the long run we are all dead. Economists set themselves too easy, too useless a task, if in tempestuous seasons they can only tell us, that when the storm is long past, the ocean is flat again. Keynes (1923)

As in economics, so too in psychology and statistics. It is not enough to know that we will *eventually* arrive at the right answer when calculating the sample mean. Knowing that an infinitely large data set will tell me the exact value of the population mean is cold comfort when my *actual* data set has a sample size of N=100. In real life, then, we must know something about the behaviour of the sample mean when it is calculated from a more modest data set!

8.3.1 Sampling distribution of the mean

With this in mind, let's abandon the idea that our studies will have sample sizes of 10000, and consider a very modest experiment indeed. This time around we'll sample N=5 people and measure their IQ scores. As before, I can simulate this experiment in R using the `rnorm()` function:

```
> IQ.1 <- round( rnorm(n=5, mean=100, sd=15 ))
> IQ.1
[1]  90  82  94  99 110
```

The mean IQ in this sample turns out to be exactly 95. Not surprisingly, this is much less accurate than the previous experiment. Now imagine that I decided to ***replicate*** the experiment. That is, I repeat the procedure as closely as possible: I randomly sample 5 new people and measure their IQ. Again, R allows me to simulate the results of this procedure:

```
> IQ.2 <- round( rnorm(n=5, mean=100, sd=15 ))
> IQ.2
[1]  78  88 111 111 117
```

This time around, the mean IQ in my sample is 101. If I repeat the experiment 10 times I obtain the results shown in Table ??, and as you can see the sample mean varies from one replication to the next.

NANA	Person.1	Person.2	Person.3	Person.4	Person.5	Sample.Mean	caption
Replication 1	90	82	94	99	110	95.0	Ten replications of the IQ experiment, each with a sample size of N=5.
Replication 2	78	88	111	111	117	101.0	Ten replications of the IQ experiment, each with a sample size of N=5.
Replication 3	111	122	91	98	86	101.6	Ten replications of the IQ experiment, each with a sample size of N=5.
Replication 4	98	96	119	99	107	103.8	Ten replications of the IQ experiment, each with a sample size of N=5.
Replication 5	105	113	103	103	98	104.4	Ten replications of the IQ experiment, each with a sample size of N=5.
Replication 6	81	89	93	85	114	92.4	Ten replications of the IQ experiment, each with a sample size of N=5.
Replication 7	100	93	108	98	133	106.4	Ten replications of the IQ experiment, each with a

NANA	Person.1	Person.2	Person.3	Person.4	Person.5	Sample.Mean	caption
							sample size of N=5.
Replication 8	107	100	105	117	85	102.8	Ten replications of the IQ experiment, each with a sample size of N=5.
Replication 9	86	119	108	73	116	100.4	Ten replications of the IQ experiment, each with a sample size of N=5.
Replication 10	95	126	112	120	76	105.8	Ten replications of the IQ experiment, each with a sample size of N=

Now suppose that I decided to keep going in this fashion, replicating this "five IQ scores" experiment over and over again. Every time I replicate the experiment I write down the sample mean. Over time, I'd be amassing a new data set, in which every experiment generates a single data point. The first 10 observations from my data set are the sample means listed in Table **??**, so my data set starts out like this:

```
95.0 101.0 101.6 103.8 104.4 ...
```

What if I continued like this for 10,000 replications, and then drew a histogram? Using the magical powers of R that's exactly what I did, and you can see the results in Figure 10.5. As this picture illustrates, the average of 5 IQ scores is usually between 90 and 110. But more importantly, what it highlights is that if we replicate an experiment over and over again, what we end up with is a *distribution* of sample means! This distribution has a special name in statistics: it's called the ***sampling distribution of the mean***.

Sampling distributions are another important theoretical idea in statistics, and they're crucial for understanding the behaviour of small samples. For instance, when I ran the very first "five IQ scores" experiment, the sample mean turned out to be 95. What the sampling distribution in Figure 10.5 tells us, though, is that the "five IQ scores" experiment is not very accurate. If I repeat the experiment, the sampling distribution tells me that I can expect to see a sample mean anywhere between 80 and 120.

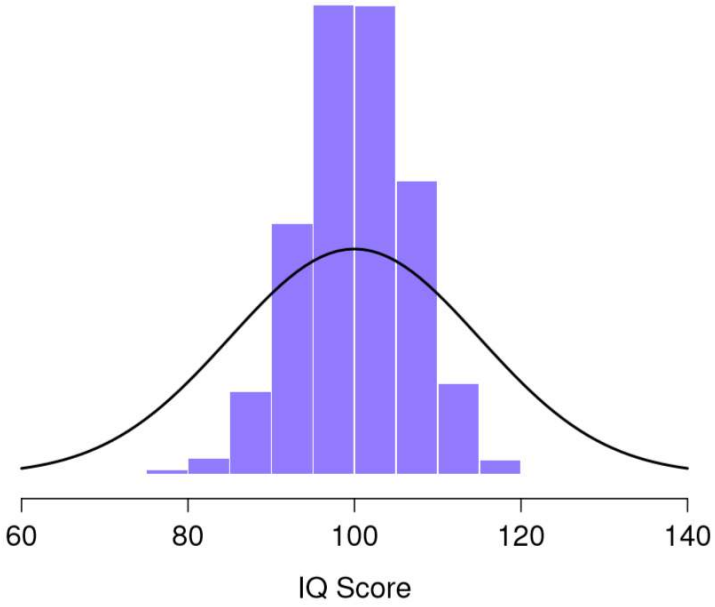


Figure 10.5: The sampling distribution of the mean for the "five IQ scores experiment". If you sample 5 people at random and calculate their *average* IQ, you'll almost certainly get a number between 80 and 120, even though there are quite a lot of individuals who have IQs above 120 or below 80. For comparison, the black line plots the population distribution of IQ scores.

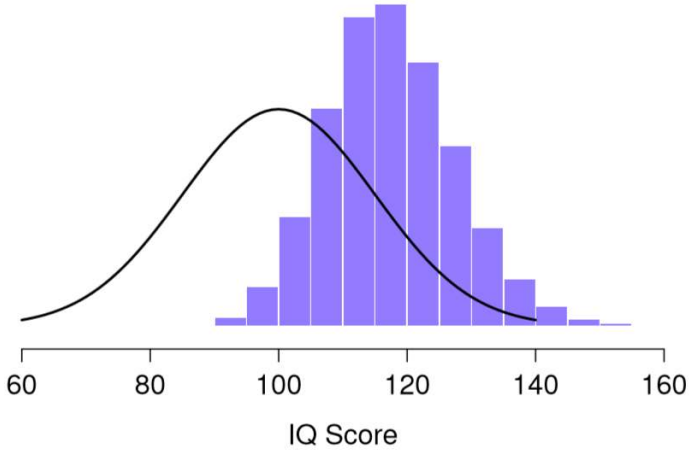


Figure 10.6: The sampling distribution of the *maximum* for the "five IQ scores experiment". If you sample 5 people at random and select the one with the highest IQ score, you'll probably see someone with an IQ between 100 and 140.

8.3.2 Sampling distributions exist for any sample statistic!

One thing to keep in mind when thinking about sampling distributions is that *any* sample statistic you might care to calculate has a sampling distribution. For example, suppose that each time I replicated the "five IQ scores" experiment I wrote down the largest IQ score in the experiment. This would give me a data set that started out like this:

```
110 117 122 119 113 ...
```

Doing this over and over again would give me a very different sampling distribution, namely the *sampling distribution of the maximum*. The sampling distribution of the maximum of 5 IQ scores is shown in Figure 10.6. Not surprisingly, if you pick 5 people at random and then find the person with the highest IQ score, they're going to have an above average IQ. Most of the time you'll end up with someone whose IQ is measured in the 100 to 140 range.

8.3.3 central limit theorem

An illustration of the how sampling distribution of the mean depends on sample size. In each panel, I generated 10,000 samples of IQ data, and calculated the mean IQ observed within each of these data sets. The histograms in these plots show the distribution of these means (i.e., the sampling distribution of the mean). Each individual IQ score was drawn from a normal distribution with mean 100 and standard deviation 15, which is shown as the solid black line).

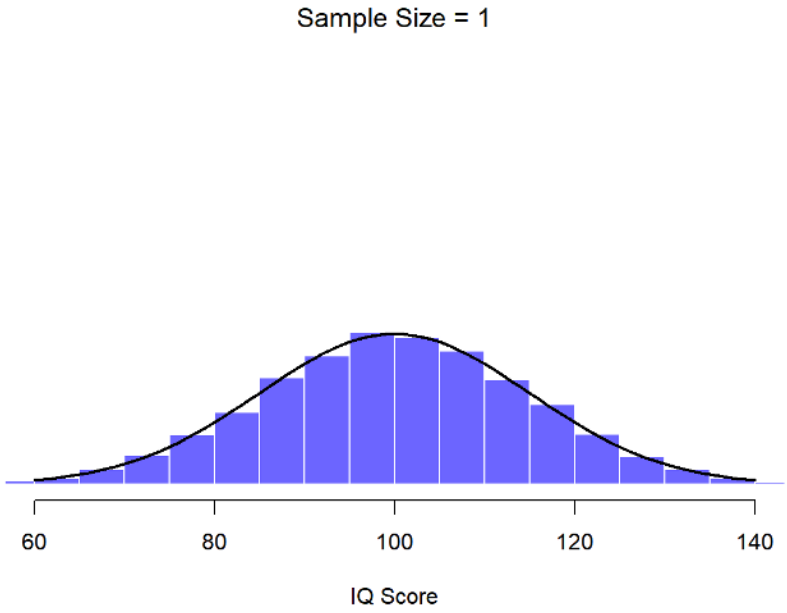


Figure 10.7: Each data set contained only a single observation, so the mean of each sample is just one person's IQ score. As a consequence, the sampling distribution of the mean is of course identical to the population distribution of IQ scores.

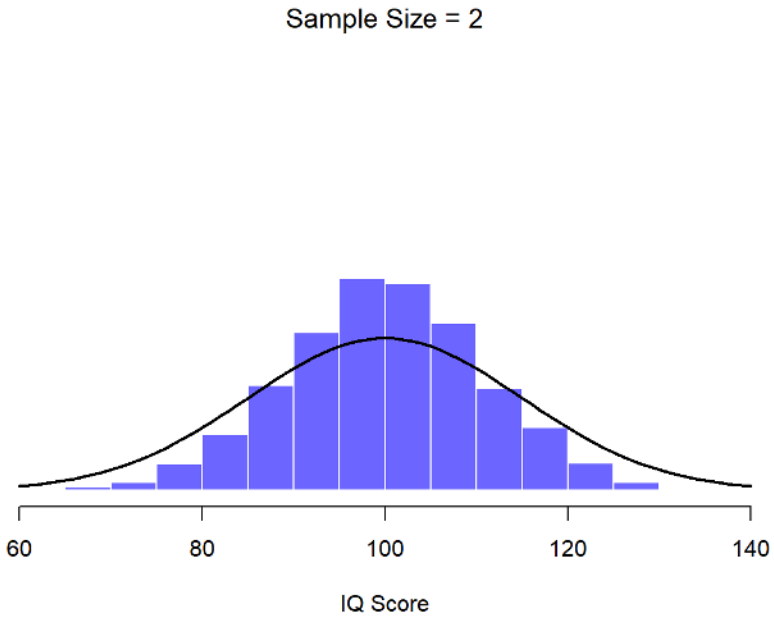


Figure 10.8: When we raise the sample size to 2, the mean of any one sample tends to be closer to the population mean than a one person's IQ score, and so the histogram (i.e., the sampling distribution) is a bit narrower than the population distribution.

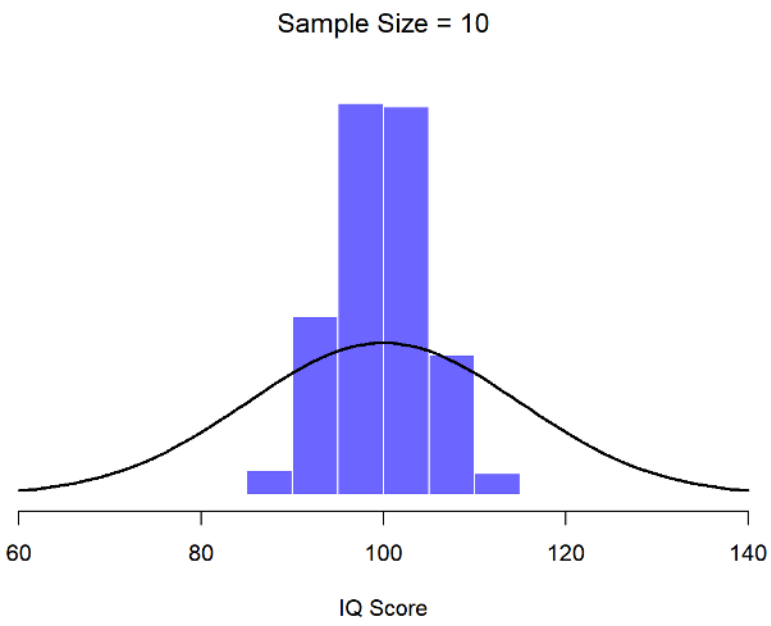


Figure 10.9: By the time we raise the sample size to 10, we can see that the distribution of sample means tend to be fairly tightly clustered around the true population mean.

At this point I hope you have a pretty good sense of what sampling distributions are, and in particular what the sampling distribution of the mean is. In this section I want to talk about how the sampling distribution of the mean changes as a function of sample size. Intuitively, you already know part of the answer: if you only have a few observations, the sample mean is likely to be quite inaccurate: if you replicate a small experiment and recalculate the mean you'll get a very different answer. In other words, the sampling distribution is quite wide. If you replicate a large experiment and recalculate the sample mean you'll probably get the same answer you got last time, so the sampling distribution will be very narrow. You can see this visually in Figures 10.7, 10.8 and 10.9: the bigger the sample size, the narrower the sampling distribution gets. We can quantify this effect by calculating the standard deviation of the sampling distribution, which is referred to as the ***standard error***. The standard error of a statistic is often denoted SE, and since we're usually interested in the standard error of the sample *mean*, we often use the acronym SEM. As you can see just by looking at the picture, as the sample size N increases, the SEM decreases.

Okay, so that's one part of the story. However, there's something I've been glossing over so far. All my examples up to this point have been based on the "IQ scores" experiments, and because IQ scores are roughly normally distributed, I've assumed that the population distribution is normal. What if it isn't normal? What happens to the sampling distribution of the mean? The remarkable thing is this: no matter what shape your population distribution is, as N increases the sampling distribution of the mean starts to look more like a normal distribution. To give you a sense of this, I ran some simulations using R. To do this, I started with the "ramped" distribution shown in the histogram in Figure 10.10. As you can see by comparing the triangular shaped histogram to the bell curve plotted by the black line, the population distribution doesn't look very much like a normal distribution at all. Next, I used R to simulate the results of a large number of experiments. In each experiment I took N=2 samples from this distribution, and then calculated the sample mean. Figure **??** plots the histogram of these sample means (i.e., the sampling distribution of the mean for N=2). This time, the histogram produces a ∩-shaped distribution: it's still not normal, but it's a lot closer to the black line than the population distribution in Figure **??**. When I increase the sample size to N=4, the sampling distribution of the mean is very close to normal (Figure **??**, and by the time we reach a sample size of N=8 it's almost perfectly normal. In other words, as long as your sample size isn't tiny, the sampling distribution of the mean will be approximately normal no matter what your population distribution looks like!

```
# needed for printing
   width <- 6
   height <- 6

   # parameters of the beta
   a <- 2
   b <- 1

   # mean and standard deviation of the beta
   s <- sqrt( a*b / (a+b)^2 / (a+b+1) )
   m <- a / (a+b)

   # define function to draw a plot
   plotOne <- function(n,N=50000) {

       # generate N random sample means of size n
       X <- matrix(rbeta(n*N,a,b),n,N)
       X <- colMeans(X)

       # plot the data
       hist( X, breaks=seq(0,1,.025), border="white", freq=FALSE,
           col=ifelse(colour,emphColLight,emphGrey),
           xlab="Sample Mean", ylab="", xlim=c(0,1.2),
           main=paste("Sample Size =",n), axes=FALSE,
           font.main=1, ylim=c(0,5)
       )
       box()
       axis(1)
       #axis(2)

       # plot the theoretical distribution
       lines( x <- seq(0,1.2,.01), dnorm(x,m,s/sqrt(n)),
           lwd=2, col="black", type="l"
       )
   }

   for( i in c(1,2,4,8)) {
       plotOne(i)}
```

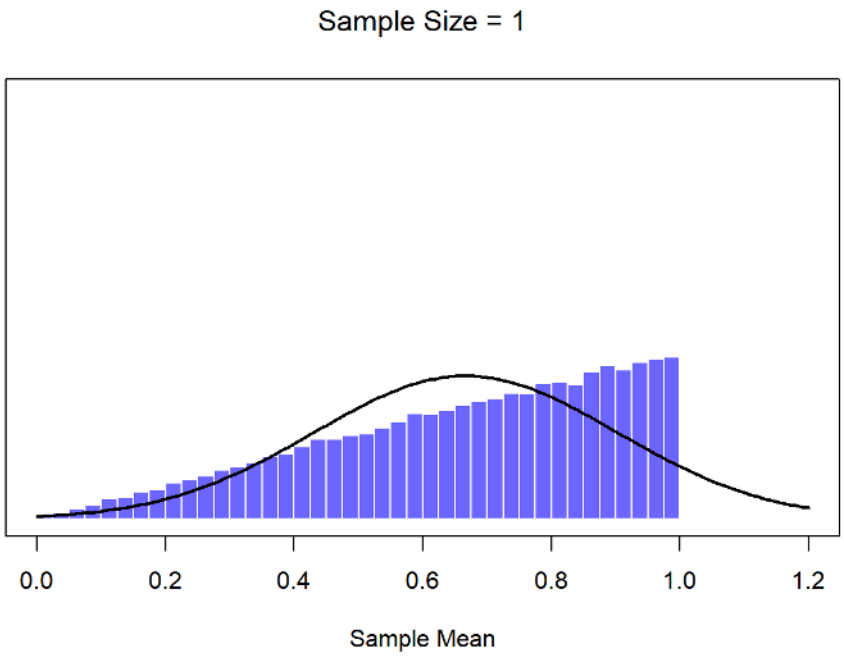


Figure 10.10: A demonstration of the central limit theorem. In panel a, we have a non-normal population distribution; and panels b-d show the sampling distribution of the mean for samples of size 2,4 and 8, for data drawn from the distribution in panel a. As you can see, even though the original population distribution is non-normal, the sampling distribution of the mean becomes pretty close to normal by the time you have a sample of even 4 observations.

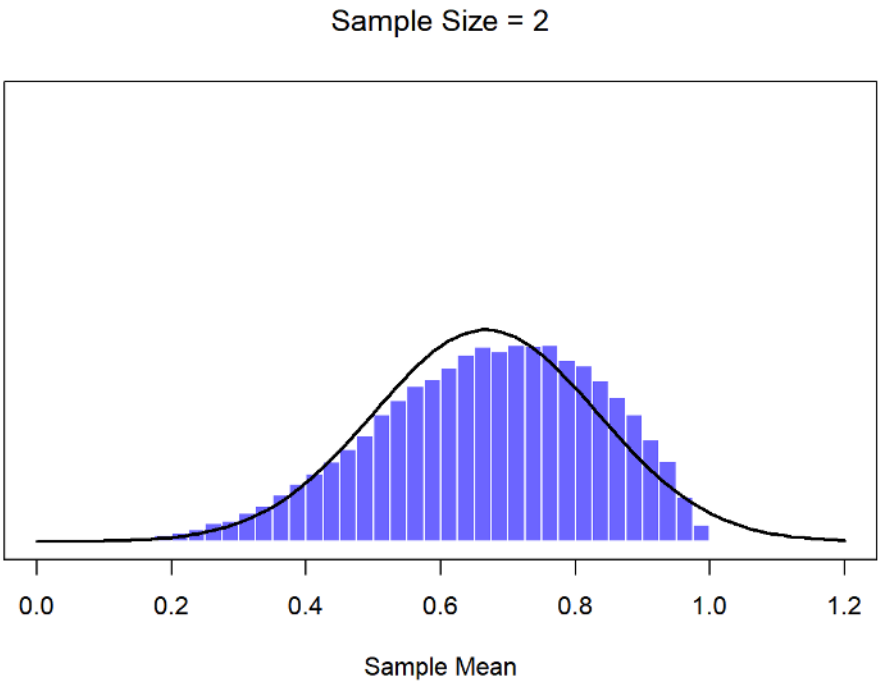


Figure 10.10: A demonstration of the central limit theorem. In panel a, we have a non-normal population distribution; and panels b-d show the sampling distribution of the mean for samples of size 2,4 and 8, for data drawn from the distribution in panel a. As you can see, even though the original population distribution is non-normal, the sampling distribution of the mean becomes pretty close to normal by the time you have a sample of even 4 observations.

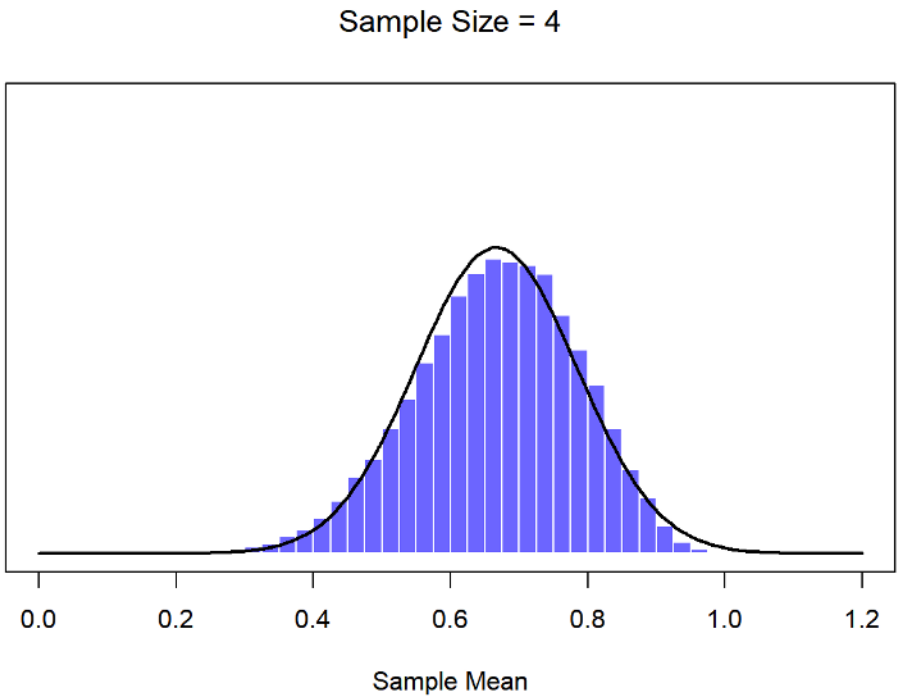


Figure 10.10: A demonstration of the central limit theorem. In panel a, we have a non-normal population distribution; and panels b-d show the sampling distribution of the mean for samples of size 2,4 and 8, for data drawn from the distribution in panel a. As you can see, even though the original population distribution is non-normal, the sampling distribution of the mean becomes pretty close to normal by the time you have a sample of even 4 observations.

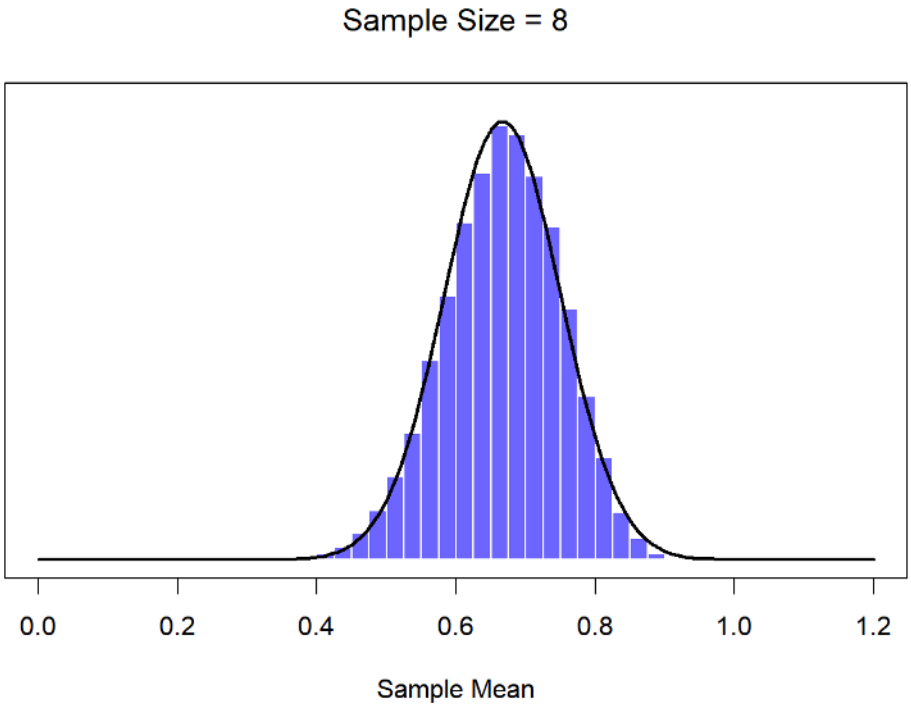


Figure 10.10: A demonstration of the central limit theorem. In panel a, we have a non-normal population distribution; and panels b-d show the sampling distribution of the mean for samples of size 2,4 and 8, for data drawn from the distribution in panel a. As you can see, even though the original population distribution is non-normal, the sampling distribution of the mean becomes pretty close to normal by the time you have a sample of even 4 observations.

On the basis of these figures, it seems like we have evidence for all of the following claims about the sampling distribution of the mean:

- The mean of the sampling distribution is the same as the mean of the population

- The standard deviation of the sampling distribution (i.e., the standard error) gets smaller as the sample size increases
- The shape of the sampling distribution becomes normal as the sample size increases

As it happens, not only are all of these statements true, there is a very famous theorem in statistics that proves all three of them, known as the ***central limit theorem***. Among other things, the central limit theorem tells us that if the population distribution has mean μ and standard deviation σ, then the sampling distribution of the mean also has mean μ, and the standard error of the mean is

$$\text{SEM} = \frac{\sigma}{\sqrt{N}}$$

Because we divide the population standard devation σ by the square root of the sample size N, the SEM gets smaller as the sample size increases. It also tells us that the shape of the sampling distribution becomes normal.[150]

This result is useful for all sorts of things. It tells us why large experiments are more reliable than small ones, and because it gives us an explicit formula for the standard error it tells us *how much* more reliable a large experiment is. It tells us why the normal distribution is, well, *normal*. In real experiments, many of the things that we want to measure are actually averages of lots of different quantities (e.g., arguably, "general" intelligence as measured by IQ is an average of a large number of "specific" skills and abilities), and when that happens, the averaged quantity should follow a normal distribution. Because of this mathematical law, the normal distribution pops up over and over again in real data.

This page titled 8.3: Sampling Distributions and the Central Limit Theorem is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **10.3: Sampling Distributions and the Central Limit Theorem** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

8.4: Estimating Population Parameters

In all the IQ examples in the previous sections, we actually knew the population parameters ahead of time. As every undergraduate gets taught in their very first lecture on the measurement of intelligence, IQ scores are *defined* to have mean 100 and standard deviation 15. However, this is a bit of a lie. How do we know that IQ scores have a true population mean of 100? Well, we know this because the people who designed the tests have administered them to very large samples, and have then "rigged" the scoring rules so that their sample has mean 100. That's not a bad thing of course: it's an important part of designing a psychological measurement. However, it's important to keep in mind that this theoretical mean of 100 only attaches to the population that the test designers used to design the tests. Good test designers will actually go to some lengths to provide "test norms" that can apply to lots of different populations (e.g., different age groups, nationalities etc).

This is very handy, but of course almost every research project of interest involves looking at a different population of people to those used in the test norms. For instance, suppose you wanted to measure the effect of low level lead poisoning on cognitive functioning in Port Pirie, a South Australian industrial town with a lead smelter. Perhaps you decide that you want to compare IQ scores among people in Port Pirie to a comparable sample in Whyalla, a South Australian industrial town with a steel refinery.[151] Regardless of which town you're thinking about, it doesn't make a lot of sense simply to *assume* that the true population mean IQ is 100. No-one has, to my knowledge, produced sensible norming data that can automatically be applied to South Australian industrial towns. We're going to have to ***estimate*** the population parameters from a sample of data. So how do we do this?

8.4.1 Estimating the population mean

Suppose we go to Port Pirie and 100 of the locals are kind enough to sit through an IQ test. The average IQ score among these people turns out to be $\bar{X}$ =98.5. So what is the true mean IQ for the entire population of Port Pirie? Obviously, we don't know the answer to that question. It could be 97.2, but if could also be 103.5. Our sampling isn't exhaustive so we cannot give a definitive answer. Nevertheless if I was forced at gunpoint to give a "best guess" I'd have to say 98.5. That's the essence of statistical estimation: giving a best guess.

In this example, estimating the unknown poulation parameter is straightforward. I calculate the sample mean, and I use that as my ***estimate of the population mean***. It's pretty simple, and in the next section I'll explain the statistical justification for this intuitive answer. However, for the moment what I want to do is make sure you recognise that the sample statistic and the estimate of the population parameter are conceptually different things. A sample statistic is a description of your data, whereas the estimate is a guess about the population. With that in mind, statisticians often different notation to refer to them. For instance, if true population mean is denoted μ, then we would use $\hat{\mu}$ to refer to our estimate of the population mean. In contrast, the sample mean is denoted $\bar{X}$ or sometimes m. However, in simple random samples, the estimate of the population mean is identical to the sample mean: if I observe a sample mean of $\bar{X}$ =98.5, then my estimate of the population mean is also $\hat{\mu}$=98.5. To help keep the notation clear, here's a handy table:

```
knitr::kable(data.frame(stringsAsFactors=FALSE,
                  Symbol = c("$\\bar{X}$", "$\\mu$", "$\\hat{\\mu}$"),
              What.is.it = c("Sample mean", "True population mean",
                             "Estimate of the population mean"),
   Do.we.know.what.it.is = c("Yes  calculated from the raw data",
                             "Almost never known for sure",
                             "Yes  identical to the sample mean")))
```

Symbol	What.is.it	Do.we.know.what.it.is
$\bar{X}$	Sample mean	Yes calculated from the raw data
μ	True population mean	Almost never known for sure
$\hat{\mu}$	Estimate of the population mean	Yes identical to the sample mean

8.4.2 Estimating the population standard deviation

So far, estimation seems pretty simple, and you might be wondering why I forced you to read through all that stuff about sampling theory. In the case of the mean, our estimate of the population parameter (i.e. $\hat{\mu}$) turned out to identical to the corresponding sample statistic (i.e. $\bar{X}$). However, that's not always true. To see this, let's have a think about how to construct an ***estimate of the population standard deviation***, which we'll denote $\hat{\sigma}$. What shall we use as our estimate in this case? Your first thought might be that we could do the same thing we did when estimating the mean, and just use the sample statistic as our estimate. That's almost the right thing to do, but not quite.

Here's why. Suppose I have a sample that contains a single observation. For this example, it helps to consider a sample where you have no intutions at all about what the true population values might be, so let's use something completely fictitious. Suppose the observation in question measures the *cromulence* of my shoes. It turns out that my shoes have a cromulence of 20. So here's my sample:

```
20
```

This is a perfectly legitimate sample, even if it does have a sample size of N=1. It has a sample mean of 20, and because every observation in this sample is equal to the sample mean (obviously!) it has a sample standard deviation of 0. As a description of the *sample* this seems quite right: the sample contains a single observation and therefore there is no variation observed within the sample. A sample standard deviation of s=0 is the right answer here. But as an estimate of the *population* standard deviation, it feels completely insane, right? Admittedly, you and I don't know anything at all about what "cromulence" is, but we know something about data: the only reason that we don't see any variability in the *sample* is that the sample is too small to display any variation! So, if you have a sample size of N=1, it *feels* like the right answer is just to say "no idea at all".

Notice that you *don't* have the same intuition when it comes to the sample mean and the population mean. If forced to make a best guess about the population mean, it doesn't feel completely insane to guess that the population mean is 20. Sure, you probably wouldn't feel very confident in that guess, because you have only the one observation to work with, but it's still the best guess you can make.

Let's extend this example a little. Suppose I now make a second observation. My data set now has N=2 observations of the cromulence of shoes, and the complete sample now looks like this:

```
20, 22
```

This time around, our sample is *just* large enough for us to be able to observe some variability: two observations is the bare minimum number needed for any variability to be observed! For our new data set, the sample mean is $\bar{X}$ =21, and the sample standard deviation is s=1. What intuitions do we have about the population? Again, as far as the population mean goes, the best guess we can possibly make is the sample mean: if forced to guess, we'd probably guess that the population mean cromulence is 21. What about the standard deviation? This is a little more complicated. The sample standard deviation is only based on two observations, and if you're at all like me you probably have the intuition that, with only two observations, we haven't given the population "enough of a chance" to reveal its true variability to us. It's not just that we suspect that the estimate is *wrong*: after all, with only two observations we expect it to be wrong to some degree. The worry is that the error is *systematic*. Specifically, we suspect that the sample standard deviation is likely to be smaller than the population standard deviation.

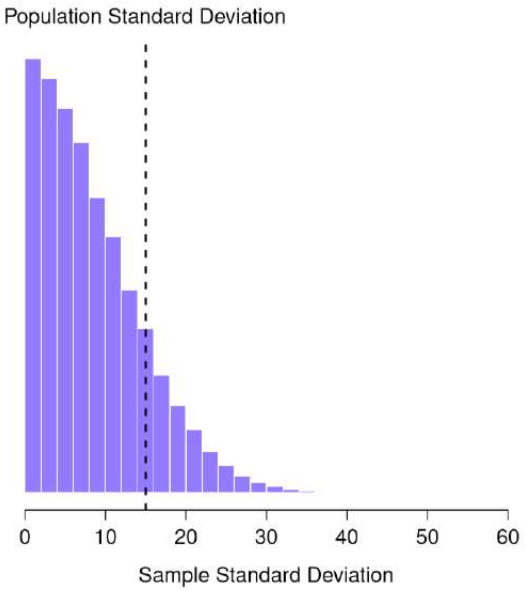


Figure 10.11: The sampling distribution of the sample standard deviation for a "two IQ scores" experiment. The true population standard deviation is 15 (dashed line), but as you can see from the histogram, the vast majority of experiments will produce a much smaller sample standard deviation than this. On average, this experiment would produce a sample standard deviation of only 8.5, well below the true value! In other words, the sample standard deviation is a *biased* estimate of the population standard deviation.

This intuition feels right, but it would be nice to demonstrate this somehow. There are in fact mathematical proofs that confirm this intuition, but unless you have the right mathematical background they don't help very much. Instead, what I'll do is use R to simulate the results of some experiments. With that in mind, let's return to our IQ studies. Suppose the true population mean IQ is 100 and the standard deviation is 15. I can use the `rnorm()` function to generate the the results of an experiment in which I measure N=2 IQ scores, and calculate the sample standard deviation. If I do this over and over again, and plot a histogram of these sample standard deviations, what I have is the *sampling distribution of the standard deviation*. I've plotted this distribution in Figure 10.11. Even though the true population standard deviation is 15, the average of the *sample* standard deviations is only 8.5. Notice that this is a very different result to what we found in Figure 10.8 when we plotted the sampling distribution of the mean. If you look at that sampling distribution, what you see is that the population mean is 100, and the average of the sample means is also 100.

Now let's extend the simulation. Instead of restricting ourselves to the situation where we have a sample size of N=2, let's repeat the exercise for sample sizes from 1 to 10. If we plot the average sample mean and average sample standard deviation as a function of sample size, you get the results shown in Figure 10.12. On the left hand side (panel a), I've plotted the average sample mean and on the right hand side (panel b), I've plotted the average standard deviation. The two plots are quite different: *on average*, the average sample mean is equal to the population mean. It is an ***unbiased estimator***, which is essentially the reason why your best estimate for the population mean is the sample mean.[152] The plot on the right is quite different: on average, the sample standard deviation s is *smaller* than the population standard deviation σ. It is a ***biased estimator***. In other words, if we want to make a "best guess" $\hat{\sigma}$ about the value of the population standard deviation σ, we should make sure our guess is a little bit larger than the sample standard deviation s.

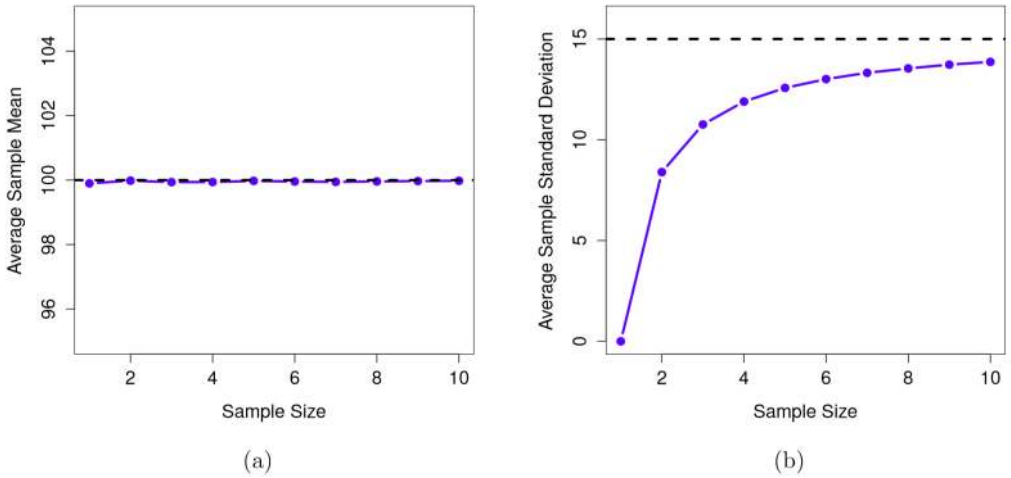


Figure 10.12: An illustration of the fact that the sample mean is an unbiased estimator of the population mean (panel a), but the sample standard deviation is a biased estimator of the population standard deviation (panel b). To generate the figure, I generated 10,000 simulated data sets with 1 observation each, 10,000 more with 2 observations, and so on up to a sample size of 10. Each data set consisted of fake IQ data: that is, the data were normally distributed with a true population mean of 100 and standard deviation 15. *On average*, the sample means turn out to be 100, regardless of sample size (panel a). However, the sample standard deviations turn out to be systematically too small (panel b), especially for small sample sizes.

The fix to this systematic bias turns out to be very simple. Here's how it works. Before tackling the standard deviation, let's look at the variance. If you recall from Section 5.2, the sample variance is defined to be the average of the squared deviations from the sample mean. That is:

$$s^2 = \frac{1}{N} \sum_{i=1}^{N} \left(X_i - \bar{X} \right)^2$$

The sample variance s^2 is a biased estimator of the population variance σ^2. But as it turns out, we only need to make a tiny tweak to transform this into an unbiased estimator. All we have to do is divide by N−1 rather than by N. If we do that, we obtain the following formula:

$$\hat{\sigma}^2 = \frac{1}{N-1} \sum_{i=1}^{N} \left(X_i - \bar{X} \right)^2$$

This is an unbiased estimator of the population variance σ. Moreover, this finally answers the question we raised in Section 5.2. Why did R give us slightly different answers when we used the `var()` function? Because the `var()` function calculates $\hat{\sigma}^2$ not s^2, that's why. A similar story applies for the standard deviation. If we divide by N−1 rather than N, our estimate of the population standard deviation becomes:

$$\hat{\sigma} = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} \left(X_i - \bar{X} \right)^2}$$

and when we use R's built in standard deviation function `sd()`, what it's doing is calculating $\hat{\sigma}$, not s.[153]

One final point: in practice, a lot of people tend to refer to $\hat{\sigma}$ (i.e., the formula where we divide by N−1) as the *sample* standard deviation. Technically, this is incorrect: the *sample* standard deviation should be equal to s (i.e., the formula where we divide by N). These aren't the same thing, either conceptually or numerically. One is a property of the sample, the other is an estimated characteristic of the population. However, in almost every real life application, what we actually care about is the estimate of the population parameter, and so people always report $\hat{\sigma}$ rather than s. This is the right number to report, of course, it's that people tend to get a little bit imprecise about terminology when they write it up, because "sample standard deviation" is shorter than "estimated population standard deviation". It's no big deal, and in practice I do the same thing everyone else does. Nevertheless, I think it's important to keep the two *concepts* separate: it's never a good idea to confuse "known properties of your sample" with "guesses about the population from which it came". The moment you start thinking that s and $\hat{\sigma}$ are the same thing, you start doing exactly that.

To finish this section off, here's another couple of tables to help keep things clear:

```
knitr::kable(data.frame(stringsAsFactors=FALSE,
                  Symbol = c("$s$", "$\\sigma$", "$\\hat{\\sigma}$", "$s^2$",
                             "$\\sigma^2$", "$\\hat{\\sigma}^2$"),
              What.is.it = c("Sample standard deviation",
                             "Population standard deviation",
                             "Estimate of the population standard deviation", "Sampl
                             "Population variance",
                             "Estimate of the population variance"),
   Do.we.know.what.it.is = c("Yes - calculated from the raw data",
                             "Almost never known for sure",
                             "Yes - but not the same as the sample standard deviatic
                             "Yes - calculated from the raw data",
                             "Almost never known for sure",
                             "Yes -  but not the same as the sample variance")
))
```

Symbol	What.is.it	Do.we.know.what.it.is
s	Sample standard deviation	Yes - calculated from the raw data
σ	Population standard deviation	Almost never known for sure
$\hat{\sigma}$	Estimate of the population standard deviation	Yes - but not the same as the sample standard deviation
s^2	Sample variance	Yes - calculated from the raw data
σ^2	Population variance	Almost never known for sure
$\hat{\sigma}^2$	Estimate of the population variance	Yes - but not the same as the sample variance

This page titled 8.4: Estimating Population Parameters is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **10.4: Estimating Population Parameters** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

8.5: Estimating a Confidence Interval

Statistics means never having to say you're certain – Unknown origin[154] but I've never found the original source.

Up to this point in this chapter, I've outlined the basics of sampling theory which statisticians rely on to make guesses about population parameters on the basis of a sample of data. As this discussion illustrates, one of the reasons we need all this sampling theory is that every data set leaves us with a some of uncertainty, so our estimates are never going to be perfectly accurate. The thing that has been missing from this discussion is an attempt to *quantify* the amount of uncertainty that attaches to our estimate. It's not enough to be able guess that, say, the mean IQ of undergraduate psychology students is 115 (yes, I just made that number up). We also want to be able to say something that expresses the degree of certainty that we have in our guess. For example, it would be nice to be able to say that there is a 95% chance that the true mean lies between 109 and 121. The name for this is a ***confidence interval*** for the mean.

Armed with an understanding of sampling distributions, constructing a confidence interval for the mean is actually pretty easy. Here's how it works. Suppose the true population mean is μ and the standard deviation is σ. I've just finished running my study that has N participants, and the mean IQ among those participants is $\bar{X}$. We know from our discussion of the central limit theorem (Section 10.3.3 that the sampling distribution of the mean is approximately normal. We also know from our discussion of the normal distribution Section 9.5 that there is a 95% chance that a normally-distributed quantity will fall within two standard deviations of the true mean. To be more precise, we can use the `qnorm()` function to compute the 2.5th and 97.5th percentiles of the normal distribution

```
qnorm( p = c(.025, .975) )
```

```
## [1] -1.959964  1.959964
```

Okay, so I lied earlier on. The more correct answer is that 95% chance that a normally-distributed quantity will fall within 1.96 standard deviations of the true mean. Next, recall that the standard deviation of the sampling distribution is referred to as the standard error, and the standard error of the mean is written as SEM. When we put all these pieces together, we learn that there is a 95% probability that the sample mean $\bar{X}$ that we have actually observed lies within 1.96 standard errors of the population mean. Mathematically, we write this as:

$$\mu - (1.96 \times \text{SEM}) \leq \bar{X} \leq \mu + (1.96 \times \text{SEM})$$

where the SEM is equal to $\sigma/\sqrt{N}$, and we can be 95% confident that this is true. However, that's not answering the question that we're actually interested in. The equation above tells us what we should expect about the sample mean, given that we know what the population parameters are. What we *want* is to have this work the other way around: we want to know what we should believe about the population parameters, given that we have observed a particular sample. However, it's not too difficult to do this. Using a little high school algebra, a sneaky way to rewrite our equation is like this:

$$\bar{X} - (1.96 \times \text{SEM}) \leq \mu \leq \bar{X} + (1.96 \times \text{SEM})$$

What this is telling is is that the range of values has a 95% probability of containing the population mean μ. We refer to this range as a ***95% confidence interval***, denoted CI_{95}. In short, as long as N is sufficiently large – large enough for us to believe that the sampling distribution of the mean is normal – then we can write this as our formula for the 95% confidence interval:

$$\text{CI}_{95} = \bar{X} \pm \left(1.96 \times \frac{\sigma}{\sqrt{N}} \right)$$

Of course, there's nothing special about the number 1.96: it just happens to be the multiplier you need to use if you want a 95% confidence interval. If I'd wanted a 70% confidence interval, I could have used the `qnorm()` function to calculate the 15th and 85th quantiles:

```
qnorm( p = c(.15, .85) )
```

```
## [1] -1.036433  1.036433
```

and so the formula for CI_{70} would be the same as the formula for CI_{95} except that we'd use 1.04 as our magic number rather than 1.96.

8.5.1 slight mistake in the formula

As usual, I lied. The formula that I've given above for the 95% confidence interval is approximately correct, but I glossed over an important detail in the discussion. Notice my formula requires you to use the standard error of the mean, SEM, which in turn requires you to use the true population standard deviation σ. Yet, in Section @ref(pointestimates I stressed the fact that we don't actually *know* the true population parameters. Because we don't know the true value of σ, we have to use an estimate of the population standard deviation $\hat{\sigma}$ instead. This is pretty straightforward to do, but this has the consequence that we need to use the quantiles of the t-distribution rather than the normal distribution to calculate our magic number; and the answer depends on the sample size. When N is very large, we get pretty much the same value using `qt()` that we would if we used `qnorm()` …

```
N <- 10000   # suppose our sample size is 10,000
qt( p = .975, df = N-1)   # calculate the 97.5th quantile of the t-dist
```

```
## [1] 1.960201
```

But when N is small, we get a much bigger number when we use the t distribution:

```
N <- 10   # suppose our sample size is 10
qt( p = .975, df = N-1)   # calculate the 97.5th quantile of the t-dist
```

```
## [1] 2.262157
```

There's nothing too mysterious about what's happening here. Bigger values mean that the confidence interval is wider, indicating that we're more uncertain about what the true value of μ actually is. When we use the t distribution instead of the normal distribution, we get bigger numbers, indicating that we have more uncertainty. And why do we have that extra uncertainty? Well, because our estimate of the population standard deviation ^σ might be wrong! If it's wrong, it implies that we're a bit less sure about what our sampling distribution of the mean actually looks like… and this uncertainty ends up getting reflected in a wider confidence interval.

8.5.2 Interpreting a confidence interval

The hardest thing about confidence intervals is understanding what they *mean*. Whenever people first encounter confidence intervals, the first instinct is almost always to say that "there is a 95% probabaility that the true mean lies inside the confidence interval". It's simple, and it seems to capture the common sense idea of what it means to say that I am "95% confident". Unfortunately, it's not quite right. The intuitive definition relies very heavily on your own personal *beliefs* about the value of the population mean. I say that I am 95% confident because those are my beliefs. In everyday life that's perfectly okay, but if you remember back to Section 9.2, you'll notice that talking about personal belief and confidence is a Bayesian idea. Personally (speaking as a Bayesian) I have no problem with the idea that the phrase "95% probability" is allowed to refer to a personal belief. However, confidence intervals are *not* Bayesian tools. Like everything else in this chapter, confidence intervals are *frequentist* tools, and if you are going to use frequentist methods then it's not appropriate to attach a Bayesian interpretation to them. If you use frequentist methods, you must adopt frequentist interpretations!

Okay, so if that's not the right answer, what is? Remember what we said about frequentist probability: the only way we are allowed to make "probability statements" is to talk about a sequence of events, and to count up the frequencies of different kinds of events. From that perspective, the interpretation of a 95% confidence interval must have something to do with replication. Specifically: if we replicated the experiment over and over again and computed a 95% confidence interval for each replication, then 95% of those

intervals would contain the true mean. More generally, 95% of all confidence intervals constructed using this procedure should contain the true population mean. This idea is illustrated in Figure 10.13, which shows 50 confidence intervals constructed for a "measure 10 IQ scores" experiment (top panel) and another 50 confidence intervals for a "measure 25 IQ scores" experiment (bottom panel). A bit fortuitously, across the 100 replications that I simulated, it turned out that exactly 95 of them contained the true mean.

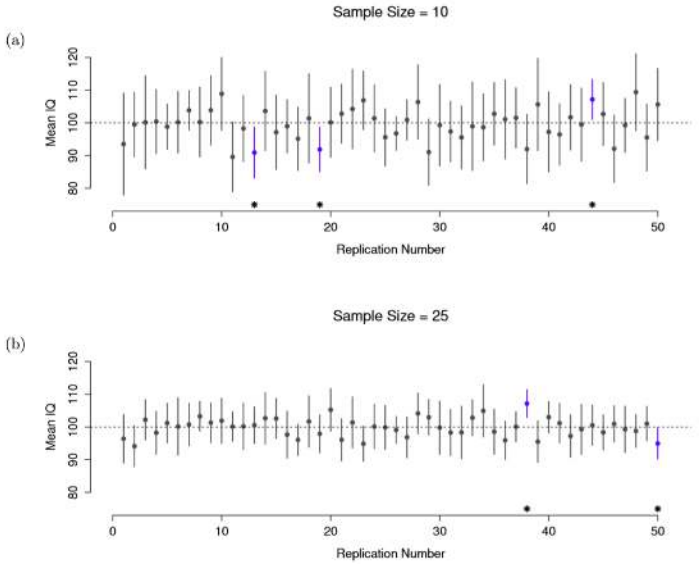


Figure 10.13: 95% confidence intervals. The top (panel a) shows 50 simulated replications of an experiment in which we measure the IQs of 10 people. The dot marks the location of the sample mean, and the line shows the 95% confidence interval. In total 47 of the 50 confidence intervals do contain the true mean (i.e., 100), but the three intervals marked with asterisks do not. The lower graph (panel b) shows a similar simulation, but this time we simulate replications of an experiment that measures the IQs of 25 people.

The critical difference here is that the Bayesian claim makes a probability statement about the population mean (i.e., it refers to our uncertainty about the population mean), which is not allowed under the frequentist interpretation of probability because you can't "replicate" a population! In the frequentist claim, the population mean is fixed and no probabilistic claims can be made about it. Confidence intervals, however, are repeatable so we can replicate experiments. Therefore a frequentist is allowed to talk about the probability that the *confidence interval* (a random variable) contains the true mean; but is not allowed to talk about the probability that the *true population mean* (not a repeatable event) falls within the confidence interval.

I know that this seems a little pedantic, but it does matter. It matters because the difference in interpretation leads to a difference in the mathematics. There is a Bayesian alternative to confidence intervals, known as *credible intervals*. In most situations credible intervals are quite similar to confidence intervals, but in other cases they are drastically different. As promised, though, I'll talk more about the Bayesian perspective in Chapter 17.

8.5.3 Calculating confidence intervals in R

As far as I can tell, the core packages in R don't include a simple function for calculating confidence intervals for the mean. They *do* include a lot of complicated, extremely powerful functions that can be used to calculate confidence intervals associated with lots of different things, such as the `confint()` function that we'll use in Chapter 15. But I figure that when you're first learning statistics, it might be useful to start with something simpler. As a consequence, the `lsr` package includes a function called `ciMean()` which you can use to calculate your confidence intervals. There are two arguments that you might want to specify:[155]

- `x` . This should be a numeric vector containing the data.
- `conf` . This should be a number, specifying the confidence level. By default, `conf = .95` , since 95% confidence intervals are the de facto standard in psychology.

So, for example, if I load the `afl24.Rdata` file, calculate the confidence interval associated with the mean attendance:

```
> ciMean( x = afl$attendance )
    2.5%    97.5%
31597.32 32593.12
```

Hopefully that's fairly clear.

8.5.4 Plotting confidence intervals in R

There's several different ways you can draw graphs that show confidence intervals as error bars. I'll show three versions here, but this certainly doesn't exhaust the possibilities. In doing so, what I'm assuming is that you want to draw is a plot showing the means and confidence intervals for one variable, broken down by different levels of a second variable. For instance, in our `afl` data that we discussed earlier, we might be interested in plotting the average `attendance` by `year` . I'll do this using two different functions, `bargraph.CI()` and `lineplot.CI()` (both of which are in the `sciplot` package). Assuming that you've installed these packages on your system (see Section 4.2 if you've forgotten how to do this), you'll need to load them. You'll also need to load the `lsr` package, because we'll make use of the `ciMean()` function to actually calculate the confidence intervals

```
load( "./rbook-master/data/afl24.Rdata" )  # contains the "afl" data frame
library( sciplot )     # bargraph.CI() and lineplot.CI() functions
library( lsr )         # ciMean() function
```

Here's how to plot the means and confidence intervals drawn using `bargraph.CI()` .

```
bargraph.CI( x.factor = year,             # grouping variable
             response = attendance,       # outcome variable
             data = afl,                  # data frame with the variables
             ci.fun= ciMean,              # name of the function to calculate CIs
             xlab = "Year",               # x-axis label
             ylab = "Average Attendance" # y-axis label
 )
```

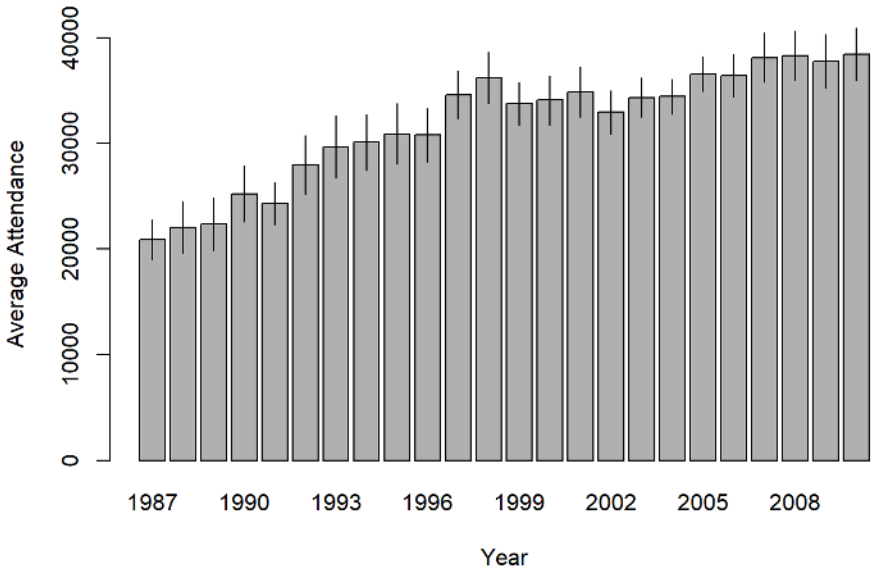


Figure 10.14: Means and 95% confidence intervals for AFL `attendance` , plotted separately for each `year` from 1987 to 2010. This graph was drawn using the `bargraph.CI()` function.

We can use the same arguments when calling the `lineplot.CI()` function:

```
lineplot.CI( x.factor = year,             # grouping variable
             response = attendance,       # outcome variable
             data = afl,                  # data frame with the variables
             ci.fun= ciMean,              # name of the function to calculate CIs
             xlab = "Year",               # x-axis label
             ylab = "Average Attendance" # y-axis label
)
```

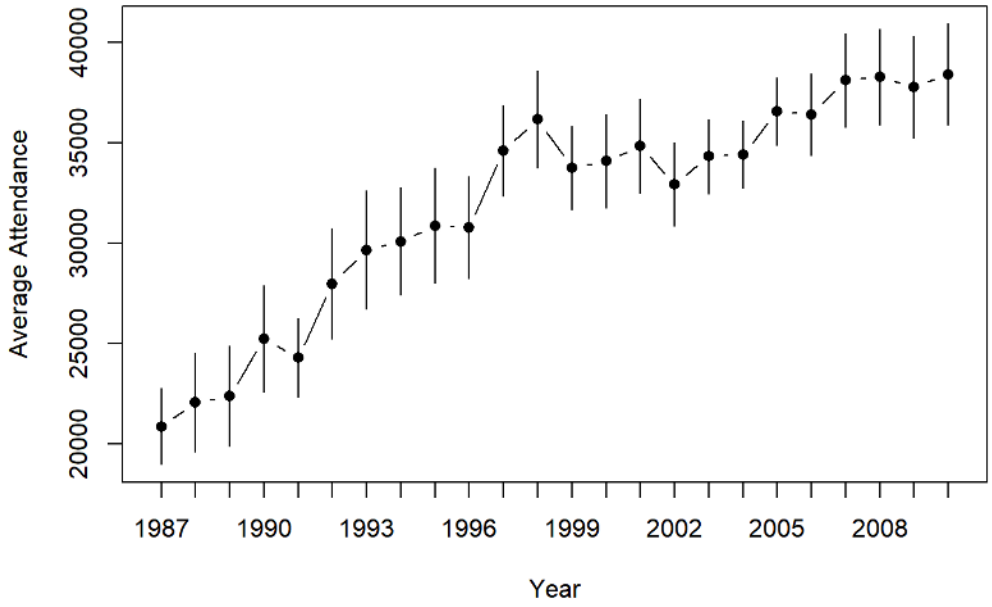


Figure 10.15: Means and 95% confidence intervals for AFL `attendance`, plotted separately for each `year` from 1987 to 2010. This graph was drawn using the `lineplot.CI()` function.

This page titled 8.5: Estimating a Confidence Interval is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **10.5: Estimating a Confidence Interval** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

8.6: Summary

In this chapter I've covered two main topics. The first half of the chapter talks about sampling theory, and the second half talks about how we can use sampling theory to construct estimates of the population parameters. The section breakdown looks like this:

- Basic ideas about samples, sampling and populations (Section 10.1)
- Statistical theory of sampling: the law of large numbers (Section 10.2), sampling distributions and the central limit theorem (Section 10.3).
- Estimating means and standard deviations (Section 10.4)
- Estimating a confidence interval (Section 10.5)

As always, there's a lot of topics related to sampling and estimation that aren't covered in this chapter, but for an introductory psychology class this is fairly comprehensive I think. For most applied researchers you won't need much more theory than this. One big question that I haven't touched on in this chapter is what you do when you don't have a simple random sample. There is a lot of statistical theory you can draw on to handle this situation, but it's well beyond the scope of this book.

References

Stigler, S. M. 1986. *The History of Statistics*. Cambridge, MA: Harvard University Press.

Keynes, John Maynard. 1923. *A Tract on Monetary Reform*. London: Macmillan; Company.

147. The proper mathematical definition of randomness is extraordinarily technical, and way beyond the scope of this book. We'll be non-technical here and say that a process has an element of randomness to it whenever it is possible to repeat the process and get different answers each time.
148. Nothing in life is that simple: there's not an obvious division of people into binary categories like "schizophrenic" and "not schizophrenic". But this isn't a clinical psychology text, so please forgive me a few simplifications here and there.
149. Technically, the law of large numbers pertains to any sample statistic that can be described as an average of independent quantities. That's certainly true for the sample mean. However, it's also possible to write many other sample statistics as averages of one form or another. The variance of a sample, for instance, can be rewritten as a kind of average and so is subject to the law of large numbers. The minimum value of a sample, however, cannot be written as an average of anything and is therefore not governed by the law of large numbers.
150. As usual, I'm being a bit sloppy here. The central limit theorem is a bit more general than this section implies. Like most introductory stats texts, I've discussed one situation where the central limit theorem holds: when you're taking an average across lots of independent events drawn from the same distribution. However, the central limit theorem is much broader than this. There's a whole class of things called "U-statistics" for instance, all of which satisfy the central limit theorem and therefore become normally distributed for large sample sizes. The mean is one such statistic, but it's not the only one.
151. Please note that if you were *actually* interested in this question, you would need to be a *lot* more careful than I'm being here. You *can't* just compare IQ scores in Whyalla to Port Pirie and assume that any differences are due to lead poisoning. Even if it were true that the only differences between the two towns corresponded to the different refineries (and it isn't, not by a long shot), you need to account for the fact that people already *believe* that lead pollution causes cognitive deficits: if you recall back to Chapter 2, this means that there are different demand effects for the Port Pirie sample than for the Whyalla sample. In other words, you might end up with an illusory group difference in your data, caused by the fact that people *think* that there is a real difference. I find it pretty implausible to think that the locals wouldn't be well aware of what you were trying to do if a bunch of researchers turned up in Port Pirie with lab coats and IQ tests, and even less plausible to think that a lot of people would be pretty resentful of you for doing it. Those people won't be as co-operative in the tests. Other people in Port Pirie might be *more* motivated to do well because they don't want their home town to look bad. The motivational effects that would apply in Whyalla are likely to be weaker, because people don't have any concept of "iron ore poisoning" in the same way that they have a concept for "lead poisoning". Psychology is *hard*.
152. I should note that I'm hiding something here. Unbiasedness is a desirable characteristic for an estimator, but there are other things that matter besides bias. However, it's beyond the scope of this book to discuss this in any detail. I just want to draw your attention to the fact that there's some hidden complexity here.

153. , I'm hiding something else here. In a bizarre and counterintuitive twist, since $\hat{\sigma}^2$ is an unbiased estimator of σ^2, you'd assume that taking the square root would be fine, and $\hat{\sigma}$ would be an unbiased estimator of σ. Right? Weirdly, it's not. There's actually a subtle, tiny bias in $\hat{\sigma}$. This is just bizarre: $\hat{\sigma}^2$ is and unbiased estimate of the population variance σ^2, but when you take the square root, it turns out that ^σ is a biased estimator of the population standard deviation σ. Weird, weird, weird, right? So, why is $\hat{\sigma}$ biased? The technical answer is "because non-linear transformations (e.g., the square root) don't commute with expectation", but that just sounds like gibberish to everyone who hasn't taken a course in mathematical statistics. Fortunately, it doesn't matter for practical purposes. The bias is small, and in real life everyone uses $\hat{\sigma}$ and it works just fine. Sometimes mathematics is just annoying.
154. This quote appears on a great many t-shirts and websites, and even gets a mention in a few academic papers (e.g., \url{http://www.amstat.org/publications/jse/v10n3/friedman.html
155. As of the current writing, these are the only arguments to the function. However, I am planning to add a bit more functionality to `ciMean()`. However, regardless of what those future changes might look like, the `x` and `conf` arguments will remain the same, and the commands used in this book will still work.

This page titled 8.6: Summary is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **10.6: Summary** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

8.7: Statistical Literacy

Learning Objectives

- No "Large Conclusions" from "Tiny" Samples?

In July of 2011, Gene Munster of Piper Jaffray reported the results of a survey in a note to clients. This research was reported throughout the media. Perhaps the fullest description was presented on the CNNMoney website (A service of CNN, Fortune, and Money) in an article entitled “Survey: iPhone retention 94% vs. Android 47%.” The data were collected by asking people in food courts and baseball stadiums what their current phone was and what phone they planned to buy next. The data were collected in the summer of 2011. Below is a portion of the data:

Table 8.7.1: Sample phone retention data

Phone	Keep	Change	Proportion
iPhone	58	4	0.94
Android	17	19	0.47

The article contains the strong caution: “It's only a tiny sample, so large conclusions must not be drawn.” This caution appears to be a welcome change from the overstating of findings typically found in the media. But has this report understated the importance of the study? Perhaps it is valid to draw some "large conclusions."

Example 8.7.1: what do you think?

Is it possible to conclude the vast majority of iPhone owners in the population sampled plan to buy another iPhone or is the sample size too small to justify this conclusion?

Solution

The confidence interval on the proportion extends from 0.87 to 1.0 (some methods give the interval from 0.85 to 0.97). Even the lower bound indicates the vast majority of iPhone owners plan to buy another iPhone. A strong conclusion can be made even with this sample size.

This page titled 8.7: Statistical Literacy is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **10.13: Statistical Literacy** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

8.E: Estimation (Exercises)

You may want to use the Analysis Lab and various calculators for some of these exercises.

Calculators:

- Inverse t Distribution: Finds t for a confidence interval.
- t Distribution: Computes areas of the t distribution.
- Fisher's r to z': Computes transformations in both directions.
- Inverse Normal Distribution: Use for confidence intervals.

General Questions

Q1

When would the mean grade in a class on a final exam be considered a statistic? When would it be considered a parameter? (relevant section)

Q2

Define bias in terms of expected value. (relevant section)

Q3

Is it possible for a statistic to be unbiased yet very imprecise? How about being very accurate but biased? (relevant section)

Q4

Why is a 99% confidence interval wider than a 95% confidence interval? (relevant section & relevant section)

Q5

When you construct a 95% confidence interval, what are you 95% confident about? (relevant section)

Q6

What is the difference in the computation of a confidence interval between cases in which you know the population standard deviation and cases in which you have to estimate it? (relevant section & relevant section)

Q7

Assume a researcher found that the correlation between a test he or she developed and job performance was 0.55 in a study of 28 employees. If correlations under 0.35 are considered unacceptable, would you have any reservations about using this test to screen job applicants? (relevant section)

Q8

What is the effect of sample size on the width of a confidence interval? (relevant section & relevant section)

Q9

How does the t distribution compare with the normal distribution? How does this difference affect the size of confidence intervals constructed using z relative to those constructed using t? Does sample size make a difference? (relevant section)

Q10

The effectiveness of a blood-pressure drug is being investigated. How might an experimenter demonstrate that, on average, the reduction in systolic blood pressure is 20 or more? (relevant section & relevant section)

Q11

A population is known to be normally distributed with a standard deviation of 2.8.

a. Compute the 95% confidence interval on the mean based on the following sample of nine: $8, 9, 10, 13, 14, 16, 17, 20, 21$.

b. Now compute the 99% confidence interval using the same data. (relevant section)

Q12

A person claims to be able to predict the outcome of flipping a coin. This person is correct 16/25 times. Compute the 95% confidence interval on the proportion of times this person can predict coin flips correctly. What conclusion can you draw about this test of his ability to predict the future? (relevant section)

Q13

What does it mean that the variance (computed by dividing by N) is a biased statistic? (relevant section)

Q14

A confidence interval for the population mean computed from an N of 16 ranges from 12 to 28. A new sample of 36 observations is going to be taken. You can't know in advance exactly what the confidence interval will be because it depends on the random sample. Even so, you should have some idea of what it will be. Give your best estimation. (relevant section)

Q15

You take a sample of 22 from a population of test scores, and the mean of your sample is 60.

a. You know the standard deviation of the population is 10. What is the 99% confidence interval on the population mean.
b. Now assume that you do not know the population standard deviation, but the standard deviation in your sample is 10. What is the 99% confidence interval on the mean now? (relevant section)

Q16

You read about a survey in a newspaper and find that 70% of the 250 people sampled prefer Candidate A. You are surprised by this survey because you thought that more like 50% of the population preferred this candidate. Based on this sample, is 50% a possible population proportion? Compute the 95% confidence interval to be sure. (relevant section)

Q17

Heights for teenage boys and girls were calculated. The mean height for the sample of 12 boys was 174 cm and the variance was 62. For the sample of 12 girls, the mean was 166 cm and the variance was 65.

a. What is the 95% confidence interval on the difference between population means?
b. What is the 99% confidence interval on the difference between population means?
c. Do you think the mean difference in the population could be about 5? Why or why not? (relevant section)

Q18

You were interested in how long the average psychology major at your college studies per night, so you asked 10 psychology majors to tell you the amount they study. They told you the following times: $2, 1.5, 3, 2, 3.5, 1, 0.5, 3, 2, 4$.

a. Find the 95% confidence interval on the population mean.
b. Find the 90% confidence interval on the population mean. (relevant section)

Q19

True/false: As the sample size gets larger, the probability that the confidence interval will contain the population mean gets higher. (relevant section & relevant section)

Q20

True/false: You have a sample of 9 men and a sample of 8 women. The degrees of freedom for the t value in your confidence interval on the difference between means is 16. (relevant section & relevant section)

Q21

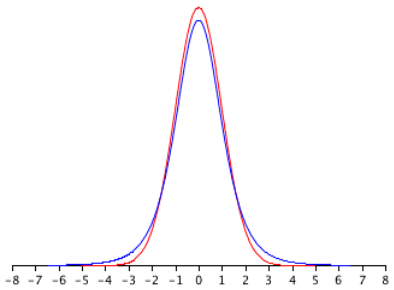
True/false: Greek letters are used for statistics as opposed to parameters. (relevant section)

Q22

True/false: In order to construct a confidence interval on the difference between means, you need to assume that the populations have the same variance and are both normally distributed. (relevant section)

Q23

True/false: The red distribution represents the t distribution and the blue distribution represents the normal distribution. (relevant section)



Questions from Case Studies

The following questions are from the Angry Moods (AM) case study.

Q24

(AM#6c) Is there a difference in how much males and females use aggressive behavior to improve an angry mood? For the "Anger-Out" scores, compute a 99% confidence interval on the difference between gender means. (relevant section)

Q25

(AM#10) Calculate the 95% confidence interval for the difference between the mean Anger-In score for the athletes and non-athletes. What can you conclude? (relevant section)

Q26

Find the 95% confidence interval on the population correlation between the Anger-Out and Control-Out scores. (relevant section)

The following questions are from the Flatulence (F) case study.

Q27

(F#8) Compare men and women on the variable "perday." Compute the 95% confidence interval on the difference between means. (relevant section)

Q28

(F#10) What is the 95% confidence interval of the mean time people wait before farting in front of a romantic partner. (relevant section)

The following questions use data from the Animal Research (AR) case study.

Q29

(AR#3) What percentage of the women studied in this sample strongly agreed (gave a rating of 7) that using animals for research is wrong?

Q30

Use the proportion you computed in #29. Compute the 95% confidence interval on the population proportion of women who strongly agree that animal research is wrong. (relevant section)

Q31

Compute a 95% confidence interval on the difference between the gender means with respect to their beliefs that animal research is wrong. (relevant section)

The following question is from the ADHD Treatment (AT) case study.

Q32

(AT#8) What is the correlation between the participants' correct number of responses after taking the placebo and their correct number of responses after taking 0.60 mg/kg of MPH? Compute the 95% confidence interval on the population correlation. (relevant section)

The following question is from the Weapons and Aggression (WA) case study.

Q33

(WA#4) Recall that the hypothesis is that a person can name an aggressive word more quickly if it is preceded by a weapon word prime than if it is preceded by a neutral word prime. The first step in testing this hypothesis is to compute the difference between

i. the naming time of aggressive words when preceded by a neutral word prime and
ii. the naming time of aggressive words when preceded by a weapon word prime separately for each of the 32 participants. That is, compute an $-aw$ for each participant.

a. Would the hypothesis of this study be supported if the difference were positive or if it were negative?
b. What is the mean of this difference score? (relevant section)
c. What is the standard deviation of this difference score? (relevant section)
d. What is the 95% confidence interval of the mean difference score? (relevant section)
e. What does the confidence interval computed in (d) say about the hypothesis.

The following question is from the Diet and Health (WA) case study.

Q34

Compute a 95% confidence interval on the proportion of people who are healthy on the AHA diet.

	Cancers	Deaths	Nonfatal illness	Healthy	Total
AHA					

15 24 25 239 303 Mediterranean 7 14 8 273 302 Total 22 38 33 512 605

The following questions are from (reproduced with permission) 

Visit the site

Q35

Suppose that you take a random sample of 10,000 Americans and find that 1,111 are left-handed. You perform a test of significance to assess whether the sample data provide evidence that more than 10% of all Americans are left-handed, and you calculate a test statistic of 3.70 and a p-value of 0.0001. Furthermore, you calculate a 99% confidence interval for the proportion of left-handers in America to be (0.103, 0.119). Consider the following statements: The sample provides strong evidence that more than 10% of all Americans are left-handed. The sample provides evidence that the proportion of left-handers in America is much larger than 10%. Which of these two statements is the more appropriate conclusion to draw? Explain your answer based on the results of the significance test and confidence interval.

Q36

A student wanted to study the ages of couples applying for marriage licenses in his county. He studied a sample of 94 marriage licenses and found that in 67 cases the husband was older than the wife. Do the sample data provide strong evidence that the husband is usually older than the wife among couples applying for marriage licenses in that county? Explain briefly and justify your answer.

Q37

Imagine that there are 100 different researchers each studying the sleeping habits of college freshmen. Each researcher takes a random sample of size 50 from the same population of freshmen. Each researcher is trying to estimate the mean hours of sleep that freshmen get at night, and each one constructs a 95% confidence interval for the mean. Approximately how many of these 100 confidence intervals will NOT capture the true mean?

1. None
2. 1 or 2
3. 3 to 7
4. about half
5. 95 to 100
6. other

Selected Answers

S11

a. $(12.39, 16.05)$

S12

$(0.43, 0.85)$

S15

b. $(53.96, 66.04)$

S17

a. $(1.25, 14.75)$

S18

a. $(1.45, 3.05)$

S26

$(-0.713, -0.414)$

S27

$(-0.98, 3.09)$

S29

41%

S33

b. 7.16

This page titled 8.E: Estimation (Exercises) is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **10.E: Estimation (Exercises)** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

CHAPTER OVERVIEW

9: Hypothesis Testing

> *The process of induction is the process of assuming the simplest law that can be made to harmonize with our experience. This process, however, has no logical foundation but only a psychological one. It is clear that there are no grounds for believing that the simplest course of events will really happen. It is an hypothesis that the sun will rise tomorrow: and this means that we do not know whether it will rise.*
>
> *– Ludwig Wittgenstein*[156]

In the last chapter, I discussed the ideas behind estimation, which is one of the two "big ideas" in inferential statistics. It's now time to turn out attention to the other big idea, which is *hypothesis testing*. In its most abstract form, hypothesis testing really a very simple idea: the researcher has some theory about the world, and wants to determine whether or not the data actually support that theory. However, the details are messy, and most people find the theory of hypothesis testing to be the most frustrating part of statistics. The structure of the chapter is as follows. Firstly, I'll describe how hypothesis testing works, in a fair amount of detail, using a simple running example to show you how a hypothesis test is "built". I'll try to avoid being too dogmatic while doing so, and focus instead on the underlying logic of the testing procedure.[157] Afterwards, I'll spend a bit of time talking about the various dogmas, rules and heresies that surround the theory of hypothesis testing.

9.1: A Menagerie of Hypotheses
9.2: Two Types of Errors
9.3: Test Statistics and Sampling Distributions
9.4: Making Decisions
9.5: The p value of a test
9.6: Reporting the Results of a Hypothesis Test
9.7: Running the Hypothesis Test in Practice
9.8: Effect Size, Sample Size and Power
9.9: Some Issues to Consider
9.10: Misconceptions of Hypothesis Testing
9.11: Summary
9.12: Statistical Literacy
9.13: Logic of Hypothesis Testing (Exercises)

This page titled 9: Hypothesis Testing is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

9.1: A Menagerie of Hypotheses

Eventually we all succumb to madness. For me, that day will arrive once I'm finally promoted to full professor. Safely ensconced in my ivory tower, happily protected by tenure, I will finally be able to take leave of my senses (so to speak), and indulge in that most thoroughly unproductive line of psychological research: the search for extrasensory perception (ESP).[158]

Let's suppose that this glorious day has come. My first study is a simple one, in which I seek to test whether clairvoyance exists. Each participant sits down at a table, and is shown a card by an experimenter. The card is black on one side and white on the other. The experimenter takes the card away, and places it on a table in an adjacent room. The card is placed black side up or white side up completely at random, with the randomisation occurring only after the experimenter has left the room with the participant. A second experimenter comes in and asks the participant which side of the card is now facing upwards. It's purely a one-shot experiment. Each person sees only one card, and gives only one answer; and at no stage is the participant actually in contact with someone who knows the right answer. My data set, therefore, is very simple. I have asked the question of N people, and some number X of these people have given the correct response. To make things concrete, let's suppose that I have tested N=100 people, and X=62 of these got the answer right… a surprisingly large number, sure, but is it large enough for me to feel safe in claiming I've found evidence for ESP? This is the situation where hypothesis testing comes in useful. However, before we talk about how to *test* hypotheses, we need to be clear about what we mean by hypotheses.

9.1.1 Research hypotheses versus statistical hypotheses

The first distinction that you need to keep clear in your mind is between research hypotheses and statistical hypotheses. In my ESP study, my overall scientific goal is to demonstrate that clairvoyance exists. In this situation, I have a clear research goal: I am hoping to discover evidence for ESP. In other situations I might actually be a lot more neutral than that, so I might say that my research goal is to determine whether or not clairvoyance exists. Regardless of how I want to portray myself, the basic point that I'm trying to convey here is that a research hypothesis involves making a substantive, testable scientific claim… if you are a psychologist, then your research hypotheses are fundamentally *about* psychological constructs. Any of the following would count as ***research hypotheses***:

- *Listening to music reduces your ability to pay attention to other things.* This is a claim about the causal relationship between two psychologically meaningful concepts (listening to music and paying attention to things), so it's a perfectly reasonable research hypothesis.
- *Intelligence is related to personality*. Like the last one, this is a relational claim about two psychological constructs (intelligence and personality), but the claim is weaker: correlational not causal.
- *Intelligence* is* speed of information processing. *This hypothesis has a quite different character: it's not actually a relational claim at all. It's an ontological claim about the fundamental character of intelligence (and I'm pretty sure it's wrong). It's worth expanding on this one actually: It's usually easier to think about how to construct experiments to test research hypotheses of the form "does X affect Y?" than it is to address claims like "what is X?" And in practice, what usually happens is that you find ways of testing relational claims that follow from your ontological ones. For instance, if I believe that intelligence* is* speed of information processing in the brain, my experiments will often involve looking for relationships between measures of intelligence and measures of speed. As a consequence, most everyday research questions do tend to be relational in nature, but they're almost always motivated by deeper ontological questions about the state of nature.

Notice that in practice, my research hypotheses could overlap a lot. My ultimate goal in the ESP experiment might be to test an ontological claim like "ESP exists", but I might operationally restrict myself to a narrower hypothesis like "Some people can `see' objects in a clairvoyant fashion". That said, there are some things that really don't count as proper research hypotheses in any meaningful sense:

- *Love is a battlefield*. This is too vague to be testable. While it's okay for a research hypothesis to have a degree of vagueness to it, it has to be possible to operationalise your theoretical ideas. Maybe I'm just not creative enough to see it, but I can't see how this can be converted into any concrete research design. If that's true, then this isn't a scientific research hypothesis, it's a pop song. That doesn't mean it's not interesting – a lot of deep questions that humans have fall into this category. Maybe one day science will be able to construct testable theories of love, or to test to see if God exists, and so on; but right now we can't, and I wouldn't bet on ever seeing a satisfying scientific approach to either.

- *The first rule of tautology club is the first rule of tautology club*. This is not a substantive claim of any kind. It's true by definition. No conceivable state of nature could possibly be inconsistent with this claim. As such, we say that this is an unfalsifiable hypothesis, and as such it is outside the domain of science. Whatever else you do in science, your claims must have the possibility of being wrong.
- *More people in my experiment will say "yes" than "no"*. This one fails as a research hypothesis because it's a claim about the data set, not about the psychology (unless of course your actual research question is whether people have some kind of "yes" bias!). As we'll see shortly, this hypothesis is starting to sound more like a statistical hypothesis than a research hypothesis.

As you can see, research hypotheses can be somewhat messy at times; and ultimately they are *scientific* claims. ***Statistical hypotheses*** are neither of these two things. Statistical hypotheses must be mathematically precise, and they must correspond to specific claims about the characteristics of the data generating mechanism (i.e., the "population"). Even so, the intent is that statistical hypotheses bear a clear relationship to the substantive research hypotheses that you care about! For instance, in my ESP study my research hypothesis is that some people are able to see through walls or whatever. What I want to do is to "map" this onto a statement about how the data were generated. So let's think about what that statement would be. The quantity that I'm interested in within the experiment is P("correct"), the true-but-unknown probability with which the participants in my experiment answer the question correctly. Let's use the Greek letter θ (theta) to refer to this probability. Here are four different statistical hypotheses:

- If ESP doesn't exist and if my experiment is well designed, then my participants are just guessing. So I should expect them to get it right half of the time and so my statistical hypothesis is that the true probability of choosing correctly is $\theta=0.5$.
- Alternatively, suppose ESP does exist and participants can see the card. If that's true, people will perform better than chance. The statistical hypotheis would be that $\theta>0.5$.
- A third possibility is that ESP does exist, but the colours are all reversed and people don't realise it (okay, that's wacky, but you never know…). If that's how it works then you'd expect people's performance to be *below* chance. This would correspond to a statistical hypothesis that $\theta<0.5$.
- Finally, suppose ESP exists, but I have no idea whether people are seeing the right colour or the wrong one. In that case, the only claim I could make about the data would be that the probability of making the correct answer is *not* equal to 50. This corresponds to the statistical hypothesis that $\theta\neq0.5$.

All of these are legitimate examples of a statistical hypothesis because they are statements about a population parameter and are meaningfully related to my experiment.

What this discussion makes clear, I hope, is that when attempting to construct a statistical hypothesis test the researcher actually has two quite distinct hypotheses to consider. First, he or she has a research hypothesis (a claim about psychology), and this corresponds to a statistical hypothesis (a claim about the data generating population). In my ESP example, these might be

Dan.s.research.hypothesis	Dan.s.statistical.hypothesis
ESP.exists	$\theta\neq0.5$

And the key thing to recognise is this: *a statistical hypothesis test is a test of the statistical hypothesis, not the research hypothesis*. If your study is badly designed, then the link between your research hypothesis and your statistical hypothesis is broken. To give a silly example, suppose that my ESP study was conducted in a situation where the participant can actually see the card reflected in a window; if that happens, I would be able to find very strong evidence that $\theta\neq0.5$, but this would tell us nothing about whether "ESP exists".

9.1.2 Null hypotheses and alternative hypotheses

So far, so good. I have a research hypothesis that corresponds to what I want to believe about the world, and I can map it onto a statistical hypothesis that corresponds to what I want to believe about how the data were generated. It's at this point that things get somewhat counterintuitive for a lot of people. Because what I'm about to do is invent a new statistical hypothesis (the "null" hypothesis, H_0) that corresponds to the exact opposite of what I want to believe, and then focus exclusively on that, almost to the neglect of the thing I'm actually interested in (which is now called the "alternative" hypothesis, H_1). In our ESP example, the null hypothesis is that $\theta=0.5$, since that's what we'd expect if ESP *didn't* exist. My hope, of course, is that ESP is totally real, and so the *alternative* to this null hypothesis is $\theta\neq0.5$. In essence, what we're doing here is dividing up the possible values of θ into two

groups: those values that I really hope aren't true (the null), and those values that I'd be happy with if they turn out to be right (the alternative). Having done so, the important thing to recognise is that the goal of a hypothesis test is *not* to show that the alternative hypothesis is (probably) true; the goal is to show that the null hypothesis is (probably) false. Most people find this pretty weird.

The best way to think about it, in my experience, is to imagine that a hypothesis test is a criminal trial[159]… *the trial of the null hypothesis*. The null hypothesis is the defendant, the researcher is the prosecutor, and the statistical test itself is the judge. Just like a criminal trial, there is a presumption of innocence: the null hypothesis is *deemed* to be true unless you, the researcher, can prove beyond a reasonable doubt that it is false. You are free to design your experiment however you like (within reason, obviously!), and your goal when doing so is to maximise the chance that the data will yield a conviction… for the crime of being false. The catch is that the statistical test sets the rules of the trial, and those rules are designed to protect the null hypothesis – specifically to ensure that if the null hypothesis is actually true, the chances of a false conviction are guaranteed to be low. This is pretty important: after all, the null hypothesis doesn't get a lawyer. And given that the researcher is trying desperately to prove it to be false, *someone* has to protect it.

This page titled 9.1: A Menagerie of Hypotheses is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **11.1: A Menagerie of Hypotheses** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

9.2: Two Types of Errors

Before going into details about how a statistical test is constructed, it's useful to understand the philosophy behind it. I hinted at it when pointing out the similarity between a null hypothesis test and a criminal trial, but I should now be explicit. Ideally, we would like to construct our test so that we never make any errors. Unfortunately, since the world is messy, this is never possible. Sometimes you're just really unlucky: for instance, suppose you flip a coin 10 times in a row and it comes up heads all 10 times. That feels like very strong evidence that the coin is biased (and it is!), but of course there's a 1 in 1024 chance that this would happen even if the coin was totally fair. In other words, in real life we *always* have to accept that there's a chance that we did the wrong thing. As a consequence, the goal behind statistical hypothesis testing is not to *eliminate* errors, but to *minimise* them.

At this point, we need to be a bit more precise about what we mean by "errors". Firstly, let's state the obvious: it is either the case that the null hypothesis is true, or it is false; and our test will either reject the null hypothesis or retain it.[160] So, as the table below illustrates, after we run the test and make our choice, one of four things might have happened:

	retain H_0	reject H_0
H_0 is true	correct decision	error (type I)
H_0 is false	error (type II)	correct decision

As a consequence there are actually *two* different types of error here. If we reject a null hypothesis that is actually true, then we have made a ***type I error***. On the other hand, if we retain the null hypothesis when it is in fact false, then we have made a ***type II error***.

Remember how I said that statistical testing was kind of like a criminal trial? Well, I meant it. A criminal trial requires that you establish "beyond a reasonable doubt" that the defendant did it. All of the evidentiary rules are (in theory, at least) designed to ensure that there's (almost) no chance of wrongfully convicting an innocent defendant. The trial is designed to protect the rights of a defendant: as the English jurist William Blackstone famously said, it is "better that ten guilty persons escape than that one innocent suffer." In other words, a criminal trial doesn't treat the two types of error in the same way~… punishing the innocent is deemed to be much worse than letting the guilty go free. A statistical test is pretty much the same: the single most important design principle of the test is to *control* the probability of a type I error, to keep it below some fixed probability. This probability, which is denoted α, is called the ***significance level*** of the test (or sometimes, the *size* of the test). And I'll say it again, because it is so central to the whole set-up~… a hypothesis test is said to have significance level α if the type I error rate is no larger than α.

So, what about the type II error rate? Well, we'd also like to keep those under control too, and we denote this probability by β. However, it's much more common to refer to the ***power*** of the test, which is the probability with which we reject a null hypothesis when it really is false, which is $1-\beta$. To help keep this straight, here's the same table again, but with the relevant numbers added:

	retain H_0	reject H_0
H_0 is true	$1-\alpha$ (probability of correct retention)	α (type I error rate)
H_0 is false	β (type II error rate)	$1-\beta$ (power of the test)

A "powerful" hypothesis test is one that has a small value of β, while still keeping α fixed at some (small) desired level. By convention, scientists make use of three different α levels: .05, .01 and .001. Notice the asymmetry here~… the tests are designed to *ensure* that the α level is kept small, but there's no corresponding guarantee regarding β. We'd certainly *like* the type II error rate to be small, and we try to design tests that keep it small, but this is very much secondary to the overwhelming need to control the type I error rate. As Blackstone might have said if he were a statistician, it is "better to retain 10 false null hypotheses than to reject a single true one". To be honest, I don't know that I agree with this philosophy – there are situations where I think it makes sense, and situations where I think it doesn't – but that's neither here nor there. It's how the tests are built.

This page titled 9.2: Two Types of Errors is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **11.2: Two Types of Errors** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

9.3: Test Statistics and Sampling Distributions

At this point we need to start talking specifics about how a hypothesis test is constructed. To that end, let's return to the ESP example. Let's ignore the actual data that we obtained, for the moment, and think about the structure of the experiment. Regardless of what the actual numbers are, the *form* of the data is that X out of N people correctly identified the colour of the hidden card. Moreover, let's suppose for the moment that the null hypothesis really is true: ESP doesn't exist, and the true probability that anyone picks the correct colour is exactly $\theta=0.5$. What would we *expect* the data to look like? Well, obviously, we'd expect the proportion of people who make the correct response to be pretty close to 50%. Or, to phrase this in more mathematical terms, we'd say that X/N is approximately 0.5. Of course, we wouldn't expect this fraction to be *exactly* 0.5: if, for example we tested N=100 people, and X=53 of them got the question right, we'd probably be forced to concede that the data are quite consistent with the null hypothesis. On the other hand, if X=99 of our participants got the question right, then we'd feel pretty confident that the null hypothesis is wrong. Similarly, if only X=3 people got the answer right, we'd be similarly confident that the null was wrong. Let's be a little more technical about this: we have a quantity X that we can calculate by looking at our data; after looking at the value of X, we make a decision about whether to believe that the null hypothesis is correct, or to reject the null hypothesis in favour of the alternative. The name for this thing that we calculate to guide our choices is a ***test statistic***.

Having chosen a test statistic, the next step is to state precisely which values of the test statistic would cause is to reject the null hypothesis, and which values would cause us to keep it. In order to do so, we need to determine what the ***sampling distribution of the test statistic*** would be if the null hypothesis were actually true (we talked about sampling distributions earlier in Section 10.3.1). Why do we need this? Because this distribution tells us exactly what values of X our null hypothesis would lead us to expect. And therefore, we can use this distribution as a tool for assessing how closely the null hypothesis agrees with our data.

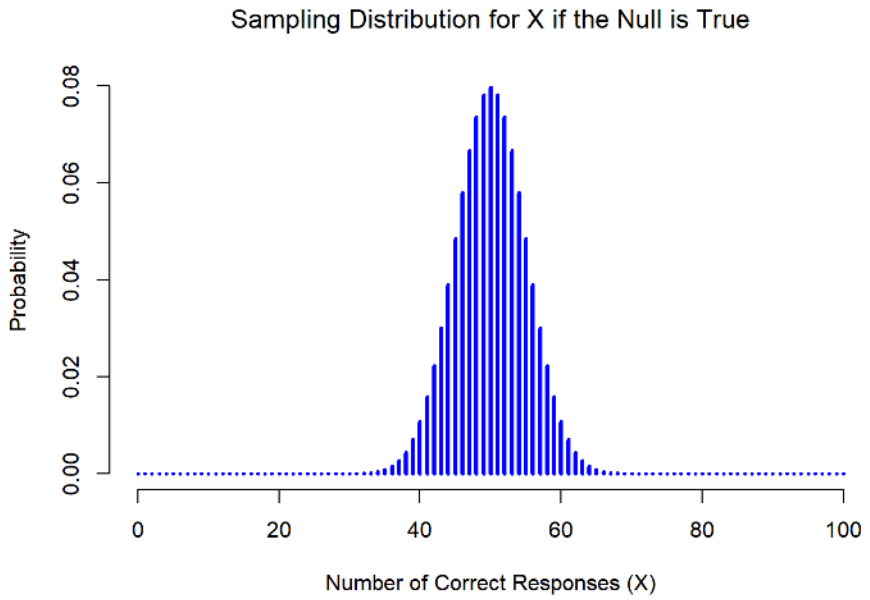


Figure 11.1: The sampling distribution for our test statistic X when the null hypothesis is true. For our ESP scenario, this is a binomial distribution. Not surprisingly, since the null hypothesis says that the probability of a correct response is $\theta=.5$, the sampling distribution says that the most likely value is 50 (our of 100) correct responses. Most of the probability mass lies between 40 and 60.

How do we actually determine the sampling distribution of the test statistic? For a lot of hypothesis tests this step is actually quite complicated, and later on in the book you'll see me being slightly evasive about it for some of the tests (some of them I don't even understand myself). However, sometimes it's very easy. And, fortunately for us, our ESP example provides us with one of the easiest cases. Our population parameter θ is just the overall probability that people respond correctly when asked the question, and our test statistic X is the *count* of the number of people who did so, out of a sample size of N. We've seen a distribution like this before, in Section 9.4: that's exactly what the binomial distribution describes! So, to use the notation and terminology that I introduced in that section, we would say that the null hypothesis predicts that X is binomially distributed, which is written

$$X \sim \text{Binomial}(\theta, N)$$

Since the null hypothesis states that $\theta=0.5$ and our experiment has $N=100$ people, we have the sampling distribution we need. This sampling distribution is plotted in Figure 11.1. No surprises really: the null hypothesis says that $X=50$ is the most likely outcome, and it says that we're almost certain to see somewhere between 40 and 60 correct responses.

This page titled 9.3: Test Statistics and Sampling Distributions is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **11.3: Test Statistics and Sampling Distributions** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

9.4: Making Decisions

Okay, we're very close to being finished. We've constructed a test statistic (X), and we chose this test statistic in such a way that we're pretty confident that if X is close to N/2 then we should retain the null, and if not we should reject it. The question that remains is this: exactly which values of the test statistic should we associate with the null hypothesis, and which exactly values go with the alternative hypothesis? In my ESP study, for example, I've observed a value of X=62. What decision should I make? Should I choose to believe the null hypothesis, or the alternative hypothesis?

9.4.1 Critical regions and critical values

To answer this question, we need to introduce the concept of a ***critical region*** for the test statistic X. The critical region of the test corresponds to those values of X that would lead us to reject null hypothesis (which is why the critical region is also sometimes called the rejection region). How do we find this critical region? Well, let's consider what we know:

- X should be very big or very small in order to reject the null hypothesis.
- If the null hypothesis is true, the sampling distribution of X is Binomial(0.5,N).
- If α=.05, the critical region must cover 5% of this sampling distribution.

It's important to make sure you understand this last point: the critical region corresponds to those values of X for which we would reject the null hypothesis, and the sampling distribution in question describes the probability that we would obtain a particular value of X if the null hypothesis were actually true. Now, let's suppose that we chose a critical region that covers 20% of the sampling distribution, and suppose that the null hypothesis is actually true. What would be the probability of incorrectly rejecting the null? The answer is of course 20%. And therefore, we would have built a test that had an α level of 0.2. If we want α=.05, the critical region is only *allowed* to cover 5% of the sampling distribution of our test statistic.

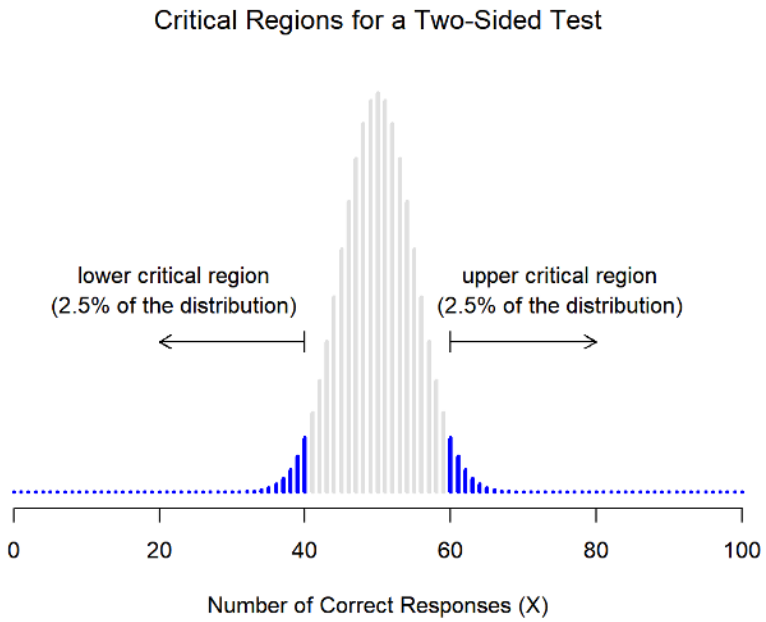


Figure 11.2: The critical region associated with the hypothesis test for the ESP study, for a hypothesis test with a significance level of α=.05. The plot itself shows the sampling distribution of X under the null hypothesis: the grey bars correspond to those values of X for which we would retain the null hypothesis. The black bars show the critical region: those values of X for which we would reject the null. Because the alternative hypothesis is two sided (i.e., allows both θ<.5 and θ>.5), the critical region covers both tails of the distribution. To ensure an α level of .05, we need to ensure that each of the two regions encompasses 2.5% of the sampling distribution.

As it turns out, those three things uniquely solve the problem: our critical region consists of the most *extreme values*, known as the ***tails*** of the distribution. This is illustrated in Figure 11.2. As it turns out, if we want α=.05, then our critical regions correspond to X≤40 and X≥60.[161] That is, if the number of people saying "true" is between 41 and 59, then we should retain the null hypothesis. If the number is between 0 to 40 or between 60 to 100, then we should reject the null hypothesis. The numbers 40 and 60 are often referred to as the ***critical values***, since they define the edges of the critical region.

At this point, our hypothesis test is essentially complete: (1) we choose an α level (e.g., α=.05, (2) come up with some test statistic (e.g., X) that does a good job (in some meaningful sense) of comparing H0 to H1, (3) figure out the sampling distribution of the test statistic on the assumption that the null hypothesis is true (in this case, binomial) and then (4) calculate the critical region that produces an appropriate α level (0-40 and 60-100). All that we have to do now is calculate the value of the test statistic for the real data (e.g., X=62) and then compare it to the critical values to make our decision. Since 62 is greater than the critical value of 60, we would reject the null hypothesis. Or, to phrase it slightly differently, we say that the test has produced a ***significant*** result.

9.4.2 note on statistical "significance"

> *Like other occult techniques of divination, the statistical method has a private jargon deliberately contrived to obscure its methods from non-practitioners.*
>
> *– Attributed to G. O. Ashley*[162]

A very brief digression is in order at this point, regarding the word "significant". The concept of statistical significance is actually a very simple one, but has a very unfortunate name. If the data allow us to reject the null hypothesis, we say that "the result is *statistically significant*", which is often shortened to "the result is significant". This terminology is rather old, and dates back to a time when "significant" just meant something like "indicated", rather than its modern meaning, which is much closer to "important". As a result, a lot of modern readers get very confused when they start learning statistics, because they think that a "significant result" must be an important one. It doesn't mean that at all. All that "statistically significant" means is that the data allowed us to reject a null hypothesis. Whether or not the result is actually important in the real world is a very different question, and depends on all sorts of other things.

9.4.3 difference between one sided and two sided tests

There's one more thing I want to point out about the hypothesis test that I've just constructed. If we take a moment to think about the statistical hypotheses I've been using,

$$H_0: \theta = .5$$

$$H_1: \theta \neq .5$$

we notice that the alternative hypothesis covers *both* the possibility that $\theta < .5$ and the possibility that $\theta > .5$. This makes sense if I really think that ESP could produce better-than-chance performance *or* worse-than-chance performance (and there are some people who think that). In statistical language, this is an example of a ***two-sided test***. It's called this because the alternative hypothesis covers the area on both "sides" of the null hypothesis, and as a consequence the critical region of the test covers both tails of the sampling distribution (2.5% on either side if α=.05), as illustrated earlier in Figure 11.2.

However, that's not the only possibility. It might be the case, for example, that I'm only willing to believe in ESP if it produces better than chance performance. If so, then my alternative hypothesis would only covers the possibility that $\theta > .5$, and as a consequence the null hypothesis now becomes $\theta \leq .5$:

$$H_0: \theta \leq .5$$

$$H_1: \theta > .5$$

When this happens, we have what's called a ***one-sided test***, and when this happens the critical region only covers one tail of the sampling distribution. This is illustrated in Figure 11.3.

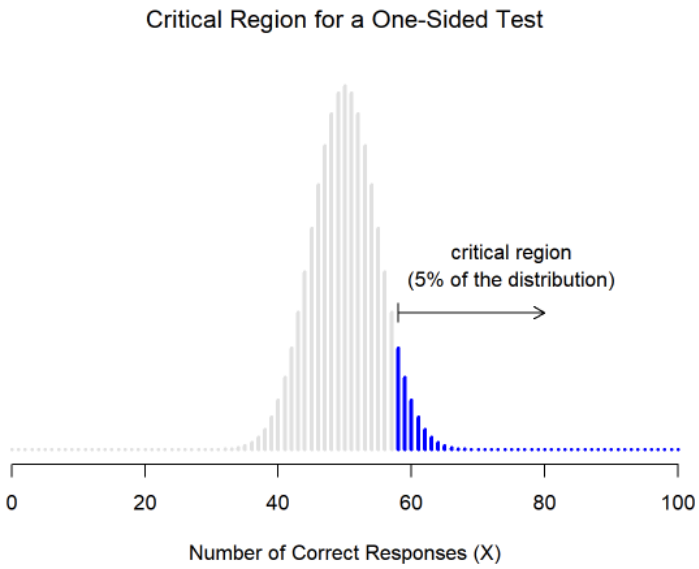


Figure 11.3: The critical region for a one sided test. In this case, the alternative hypothesis is that θ>.05, so we would only reject the null hypothesis for large values of X. As a consequence, the critical region only covers the upper tail of the sampling distribution; specifically the upper 5% of the distribution. Contrast this to the two-sided version earlier)

This page titled 9.4: Making Decisions is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **11.4: Making Decisions** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

9.5: The p value of a test

In one sense, our hypothesis test is complete; we've constructed a test statistic, figured out its sampling distribution if the null hypothesis is true, and then constructed the critical region for the test. Nevertheless, I've actually omitted the most important number of all: ***the p value***. It is to this topic that we now turn. There are two somewhat different ways of interpreting a p value, one proposed by Sir Ronald Fisher and the other by Jerzy Neyman. Both versions are legitimate, though they reflect very different ways of thinking about hypothesis tests. Most introductory textbooks tend to give Fisher's version only, but I think that's a bit of a shame. To my mind, Neyman's version is cleaner, and actually better reflects the logic of the null hypothesis test. You might disagree though, so I've included both. I'll start with Neyman's version…

9.5.1 softer view of decision making

One problem with the hypothesis testing procedure that I've described is that it makes no distinction at all between a result this "barely significant" and those that are "highly significant". For instance, in my ESP study the data I obtained only just fell inside the critical region - so I did get a significant effect, but was a pretty near thing. In contrast, suppose that I'd run a study in which X=97 out of my N=100 participants got the answer right. This would obviously be significant too, but my a much larger margin; there's really no ambiguity about this at all. The procedure that I described makes no distinction between the two. If I adopt the standard convention of allowing α=.05 as my acceptable Type I error rate, then both of these are significant results.

This is where the p value comes in handy. To understand how it works, let's suppose that we ran lots of hypothesis tests on the same data set: but with a different value of α in each case. When we do that for my original ESP data, what we'd get is something like this

Value of α	Reject the null?
0.05	Yes
0.04	Yes
0.03	Yes
0.02	No
0.01	No

When we test ESP data (X=62 successes out of N=100 observations) using α levels of .03 and above, we'd always find ourselves rejecting the null hypothesis. For α levels of .02 and below, we always end up retaining the null hypothesis. Therefore, somewhere between .02 and .03 there must be a smallest value of α that would allow us to reject the null hypothesis for this data. This is the p value; as it turns out the ESP data has p=.021. In short:

p is defined to be the smallest Type I error rate (α) that you have to be willing to tolerate if you want to reject the null hypothesis.

If it turns out that p describes an error rate that you find intolerable, then you must retain the null. If you're comfortable with an error rate equal to p, then it's okay to reject the null hypothesis in favour of your preferred alternative.

In effect, p is a summary of all the possible hypothesis tests that you could have run, taken across all possible α values. And as a consequence it has the effect of "softening" our decision process. For those tests in which $p \leq \alpha$ you would have rejected the null hypothesis, whereas for those tests in which $p > \alpha$ you would have retained the null. In my ESP study I obtained X=62, and as a consequence I've ended up with p=.021. So the error rate I have to tolerate is 2.1%. In contrast, suppose my experiment had yielded X=97. What happens to my p value now? This time it's shrunk to $p=1.36\times10-25$, which is a tiny, tiny^{163} Type I error rate. For this second case I would be able to reject the null hypothesis with a lot more confidence, because I only have to be "willing" to tolerate a type I error rate of about 1 in 10 trillion trillion in order to justify my decision to reject.

9.5.2 probability of extreme data

The second definition of the p-value comes from Sir Ronald Fisher, and it's actually this one that you tend to see in most introductory statistics textbooks. Notice how, when I constructed the critical region, it corresponded to the *tails* (i.e., extreme

values) of the sampling distribution? That's not a coincidence: almost all "good" tests have this characteristic (good in the sense of minimising our type II error rate, β). The reason for that is that a good critical region almost always corresponds to those values of the test statistic that are least likely to be observed if the null hypothesis is true. If this rule is true, then we can define the p-value as the probability that we would have observed a test statistic that is at least as extreme as the one we actually did get. In other words, if the data are extremely implausible according to the null hypothesis, then the null hypothesis is probably wrong.

9.5.3 common mistake

Okay, so you can see that there are two rather different but legitimate ways to interpret the p value, one based on Neyman's approach to hypothesis testing and the other based on Fisher's. Unfortunately, there is a third explanation that people sometimes give, especially when they're first learning statistics, and it is *absolutely and completely wrong*. This mistaken approach is to refer to the p value as "the probability that the null hypothesis is true". It's an intuitively appealing way to think, but it's wrong in two key respects: (1) null hypothesis testing is a frequentist tool, and the frequentist approach to probability does *not* allow you to assign probabilities to the null hypothesis… according to this view of probability, the null hypothesis is either true or it is not; it cannot have a "5% chance" of being true. (2) even within the Bayesian approach, which does let you assign probabilities to hypotheses, the p value would not correspond to the probability that the null is true; this interpretation is entirely inconsistent with the mathematics of how the p value is calculated. Put bluntly, despite the intuitive appeal of thinking this way, there is *no* justification for interpreting a p value this way. Never do it.

This page titled 9.5: The p value of a test is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **11.5: The p value of a test** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

9.6: Reporting the Results of a Hypothesis Test

When writing up the results of a hypothesis test, there's usually several pieces of information that you need to report, but it varies a fair bit from test to test. Throughout the rest of the book I'll spend a little time talking about how to report the results of different tests (see Section 12.1.9 for a particularly detailed example), so that you can get a feel for how it's usually done. However, regardless of what test you're doing, the one thing that you always have to do is say something about the p value, and whether or not the outcome was significant.

The fact that you have to do this is unsurprising; it's the whole point of doing the test. What might be surprising is the fact that there is some contention over exactly how you're supposed to do it. Leaving aside those people who completely disagree with the entire framework underpinning null hypothesis testing, there's a certain amount of tension that exists regarding whether or not to report the exact p value that you obtained, or if you should state only that $p<\alpha$ for a significance level that you chose in advance (e.g., $p<.05$).

issue

To see why this is an issue, the key thing to recognise is that p values are *terribly* convenient. In practice, the fact that we can compute a p value means that we don't actually have to specify any α level at all in order to run the test. Instead, what you can do is calculate your p value and interpret it directly: if you get $p=.062$, then it means that you'd have to be willing to tolerate a Type I error rate of 6.2% to justify rejecting the null. If you personally find 6.2% intolerable, then you retain the null. Therefore, the argument goes, why don't we just report the actual p value and let the reader make up their own minds about what an acceptable Type I error rate is? This approach has the big advantage of "softening" the decision making process – in fact, if you accept the Neyman definition of the p value, that's the whole point of the p value. We no longer have a fixed significance level of $\alpha=.05$ as a bright line separating "accept" from "reject" decisions; and this removes the rather pathological problem of being forced to treat $p=.051$ in a fundamentally different way to $p=.049$.

This flexibility is both the advantage and the disadvantage to the p value. The reason why a lot of people don't like the idea of reporting an exact p value is that it gives the researcher a bit *too much* freedom. In particular, it lets you change your mind about what error tolerance you're willing to put up with *after* you look at the data. For instance, consider my ESP experiment. Suppose I ran my test, and ended up with a p value of .09. Should I accept or reject? Now, to be honest, I haven't yet bothered to think about what level of Type I error I'm "really" willing to accept. I don't have an opinion on that topic. But I *do* have an opinion about whether or not ESP exists, and I *definitely* have an opinion about whether my research should be published in a reputable scientific journal. And amazingly, now that I've looked at the data I'm starting to think that a 9% error rate isn't so bad, especially when compared to how annoying it would be to have to admit to the world that my experiment has failed. So, to avoid looking like I just made it up after the fact, I now say that my α is .1: a 10% type I error rate isn't too bad, and at that level my test is significant! I win.

In other words, the worry here is that I might have the best of intentions, and be the most honest of people, but the temptation to just "shade" things a little bit here and there is really, really strong. As anyone who has ever run an experiment can attest, it's a long and difficult process, and you often get *very* attached to your hypotheses. It's hard to let go and admit the experiment didn't find what you wanted it to find. And that's the danger here. If we use the "raw" p-value, people will start interpreting the data in terms of what they *want* to believe, not what the data are actually saying… and if we allow that, well, why are we bothering to do science at all? Why not let everyone believe whatever they like about anything, regardless of what the facts are? Okay, that's a bit extreme, but that's where the worry comes from. According to this view, you really *must* specify your α value in advance, and then only report whether the test was significant or not. It's the only way to keep ourselves honest.

proposed solutions

In practice, it's pretty rare for a researcher to specify a single α level ahead of time. Instead, the convention is that scientists rely on three standard significance levels: .05, .01 and .001. When reporting your results, you indicate which (if any) of these significance levels allow you to reject the null hypothesis. This is summarised in Table 11.1. This allows us to soften the decision rule a little bit, since $p<.01$ implies that the data meet a stronger evidentiary standard than $p<.05$ would. Nevertheless, since these levels are fixed in advance by convention, it does prevent people choosing their α level after looking at the data.

Table 11.1: A commonly adopted convention for reporting p values: in many places it is conventional to report one of four different things (e.g., p<.05) as shown below. I've included the "significance stars" notation (i.e., a * indicates p<.05) because you sometimes see this notation produced by statistical software. It's also worth noting that some people will write *n.s.* (not significant) rather than p>.05.

Usual notation	Signif. stars	Signif. stars	The null is…
p>.05	NA	The test wasn't significant	Retained
p<.05	*	The test was significant at $= .05 but not at α=.01 or α=.001.$	Rejected
p<.01	**	The test was significant at α=.05 and α=.01 but not at $= .001	Rejected
p<.001	***	The test was significant at all levels	Rejected

Nevertheless, quite a lot of people still prefer to report exact p values. To many people, the advantage of allowing the reader to make up their own mind about how to interpret p=.06 outweighs any disadvantages. In practice, however, even among those researchers who prefer exact p values it is quite common to just write p<.001 instead of reporting an exact value for small p. This is in part because a lot of software doesn't actually print out the p value when it's that small (e.g., SPSS just writes p=.000 whenever p<.001), and in part because a very small p value can be kind of misleading. The human mind sees a number like .0000000001 and it's hard to suppress the gut feeling that the evidence in favour of the alternative hypothesis is a near certainty. In practice however, this is usually wrong. Life is a big, messy, complicated thing: and every statistical test ever invented relies on simplifications, approximations and assumptions. As a consequence, it's probably not reasonable to walk away from *any* statistical analysis with a feeling of confidence stronger than p<.001 implies. In other words, p<.001 is really code for "as far as *this test* is concerned, the evidence is overwhelming."

In light of all this, you might be wondering exactly what you should do. There's a fair bit of contradictory advice on the topic, with some people arguing that you should report the exact p value, and other people arguing that you should use the tiered approach illustrated in Table 11.1. As a result, the best advice I can give is to suggest that you look at papers/reports written in your field and see what the convention seems to be. If there doesn't seem to be any consistent pattern, then use whichever method you prefer.

This page titled 9.6: Reporting the Results of a Hypothesis Test is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **11.6: Reporting the Results of a Hypothesis Test** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

9.7: Running the Hypothesis Test in Practice

At this point some of you might be wondering if this is a "real" hypothesis test, or just a toy example that I made up. It's real. In the previous discussion I built the test from first principles, thinking that it was the simplest possible problem that you might ever encounter in real life. However, this test already exists: it's called the *binomial test*, and it's implemented by an R function called `binom.test()` . To test the null hypothesis that the response probability is one-half `p = .5` ,[164] using data in which `x = 62` of `n = 100` people made the correct response, here's how to do it in R:

```
binom.test( x=62, n=100, p=.5 )
```

```
##
##  Exact binomial test
##
## data:  62 and 100
## number of successes = 62, number of trials = 100, p-value =
## 0.02098
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.5174607 0.7152325
## sample estimates:
## probability of success
##                   0.62
```

Right now, this output looks pretty unfamiliar to you, but you can see that it's telling you more or less the right things. Specifically, the p-value of 0.02 is less than the usual choice of α=.05, so you can reject the null. We'll talk a lot more about how to read this sort of output as we go along; and after a while you'll hopefully find it quite easy to read and understand. For now, however, I just wanted to make the point that R contains a whole lot of functions corresponding to different kinds of hypothesis test. And while I'll usually spend quite a lot of time explaining the logic behind how the tests are built, every time I discuss a hypothesis test the discussion will end with me showing you a fairly simple R command that you can use to run the test in practice.

This page titled 9.7: Running the Hypothesis Test in Practice is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **11.7: Running the Hypothesis Test in Practice** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

9.8: Effect Size, Sample Size and Power

In previous sections I've emphasised the fact that the major design principle behind statistical hypothesis testing is that we try to control our Type I error rate. When we fix α=.05 we are attempting to ensure that only 5% of true null hypotheses are incorrectly rejected. However, this doesn't mean that we don't care about Type II errors. In fact, from the researcher's perspective, the error of failing to reject the null when it is actually false is an extremely annoying one. With that in mind, a secondary goal of hypothesis testing is to try to minimise β, the Type II error rate, although we don't usually *talk* in terms of minimising Type II errors. Instead, we talk about maximising the *power* of the test. Since power is defined as $1-\beta$, this is the same thing.

9.8.1 power function

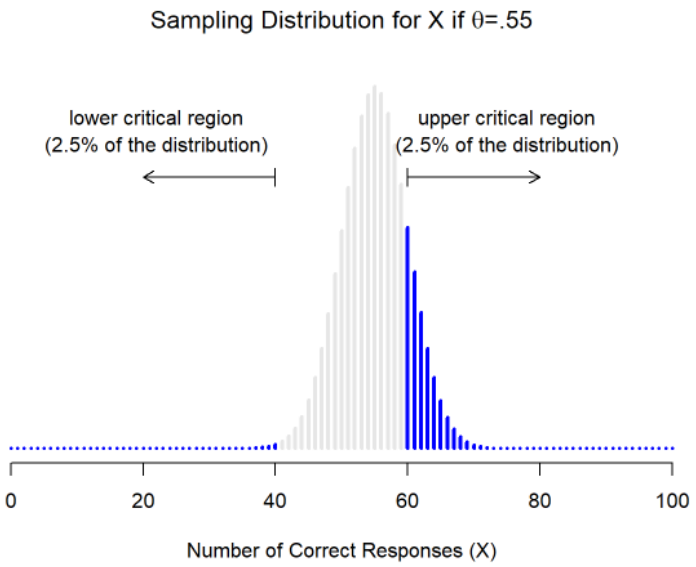


Figure 11.4: Sampling distribution under the *alternative* hypothesis, for a population parameter value of θ=0.55. A reasonable proportion of the distribution lies in the rejection region.

Let's take a moment to think about what a Type II error actually is. A Type II error occurs when the alternative hypothesis is true, but we are nevertheless unable to reject the null hypothesis. Ideally, we'd be able to calculate a single number β that tells us the Type II error rate, in the same way that we can set α=.05 for the Type I error rate. Unfortunately, this is a lot trickier to do. To see this, notice that in my ESP study the alternative hypothesis actually corresponds to lots of possible values of θ. In fact, the alternative hypothesis corresponds to every value of θ *except* 0.5. Let's suppose that the true probability of someone choosing the correct response is 55% (i.e., θ=.55). If so, then the *true* sampling distribution for X is not the same one that the null hypothesis predicts: the most likely value for X is now 55 out of 100. Not only that, the whole sampling distribution has now shifted, as shown in Figure 11.4. The critical regions, of course, do not change: by definition, the critical regions are based on what the null hypothesis predicts. What we're seeing in this figure is the fact that when the null hypothesis is wrong, a much larger proportion of the sampling distribution distribution falls in the critical region. And of course that's what should happen: the probability of rejecting the null hypothesis is larger when the null hypothesis is actually false! However θ=.55 is not the only possibility consistent with the alternative hypothesis. Let's instead suppose that the true value of θ is actually 0.7. What happens to the sampling distribution when this occurs? The answer, shown in Figure 11.5, is that almost the entirety of the sampling distribution has now moved into the critical region. Therefore, if θ=0.7 the probability of us correctly rejecting the null hypothesis (i.e., the power of the test) is much larger than if θ=0.55. In short, while θ=.55 and θ=.70 are both part of the alternative hypothesis, the Type II error rate is different.

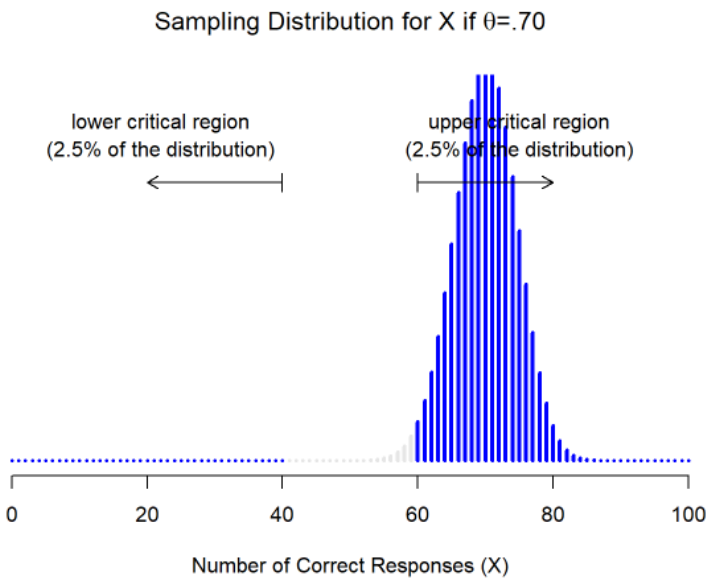


Figure 11.5: Sampling distribution under the *alternative* hypothesis, for a population parameter value of θ=0.70. Almost all of the distribution lies in the rejection region.

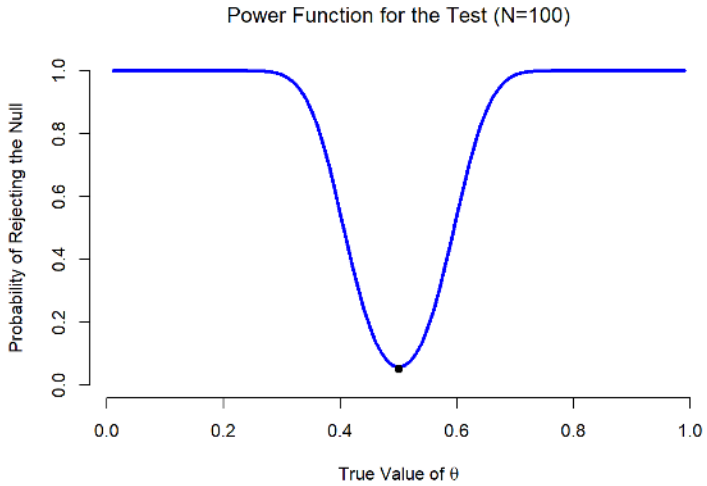


Figure 11.6: The probability that we will reject the null hypothesis, plotted as a function of the true value of θ. Obviously, the test is more powerful (greater chance of correct rejection) if the true value of θ is very different from the value that the null hypothesis specifies (i.e., θ=.5). Notice that when θ actually is equal to .5 (plotted as a black dot), the null hypothesis is in fact true: rejecting the null hypothesis in this instance would be a Type I error.

What all this means is that the power of a test (i.e., 1−β) depends on the true value of θ. To illustrate this, I've calculated the expected probability of rejecting the null hypothesis for all values of θ, and plotted it in Figure 11.6. This plot describes what is usually called the ***power function*** of the test. It's a nice summary of how good the test is, because it actually tells you the power (1−β) for all possible values of θ. As you can see, when the true value of θ is very close to 0.5, the power of the test drops very sharply, but when it is further away, the power is large.

9.8.2 Effect size

> *Since all models are wrong the scientist must be alert to what is importantly wrong. It is inappropriate to be concerned with mice when there are tigers abroad*
>
> *– George Box 1976*

The plot shown in Figure 11.6 captures a fairly basic point about hypothesis testing. If the true state of the world is very different from what the null hypothesis predicts, then your power will be very high; but if the true state of the world is similar to the null (but not identical) then the power of the test is going to be very low. Therefore, it's useful to be able to have some way of quantifying how "similar" the true state of the world is to the null hypothesis. A statistic that does this is called a measure of ***effect size*** (e.g. Cohen 1988; Ellis 2010). Effect size is defined slightly differently in different contexts,[165] (and so this section just talks in general terms) but the qualitative idea that it tries to capture is always the same: how big is the difference between the *true* population parameters, and the parameter values that are assumed by the null hypothesis? In our ESP example, if we let θ_0=0.5 denote the value assumed by the null hypothesis, and let θ denote the true value, then a simple measure of effect size could be something like the difference between the true value and null (i.e., $\theta-\theta_0$), or possibly just the magnitude of this difference, abs($\theta-\theta_0$).

	big effect size	small effect size
significant result	difference is real, and of practical importance	difference is real, but might not be interesting
non-significant result	no effect observed	no effect observed

Why calculate effect size? Let's assume that you've run your experiment, collected the data, and gotten a significant effect when you ran your hypothesis test. Isn't it enough just to say that you've gotten a significant effect? Surely that's the *point* of hypothesis testing? Well, sort of. Yes, the point of doing a hypothesis test is to try to demonstrate that the null hypothesis is wrong, but that's hardly the only thing we're interested in. If the null hypothesis claimed that θ=.5, and we show that it's wrong, we've only really told half of the story. Rejecting the null hypothesis implies that we believe that $\theta\neq$.5, but there's a big difference between θ=.51 and θ=.8. If we find that θ=.8, then not only have we found that the null hypothesis is wrong, it appears to be *very* wrong. On the other hand, suppose we've successfully rejected the null hypothesis, but it looks like the true value of θ is only .51 (this would only be possible with a large study). Sure, the null hypothesis is wrong, but it's not at all clear that we actually *care*, because the effect size is so small. In the context of my ESP study we might still care, since any demonstration of real psychic powers would actually be pretty cool[166], but in other contexts a 1% difference isn't very interesting, even if it is a real difference. For instance, suppose we're looking at differences in high school exam scores between males and females, and it turns out that the female scores are 1% higher on average than the males. If I've got data from thousands of students, then this difference will almost certainly be *statistically significant*, but regardless of how small the p value is it's just not very interesting. You'd hardly want to go around proclaiming a crisis in boys education on the basis of such a tiny difference would you? It's for this reason that it is becoming more standard (slowly, but surely) to report some kind of standard measure of effect size along with the the results of the hypothesis test. The hypothesis test itself tells you whether you should believe that the effect you have observed is real (i.e., not just due to chance); the effect size tells you whether or not you should care.

9.8.3 Increasing the power of your study

Not surprisingly, scientists are fairly obsessed with maximising the power of their experiments. We want our experiments to work, and so we want to maximise the chance of rejecting the null hypothesis if it is false (and of course we usually want to believe that it is false!) As we've seen, one factor that influences power is the effect size. So the first thing you can do to increase your power is to increase the effect size. In practice, what this means is that you want to design your study in such a way that the effect size gets magnified. For instance, in my ESP study I might believe that psychic powers work best in a quiet, darkened room; with fewer distractions to cloud the mind. Therefore I would try to conduct my experiments in just such an environment: if I can strengthen people's ESP abilities somehow, then the true value of θ will go up[167] and therefore my effect size will be larger. In short, clever experimental design is one way to boost power; because it can alter the effect size.

Unfortunately, it's often the case that even with the best of experimental designs you may have only a small effect. Perhaps, for example, ESP really does exist, but even under the best of conditions it's very very weak. Under those circumstances, your best bet for increasing power is to increase the sample size. In general, the more observations that you have available, the more likely it is that you can discriminate between two hypotheses. If I ran my ESP experiment with 10 participants, and 7 of them correctly guessed the colour of the hidden card, you wouldn't be terribly impressed. But if I ran it with 10,000 participants and 7,000 of them got the answer right, you would be much more likely to think I had discovered something. In other words, power increases with the

sample size. This is illustrated in Figure 11.7, which shows the power of the test for a true parameter of θ=0.7, for all sample sizes N from 1 to 100, where I'm assuming that the null hypothesis predicts that θ_0=0.5.

```
##   [1] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.11837800
##   [7] 0.08257300 0.05771362 0.19643626 0.14945203 0.11303734 0.25302172
##  [13] 0.20255096 0.16086106 0.29695959 0.24588947 0.38879291 0.33269435
##  [19] 0.28223844 0.41641377 0.36272868 0.31341925 0.43996501 0.38859619
##  [25] 0.51186665 0.46049782 0.41129777 0.52752694 0.47870819 0.58881596
##  [31] 0.54162450 0.49507894 0.59933871 0.55446069 0.65155826 0.60907715
##  [37] 0.69828554 0.65867614 0.61815357 0.70325017 0.66542910 0.74296156
##  [43] 0.70807163 0.77808343 0.74621569 0.71275488 0.78009449 0.74946571
##  [49] 0.81000236 0.78219322 0.83626633 0.81119597 0.78435605 0.83676444
##  [55] 0.81250680 0.85920268 0.83741123 0.87881491 0.85934395 0.83818214
##  [61] 0.87858194 0.85962510 0.89539581 0.87849413 0.91004390 0.89503851
##  [67] 0.92276845 0.90949768 0.89480727 0.92209753 0.90907263 0.93304809
##  [73] 0.92153987 0.94254237 0.93240638 0.92108426 0.94185449 0.93185881
##  [79] 0.95005094 0.94125189 0.95714694 0.94942195 0.96327866 0.95651332
##  [85] 0.94886329 0.96265653 0.95594208 0.96796884 0.96208909 0.97255504
##  [91] 0.96741721 0.97650832 0.97202770 0.97991117 0.97601093 0.97153910
##  [97] 0.97944717 0.97554675 0.98240749 0.97901142
```

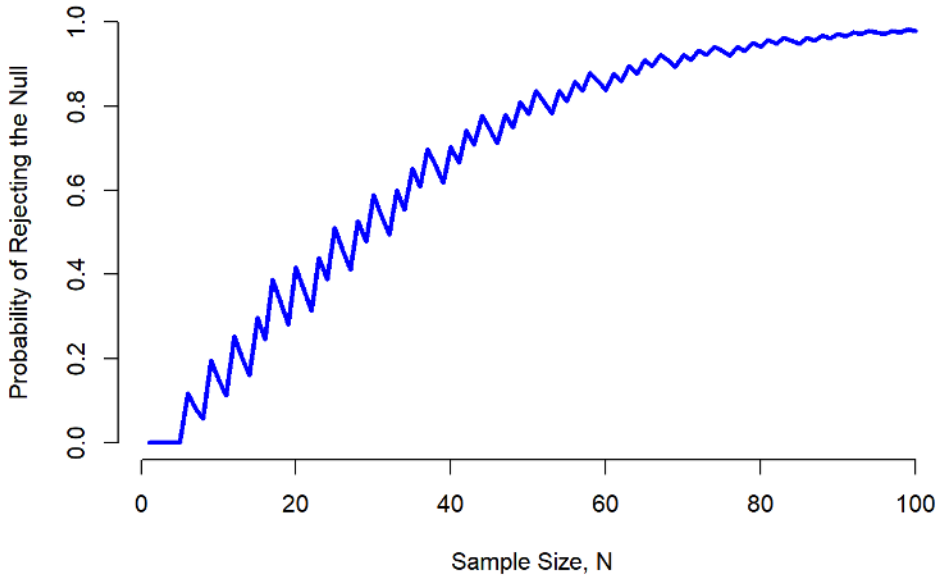


Figure 11.7: The power of our test, plotted as a function of the sample size N. In this case, the true value of θ is 0.7, but the null hypothesis is that θ=0.5. Overall, larger N means greater power. (The small zig-zags in this function occur because of some odd interactions between θ, α and the fact that the binomial distribution is discrete; it doesn't matter for any serious purpose)

Because power is important, whenever you're contemplating running an experiment it would be pretty useful to know how much power you're likely to have. It's never possible to know for sure, since you can't possibly know what your effect size is. However, it's often (well, sometimes) possible to guess how big it should be. If so, you can guess what sample size you need! This idea is called ***power analysis***, and if it's feasible to do it, then it's very helpful, since it can tell you something about whether you have enough time or money to be able to run the experiment successfully. It's increasingly common to see people arguing that power analysis should be a required part of experimental design, so it's worth knowing about. I don't discuss power analysis in this book, however. This is partly for a boring reason and partly for a substantive one. The boring reason is that I haven't had time to write about power analysis yet. The substantive one is that I'm still a little suspicious of power analysis. Speaking as a researcher, I have very rarely found myself in a position to be able to do one – it's either the case that (a) my experiment is a bit non-standard and I don't know how to define effect size properly, (b) I literally have so little idea about what the effect size will be that I wouldn't

know how to interpret the answers. Not only that, after extensive conversations with someone who does stats consulting for a living (my wife, as it happens), I can't help but notice that in practice the *only* time anyone ever asks her for a power analysis is when she's helping someone write a grant application. In other words, the only time any scientist ever seems to want a power analysis in real life is when they're being forced to do it by bureaucratic process. It's not part of anyone's day to day work. In short, I've always been of the view that while power is an important concept, power *analysis* is not as useful as people make it sound, except in the rare cases where (a) someone has figured out how to calculate power for your actual experimental design and (b) you have a pretty good idea what the effect size is likely to be. Maybe other people have had better experiences than me, but I've personally never been in a situation where both (a) and (b) were true. Maybe I'll be convinced otherwise in the future, and probably a future version of this book would include a more detailed discussion of power analysis, but for now this is about as much as I'm comfortable saying about the topic.

This page titled 9.8: Effect Size, Sample Size and Power is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **11.8: Effect Size, Sample Size and Power** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

9.9: Some Issues to Consider

What I've described to you in this chapter is the orthodox framework for null hypothesis significance testing (NHST). Understanding how NHST works is an absolute necessity, since it has been the dominant approach to inferential statistics ever since it came to prominence in the early 20th century. It's what the vast majority of working scientists rely on for their data analysis, so even if you hate it you need to know it. However, the approach is not without problems. There are a number of quirks in the framework, historical oddities in how it came to be, theoretical disputes over whether or not the framework is right, and a lot of practical traps for the unwary. I'm not going to go into a lot of detail on this topic, but I think it's worth briefly discussing a few of these issues.

9.9.1 Neyman versus Fisher

The first thing you should be aware of is that orthodox NHST is actually a mash-up of two rather different approaches to hypothesis testing, one proposed by Sir Ronald Fisher and the other proposed by Jerzy Neyman (for a historical summary see Lehmann 2011). The history is messy because Fisher and Neyman were real people whose opinions changed over time, and at no point did either of them offer "the definitive statement" of how we should interpret their work many decades later. That said, here's a quick summary of what I take these two approaches to be.

First, let's talk about Fisher's approach. As far as I can tell, Fisher assumed that you only had the one hypothesis (the null), and what you want to do is find out if the null hypothesis is inconsistent with the data. From his perspective, what you should do is check to see if the data are "sufficiently unlikely" according to the null. In fact, if you remember back to our earlier discussion, that's how Fisher defines the p-value. According to Fisher, if the null hypothesis provided a very poor account of the data, you could safely reject it. But, since you don't have any other hypotheses to compare it to, there's no way of "accepting the alternative" because you don't necessarily have an explicitly stated alternative. That's more or less all that there was to it.

In contrast, Neyman thought that the point of hypothesis testing was as a guide to action, and his approach was somewhat more formal than Fisher's. His view was that there are multiple things that you could *do* (accept the null or accept the alternative) and the point of the test was to tell you which one the data support. From this perspective, it is critical to specify your alternative hypothesis properly. If you don't know what the alternative hypothesis is, then you don't know how powerful the test is, or even which action makes sense. His framework genuinely requires a competition between different hypotheses. For Neyman, the p value didn't directly measure the probability of the data (or data more extreme) under the null, it was more of an abstract description about which "possible tests" were telling you to accept the null, and which "possible tests" were telling you to accept the alternative.

As you can see, what we have today is an odd mishmash of the two. We talk about having both a null hypothesis and an alternative (Neyman), but usually[168] define the p value in terms of exreme data (Fisher), but we still have α values (Neyman). Some of the statistical tests have explicitly specified alternatives (Neyman) but others are quite vague about it (Fisher). And, according to some people at least, we're not allowed to talk about accepting the alternative (Fisher). It's a mess: but I hope this at least explains why it's a mess.

9.9.2 Bayesians versus frequentists

Earlier on in this chapter I was quite emphatic about the fact that you *cannot* interpret the p value as the probability that the null hypothesis is true. NHST is fundamentally a frequentist tool (see Chapter 9) and as such it does not allow you to assign probabilities to hypotheses: the null hypothesis is either true or it is not. The Bayesian approach to statistics interprets probability as a degree of belief, so it's totally okay to say that there is a 10% chance that the null hypothesis is true: that's just a reflection of the degree of confidence that you have in this hypothesis. You aren't allowed to do this within the frequentist approach. Remember, if you're a frequentist, a probability can only be defined in terms of what happens after a large number of independent replications (i.e., a long run frequency). If this is your interpretation of probability, talking about the "probability" that the null hypothesis is true is complete gibberish: a null hypothesis is either true or it is false. There's no way you can talk about a long run frequency for this statement. To talk about "the probability of the null hypothesis" is as meaningless as "the colour of freedom". It doesn't have one!

Most importantly, this *isn't* a purely ideological matter. If you decide that you are a Bayesian and that you're okay with making probability statements about hypotheses, you have to follow the Bayesian rules for calculating those probabilities. I'll talk more about this in Chapter 17, but for now what I want to point out to you is the p value is a *terrible* approximation to the probability that H_0 is true. If what you want to know is the probability of the null, then the p value is not what you're looking for!

9.9.3 Traps

As you can see, the theory behind hypothesis testing is a mess, and even now there are arguments in statistics about how it "should" work. However, disagreements among statisticians are not our real concern here. Our real concern is practical data analysis. And while the "orthodox" approach to null hypothesis significance testing has many drawbacks, even an unrepentant Bayesian like myself would agree that they can be useful if used responsibly. Most of the time they give sensible answers, and you can use them to learn interesting things. Setting aside the various ideologies and historical confusions that we've discussed, the fact remains that the biggest danger in all of statistics is *thoughtlessness*. I don't mean stupidity, here: I literally mean thoughtlessness. The rush to interpret a result without spending time thinking through what each test actually says about the data, and checking whether that's consistent with how you've interpreted it. That's where the biggest trap lies.

To give an example of this, consider the following example (see Gelman and Stern 2006). Suppose I'm running my ESP study, and I've decided to analyse the data separately for the male participants and the female participants. Of the male participants, 33 out of 50 guessed the colour of the card correctly. This is a significant effect (p=.03). Of the female participants, 29 out of 50 guessed correctly. This is not a significant effect (p=.32). Upon observing this, it is extremely tempting for people to start wondering why there is a difference between males and females in terms of their psychic abilities. However, this is wrong. If you think about it, we haven't *actually* run a test that explicitly compares males to females. All we have done is compare males to chance (binomial test was significant) and compared females to chance (binomial test was non significant). If we want to argue that there is a real difference between the males and the females, we should probably run a test of the null hypothesis that there is no difference! We can do that using a different hypothesis test,[169] but when we do that it turns out that we have no evidence that males and females are significantly different (p=.54). *Now* do you think that there's anything fundamentally different between the two groups? Of course not. What's happened here is that the data from both groups (male and female) are pretty borderline: by pure chance, one of them happened to end up on the magic side of the p=.05 line, and the other one didn't. That doesn't actually imply that males and females are different. This mistake is so common that you should always be wary of it: the difference between significant and not-significant is *not* evidence of a real difference – if you want to say that there's a difference between two groups, then you have to test for that difference!

The example above is just that: an example. I've singled it out because it's such a common one, but the bigger picture is that data analysis can be tricky to get right. Think about *what* it is you want to test, *why* you want to test it, and whether or not the answers that your test gives could possibly make any sense in the real world.

This page titled 9.9: Some Issues to Consider is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **11.9: Some Issues to Consider** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

9.10: Misconceptions of Hypothesis Testing

Learning Objectives

- State why the probability value is not the probability the null hypothesis is false
- Explain why a low probability value does not necessarily mean there is a large effect
- Explain why a non-significant outcome does not mean the null hypothesis is probably true

Misconceptions about significance testing are common. This section lists three important ones.

1. Misconception: The probability value is the probability that the null hypothesis is false.

   Proper interpretation: The probability value is the probability of a result as extreme or more extreme given that the null hypothesis is true. It is the probability of the data given the null hypothesis. It is not the probability that the null hypothesis is false.
2. Misconception: A low probability value indicates a large effect.

   Proper interpretation: A low probability value indicates that the sample outcome (or one more extreme) would be very unlikely if the null hypothesis were true. A low probability value can occur with small effect sizes, particularly if the sample size is large.
3. Misconception: A non-significant outcome means that the null hypothesis is probably true.

   Proper interpretation: A non-significant outcome means that the data do not conclusively demonstrate that the null hypothesis is false.

This page titled 9.10: Misconceptions of Hypothesis Testing is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **11.9: Misconceptions of Hypothesis Testing** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

9.11: Summary

Null hypothesis testing is one of the most ubiquitous elements to statistical theory. The vast majority of scientific papers report the results of some hypothesis test or another. As a consequence it is almost impossible to get by in science without having at least a cursory understanding of what a p-value means, making this one of the most important chapters in the book. As usual, I'll end the chapter with a quick recap of the key ideas that we've talked about:

- Research hypotheses and statistical hypotheses. Null and alternative hypotheses. (Section 11.1).
- Type 1 and Type 2 errors (Section 11.2)
- Test statistics and sampling distributions (Section 11.3)
- Hypothesis testing as a decision making process (Section 11.4)
- p-values as "soft" decisions (Section 11.5)
- Writing up the results of a hypothesis test (Section 11.6)
- Effect size and power (Section 11.8)
- A few issues to consider regarding hypothesis testing (Section 11.9)

Later in the book, in Chapter 17, I'll revisit the theory of null hypothesis tests from a Bayesian perspective, and introduce a number of new tools that you can use if you aren't particularly fond of the orthodox approach. But for now, though, we're done with the abstract statistical theory, and we can start discussing specific data analysis tools.

References

Cohen, J. 1988. *Statistical Power Analysis for the Behavioral Sciences*. 2nd ed. Lawrence Erlbaum.

Ellis, P. D. 2010. *The Essential Guide to Effect Sizes: Statistical Power, Meta-Analysis, and the Interpretation of Research Results*. Cambridge, UK: Cambridge University Press.

Lehmann, Erich L. 2011. *Fisher, Neyman, and the Creation of Classical Statistics*. Springer.

Gelman, A., and H. Stern. 2006. "The Difference Between 'Significant' and 'Not Significant' Is Not Itself Statistically Significant." *The American Statistician* 60: 328–31.

156. The quote comes from Wittgenstein's (1922) text, *Tractatus Logico-Philosphicus*.
157. A technical note. The description below differs subtly from the standard description given in a lot of introductory texts. The orthodox theory of null hypothesis testing emerged from the work of Sir Ronald Fisher and Jerzy Neyman in the early 20th century; but Fisher and Neyman actually had very different views about how it should work. The standard treatment of hypothesis testing that most texts use is a hybrid of the two approaches. The treatment here is a little more Neyman-style than the orthodox view, especially as regards the meaning of the p value.
158. My apologies to anyone who actually believes in this stuff, but on my reading of the literature on ESP, it's just not reasonable to think this is real. To be fair, though, some of the studies are rigorously designed; so it's actually an interesting area for thinking about psychological research design. And of course it's a free country, so you can spend your own time and effort proving me wrong if you like, but I wouldn't think that's a terribly practical use of your intellect.
159. This analogy only works if you're from an adversarial legal system like UK/US/Australia. As I understand these things, the French inquisitorial system is quite different.
160. An aside regarding the language you use to talk about hypothesis testing. Firstly, one thing you really want to avoid is the word "prove": a statistical test really doesn't *prove* that a hypothesis is true or false. Proof implies certainty, and as the saying goes, statistics means never having to say you're certain. On that point almost everyone would agree. However, beyond that there's a fair amount of confusion. Some people argue that you're only allowed to make statements like "rejected the null", "failed to reject the null", or possibly "retained the null". According to this line of thinking, you can't say things like "accept the alternative" or "accept the null". Personally I think this is too strong: in my opinion, this conflates null hypothesis testing with Karl Popper's falsificationist view of the scientific process. While there are similarities between falsificationism and null hypothesis testing, they aren't equivalent. However, while I personally think it's fine to talk about accepting a hypothesis (on the proviso that "acceptance" doesn't actually mean that it's necessarily true, especially in the case of the null hypothesis),

many people will disagree. And more to the point, you should be aware that this particular weirdness exists, so that you're not caught unawares by it when writing up your own results.

161. Strictly speaking, the test I just constructed has α=.057, which is a bit too generous. However, if I'd chosen 39 and 61 to be the boundaries for the critical region, then the critical region only covers 3.5% of the distribution. I figured that it makes more sense to use 40 and 60 as my critical values, and be willing to tolerate a 5.7% type I error rate, since that's as close as I can get to a value of α=.05.
162. The internet seems fairly convinced that Ashley said this, though I can't for the life of me find anyone willing to give a source for the claim.
163. That's p=.00000000000000000000000136 for folks that don't like scientific notation!
164. Note that the `p` here has nothing to do with a p value. The `p` argument in the `binom.test()` function corresponds to the probability of making a correct response, according to the null hypothesis. In other words, it's the θ value.
165. There's an R package called `compute.es` that can be used for calculating a very broad range of effect size measures; but for the purposes of the current book we won't need it: all of the effect size measures that I'll talk about here have functions in the `lsr` package
166. Although in practice a very small effect size is worrying, because even very minor methodological flaws might be responsible for the effect; and in practice no experiment is perfect, so there are always methodological issues to worry about.
167. Notice that the true population parameter θ doesn't necessarily correspond to an immutable fact of nature. In this context θ is just the true probability that people would correctly guess the colour of the card in the other room. As such the population parameter can be influenced by all sorts of things. Of course, this is all on the assumption that ESP actually exists!
168. Although this book describes both Neyman's and Fisher's definition of the p value, most don't. Most introductory textbooks will only give you the Fisher version.
169. In this case, the Pearson chi-square test of independence (Chapter 12; `chisq.test()` in R) is what we use; see also the `prop.test()` function.

This page titled 9.11: Summary is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **11.10: Summary** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

9.12: Statistical Literacy

Learning Objectives

- Evidence for the Higgs Boson

Research in March, 2012 reported here found evidence for the existence of the Higgs Boson particle. However, the evidence for the existence of the particle was not statistically significant.

Example 9.12.1: what do you think?

Did the researchers conclude that their investigation had been a failure or did they conclude they have evidence of the particle, just not strong enough evidence to draw a confident conclusion?

Solution

One of the investigators stated, "We see some tantalizing evidence but not significant enough to make a stronger statement." Therefore, they were encouraged by the result. In a subsequent study, the evidence was significant.

This page titled 9.12: Statistical Literacy is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **11.10: Statistical Literacy** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

9.13: Logic of Hypothesis Testing (Exercises)

You may want to use the Binomial Calculator for some of these exercises.

General Questions

Q1

An experiment is conducted to test the claim that James Bond can taste the difference between a Martini that is shaken and one that is stirred. What is the null hypothesis? (relevant section)

Q2

The following explanation is incorrect. What three words should be added to make it correct? (relevant section)

> The probability value is the probability of obtaining a statistic as different from the parameter specified in the null hypothesis as the statistic obtained in the experiment. The probability value is computed assuming that the null hypothesis is true.

Q3

Why do experimenters test hypotheses they think are false? (relevant section)

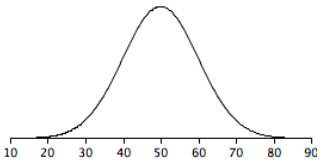
Q4

State the null hypothesis for:

a. An experiment testing whether echinacea decreases the length of colds.
b. A correlational study on the relationship between brain size and intelligence.
c. An investigation of whether a self-proclaimed psychic can predict the outcome of a coin flip.
d. A study comparing a drug with a placebo on the amount of pain relief. (A one-tailed test was used.)
(relevant section & relevant section)

Q5

Assume the null hypothesis is that $\mu = 50$ and that the graph shown below is the sampling distribution of the mean (M). Would a sample value of $M = 60$ be significant in a two-tailed test at the 0.05 level? Roughly what value of M would be needed to be significant? (relevant section & relevant section)



Q6

A researcher develops a new theory that predicts that vegetarians will have more of a particular vitamin in their blood than non-vegetarians. An experiment is conducted and vegetarians do have more of the vitamin, but the difference is not significant. The probability value is 0.13. Should the experimenter's confidence in the theory increase, decrease, or stay the same? (relevant section)

Q7

A researcher hypothesizes that the lowering in cholesterol associated with weight loss is really due to exercise. To test this, the researcher carefully controls for exercise while comparing the cholesterol levels of a group of subjects who lose weight by dieting with a control group that does not diet. The difference between groups in cholesterol is not significant. Can the researcher claim that weight loss has no effect? (relevant section)

Q8

A significance test is performed and $p = 0.20$. Why can't the experimenter claim that the probability that the null hypothesis is true is 0.20? (relevant section, relevant section & relevant section)

Q9

For a drug to be approved by the FDA, the drug must be shown to be safe and effective. If the drug is significantly more effective than a placebo, then the drug is deemed effective. What do you know about the effectiveness of a drug once it has been approved by the FDA (assuming that there has not been a **Type I** error)? (relevant section)

Q10

When is it valid to use a one-tailed test? What is the advantage of a one-tailed test? Give an example of a null hypothesis that would be tested by a one-tailed test. (relevant section)

Q11

Distinguish between probability value and significance level. (relevant section)

Q12

Suppose a study was conducted on the effectiveness of a class on "How to take tests." The SAT scores of an experimental group and a control group were compared. (There were 100 subjects in each group.) The mean score of the experimental group was 503 and the mean score of the control group was 499. The difference between means was found to be significant, $p = 0.037$. What do you conclude about the effectiveness of the class? (relevant section & relevant section)

Q13

Is it more conservative to use an alpha level of 0.01 or an alpha level of 0.05? Would beta be higher for an alpha of 0.05 or for an alpha of 0.01? (relevant section)

Q14

Why is $H_o : M_1 = M_2$ not a proper null hypothesis? (relevant section)

Q15

An experimenter expects an effect to come out in a certain direction. Is this sufficient basis for using a one-tailed test? Why or why not? (relevant section)

Q16

How do the **Type I** and **Type II** error rates of one-tailed and two-tailed tests differ? (relevant section & relevant section)

Q17

A two-tailed probability is 0.03. What is the one-tailed probability if the effect were in the specified direction? What would it be if the effect were in the other direction? (relevant section)

Q18

You choose an alpha level of 0.01 and then analyze your data.

a. What is the probability that you will make a **Type I** error given that the null hypothesis is true?
b. What is the probability that you will make a **Type I** error given that the null hypothesis is false? (relevant section)

Q19

Why doesn't it make sense to test the hypothesis that the sample mean is 42? (relevant section & relevant section)

Q20

True/false: It is easier to reject the null hypothesis if the researcher uses a smaller alpha (α) level. (relevant section & relevant section)

Q21

True/false: You are more likely to make a Type I error when using a small sample than when using a large sample. (relevant section)

Q22

True/false: You accept the alternative hypothesis when you reject the null hypothesis. (relevant section)

Q23

True/false: You do not accept the null hypothesis when you fail to reject it. (relevant section)

Q24

True/false: A researcher risks making a **Type I** error any time the null hypothesis is rejected. (relevant section)

This page titled 9.13: Logic of Hypothesis Testing (Exercises) is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **11.E: Logic of Hypothesis Testing (Exercises)** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

CHAPTER OVERVIEW

10: Categorical Data Analysis

Now that we've got the basic theory behind hypothesis testing, it's time to start looking at specific tests that are commonly used in psychology. So where should we start? Not every textbook agrees on where to start, but I'm going to start with "χ^2 tests" (this chapter) and "t-tests" (Chapter 13). Both of these tools are very frequently used in scientific practice, and while they're not as powerful as "analysis of variance" (Chapter 14) and "regression" (Chapter 15) they're much easier to understand.

The term "categorical data" is just another name for "nominal scale data". It's nothing that we haven't already discussed, it's just that in the context of data analysis people tend to use the term "categorical data" rather than "nominal scale data". I don't know why. In any case, ***categorical data analysis*** refers to a collection of tools that you can use when your data are nominal scale. However, there are a lot of different tools that can be used for categorical data analysis, and this chapter only covers a few of the more common ones.

10.1: The χ2 Goodness-of-fit Test
10.2: The χ2 test of independence (or association)
10.3: The Continuity Correction
10.4: Effect Size
10.5: Assumptions of the Test(s)
10.6: The Most Typical Way to Do Chi-square Tests in R
10.7: The Fisher Exact Test
10.8: The McNemar Test
10.9: What's the Difference Between McNemar and Independence?
10.10: Summary
10.11: Statistical Literacy
10.12: Chi Square (Exercises)

This page titled 10: Categorical Data Analysis is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

10.1: The χ2 Goodness-of-fit Test

The χ^2 goodness-of-fit test is one of the oldest hypothesis tests around: it was invented by Karl Pearson around the turn of the century (Pearson 1900), with some corrections made later by Sir Ronald Fisher (Fisher 1922a). To introduce the statistical problem that it addresses, let's start with some psychology…

10.1.1 cards data

Over the years, there have been a lot of studies showing that humans have a lot of difficulties in simulating randomness. Try as we might to "act" random, we *think* in terms of patterns and structure, and so when asked to "do something at random", what people actually do is anything but random. As a consequence, the study of human randomness (or non-randomness, as the case may be) opens up a lot of deep psychological questions about how we think about the world. With this in mind, let's consider a very simple study. Suppose I asked people to imagine a shuffled deck of cards, and mentally pick one card from this imaginary deck "at random". After they've chosen one card, I ask them to mentally select a second one. For both choices, what we're going to look at is the suit (hearts, clubs, spades or diamonds) that people chose. After asking, say, N=200 people to do this, I'd like to look at the data and figure out whether or not the cards that people pretended to select were really random. The data are contained in the `randomness.Rdata` file, which contains a single data frame called `cards`. Let's take a look:

```
library( lsr )
load( "./rbook-master/data/randomness.Rdata" )
str(cards)
```

```
## 'data.frame':    200 obs. of  3 variables:
##  $ id      : Factor w/ 200 levels "subj1","subj10",..: 1 112 124 135 146 157 168 1
##  $ choice_1: Factor w/ 4 levels "clubs","diamonds",..: 4 2 3 4 3 1 3 2 4 2 ...
##  $ choice_2: Factor w/ 4 levels "clubs","diamonds",..: 1 1 1 1 4 3 2 1 1 4 ...
```

As you can see, the `cards` data frame contains three variables, an `id` variable that assigns a unique identifier to each participant, and the two variables `choice_1` and `choice_2` that indicate the card suits that people chose. Here's the first few entries in the data frame:

```
head( cards )
```

```
##      id choice_1 choice_2
## 1 subj1   spades    clubs
## 2 subj2 diamonds    clubs
## 3 subj3   hearts    clubs
## 4 subj4   spades    clubs
## 5 subj5   hearts   spades
## 6 subj6    clubs   hearts
```

For the moment, let's just focus on the first choice that people made. We'll use the `table()` function to count the number of times that we observed people choosing each suit. I'll save the table to a variable called `observed`, for reasons that will become clear very soon:

```
observed <- table( cards$choice_1 )
observed
```

```
##
##    clubs diamonds   hearts   spades
##       35       51       64       50
```

That little frequency table is quite helpful. Looking at it, there's a bit of a hint that people *might* be more likely to select hearts than clubs, but it's not completely obvious just from looking at it whether that's really true, or if this is just due to chance. So we'll probably have to do some kind of statistical analysis to find out, which is what I'm going to talk about in the next section.

Excellent. From this point on, we'll treat this table as the data that we're looking to analyse. However, since I'm going to have to talk about this data in mathematical terms (sorry!) it might be a good idea to be clear about what the notation is. In R, if I wanted to pull out the number of people that selected diamonds, I could do it by name by typing `observed["diamonds"]` but, since `"diamonds"` is second element of the `observed` vector, it's equally effective to refer to it as `observed[2]` . The mathematical notation for this is pretty similar, except that we shorten the human-readable word "observed" to the letter O, and we use subscripts rather than brackets: so the second observation in our table is written as `observed[2]` in R, and is written as O_2 in maths. The relationship between the English descriptions, the R commands, and the mathematical symbols are illustrated below:

label	index i	math. symbol	R command	the value
clubs ♣	1	O1	`observed[1]`	35
diamonds ♢	2	O2	`observed[2]`	51
hearts ♡	3	O3	`observed[3]`	64
spades ♠	4	O4	`observed[4]`	50

Hopefully that's pretty clear. It's also worth nothing that mathematicians prefer to talk about things in general rather than specific things, so you'll also see the notation O_i, which refers to the number of observations that fall within the i-th category (where i could be 1, 2, 3 or 4). Finally, if we want to refer to the set of all observed frequencies, statisticians group all of observed values into a vector, which I'll refer to as O.

$$O=(O_1,O_2,O_3,O_4)$$

Again, there's nothing new or interesting here: it's just notation. If I say that O = (35,51,64,50) all I'm doing is describing the table of observed frequencies (i.e., `observed`), but I'm referring to it using mathematical notation, rather than by referring to an R variable.

10.1.2 null hypothesis and the alternative hypothesis

As the last section indicated, our research hypothesis is that "people don't choose cards randomly". What we're going to want to do now is translate this into some statistical hypotheses, and construct a statistical test of those hypotheses. The test that I'm going to describe to you is ***Pearson's χ^2 goodness of fit test***, and as is so often the case, we have to begin by carefully constructing our null hypothesis. In this case, it's pretty easy. First, let's state the null hypothesis in words:

H_0
All four suits are chosen with equal probability

Now, because this is statistics, we have to be able to say the same thing in a mathematical way. To do this, let's use the notation P_j to refer to the true probability that the j-th suit is chosen. If the null hypothesis is true, then each of the four suits has a 25% chance of being selected: in other words, our null hypothesis claims that P_1=.25, P2=.25, P3=.25 and finally that P_4=.25. However, in the same way that we can group our observed frequencies into a vector O that summarises the entire data set, we can use P to refer to the probabilities that correspond to our null hypothesis. So if I let the vector P=(P_1,P_2,P_3,P_4) refer to the collection of probabilities that describe our null hypothesis, then we have

$$H_0:P=(.25,.25,.25,.25)$$

In this particular instance, our null hypothesis corresponds to a vector of probabilities P in which all of the probabilities are equal to one another. But this doesn't have to be the case. For instance, if the experimental task was for people to imagine they were drawing from a deck that had twice as many clubs as any other suit, then the null hypothesis would correspond to something like P=(.4,.2,.2,.2). As long as the probabilities are all positive numbers, and they all sum to 1, them it's a perfectly legitimate choice for the null hypothesis. However, the most common use of the goodness of fit test is to test a null hypothesis that all of the categories are equally likely, so we'll stick to that for our example.

What about our alternative hypothesis, H_1? All we're really interested in is demonstrating that the probabilities involved aren't all identical (that is, people's choices weren't completely random). As a consequence, the "human friendly" versions of our hypotheses look like this:

H_0	H_1
All four suits are chosen with equal probability	At least one of the suit-choice probabilities *isn't* .25

H_0	H_1
P=(.25,.25,.25,.25)	P≠(.25,.25,.25,.25)

Conveniently, the mathematical version of the hypotheses looks quite similar to an R command defining a vector. So maybe what I should do is store the P vector in R as well, since we're almost certainly going to need it later. And because I'm so imaginative, I'll call this R vector `probabilities` ,

```
probabilities <- c(clubs = .25, diamonds = .25, hearts = .25, spades = .25)
probabilities
```

```
##    clubs diamonds   hearts   spades
##     0.25     0.25     0.25     0.25
```

10.1.3 "goodness of fit" test statistic

At this point, we have our observed frequencies O and a collection of probabilities P corresponding the null hypothesis that we want to test. We've stored these in R as the corresponding variables `observed` and `probabilities` . What we now want to do is construct a test of the null hypothesis. As always, if we want to test H_0 against H_1, we're going to need a test statistic. The basic trick that a goodness of fit test uses is to construct a test statistic that measures how "close" the data are to the null hypothesis. If the data don't resemble what you'd "expect" to see if the null hypothesis were true, then it probably isn't true. Okay, if the null hypothesis were true, what would we expect to see? Or, to use the correct terminology, what are the ***expected frequencies***. There are N=200 observations, and (if the null is true) the probability of any one of them choosing a heart is P_3=.25, so I guess we're expecting 200×.25=50 hearts, right? Or, more specifically, if we let E_i refer to "the number of category i responses that we're expecting if the null is true", then

$$E_i = N \times P_i$$

This is pretty easy to calculate in R:

```
N <- 200  # sample size
expected <- N * probabilities # expected frequencies
expected
```

```
##    clubs diamonds   hearts   spades
##       50       50       50       50
```

None of which is very surprising: if there are 200 observation that can fall into four categories, and we think that all four categories are equally likely, then on average we'd expect to see 50 observations in each category, right?

Now, how do we translate this into a test statistic? Clearly, what we want to do is compare the *expected* number of observations in each category (E_i) with the *observed* number of observations in that category (O_i). And on the basis of this comparison, we ought to be able to come up with a good test statistic. To start with, let's calculate the difference between what the null hypothesis expected us to find and what we actually did find. That is, we calculate the "observed minus expected" difference score, $O_i - E_i$. This is illustrated in the following table.

		♣	♢	♡	♠
expected frequency	E_i	50	50	50	50
observed frequency	O_i	35	51	64	50
difference score	$O_i - E_i$	-15	1	14	0

The same calculations can be done in R, using our `expected` and `observed` variables:

```
observed - expected
```

```
##
##    clubs diamonds   hearts   spades
##      -15        1       14        0
```

Regardless of whether we do the calculations by hand or whether we do them in R, it's clear that people chose more hearts and fewer clubs than the null hypothesis predicted. However, a moment's thought suggests that these raw differences aren't quite what we're looking for. Intuitively, it feels like it's just as bad when the null hypothesis predicts too few observations (which is what happened with hearts) as it is when it predicts too many (which is what happened with clubs). So it's a bit weird that we have a negative number for clubs and a positive number for heards. One easy way to fix this is to square everything, so that we now calculate the squared differences, $(E_i - O_i)^2$. As before, we could do this by hand, but it's easier to do it in R…

```
(observed - expected)^2
```

```
##
##    clubs diamonds   hearts   spades
##      225        1      196        0
```

Now we're making progress. What we've got now is a collection of numbers that are big whenever the null hypothesis makes a bad prediction (clubs and hearts), but are small whenever it makes a good one (diamonds and spades). Next, for some technical reasons that I'll explain in a moment, let's also divide all these numbers by the expected frequency E_i, so we're actually calculating $\frac{(E_i - O_i)^2}{E_i}$. Since E_i=50 for all categories in our example, it's not a very interesting calculation, but let's do it anyway. The R command becomes:

```
(observed - expected)^2 / expected
```

```
##
##    clubs diamonds   hearts   spades
##     4.50     0.02     3.92     0.00
```

In effect, what we've got here are four different "error" scores, each one telling us how big a "mistake" the null hypothesis made when we tried to use it to predict our observed frequencies. So, in order to convert this into a useful test statistic, one thing we could do is just add these numbers up. The result is called the ***goodness of fit*** statistic, conventionally referred to either as X^2 or GOF. We can calculate it using this command in R

```
sum( (observed - expected)^2 / expected )
```

```
## [1] 8.44
```

The formula for this statistic looks remarkably similar to the R command. If we let k refer to the total number of categories (i.e., k=4 for our cards data), then the X^2 statistic is given by:

$$X^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

Intuitively, it's clear that if X^2 is small, then the observed data O_i are very close to what the null hypothesis predicted E_i, so we're going to need a large X^2 statistic in order to reject the null. As we've seen from our calculations, in our cards data set we've got a value of X^2=8.44. So now the question becomes, is this a big enough value to reject the null?

10.1.4 sampling distribution of the GOF statistic (advanced)

To determine whether or not a particular value of X^2 is large enough to justify rejecting the null hypothesis, we're going to need to figure out what the sampling distribution for X^2 would be if the null hypothesis were true. So that's what I'm going to do in this section. I'll show you in a fair amount of detail how this sampling distribution is constructed, and then – in the next section – use it to build up a hypothesis test. If you want to cut to the chase and are willing to take it on faith that the sampling distribution is a ***chi-squared (χ^2) distribution*** with k−1 degrees of freedom, you can skip the rest of this section. However, if you want to understand *why* the goodness of fit test works the way it does, read on…

Okay, let's suppose that the null hypothesis is actually true. If so, then the true probability that an observation falls in the i-th category is P_i – after all, that's pretty much the definition of our null hypothesis. Let's think about what this actually means. If you think about it, this is kind of like saying that "nature" makes the decision about whether or not the observation ends up in category i by flipping a weighted coin (i.e., one where the probability of getting a head is P_j). And therefore, we can think of our observed frequency O_i by imagining that nature flipped N of these coins (one for each observation in the data set)… and exactly O_i of them came up heads. Obviously, this is a pretty weird way to think about the experiment. But what it does (I hope) is remind you that we've actually seen this scenario before. It's exactly the same set up that gave rise to the binomial distribution in Section 9.4. In other words, if the null hypothesis is true, then it follows that our observed frequencies were generated by sampling from a binomial distribution:

$$O_i \sim \text{Binomial}(P_i, N)$$

Now, if you remember from our discussion of the central limit theorem (Section 10.3.3), the binomial distribution starts to look pretty much identical to the normal distribution, especially when N is large and when P_i isn't *too* close to 0 or 1. In other words as long as $N \times P_i$ is large enough – or, to put it another way, when the expected frequency E_i is large enough – the theoretical distribution of O_i is approximately normal. Better yet, if O_i is normally distributed, then so is $(O_i - E_i)/\sqrt{E_i}$ … since E_i is a fixed value, subtracting off E_i and dividing by $\sqrt{E_i}$ changes the mean and standard deviation of the normal distribution; but that's all it does. Okay, so now let's have a look at what our goodness of fit statistic actually *is*. What we're doing is taking a bunch of things that are normally-distributed, squaring them, and adding them up. Wait. We've seen that before too! As we discussed in Section 9.6, when you take a bunch of things that have a standard normal distribution (i.e., mean 0 and standard deviation 1), square them, then add them up, then the resulting quantity has a chi-square distribution. So now we know that the null hypothesis predicts that the sampling distribution of the goodness of fit statistic is a chi-square distribution. Cool.

There's one last detail to talk about, namely the degrees of freedom. If you remember back to Section 9.6, I said that if the number of things you're adding up is k, then the degrees of freedom for the resulting chi-square distribution is k. Yet, what I said at the start of this section is that the actual degrees of freedom for the chi-square goodness of fit test is k−1. What's up with that? The answer

here is that what we're supposed to be looking at is the number of genuinely *independent* things that are getting added together. And, as I'll go on to talk about in the next section, even though there's k things that we're adding, only k−1 of them are truly independent; and so the degrees of freedom is actually only k−1. That's the topic of the next section.[170]

10.1.5 Degrees of freedom

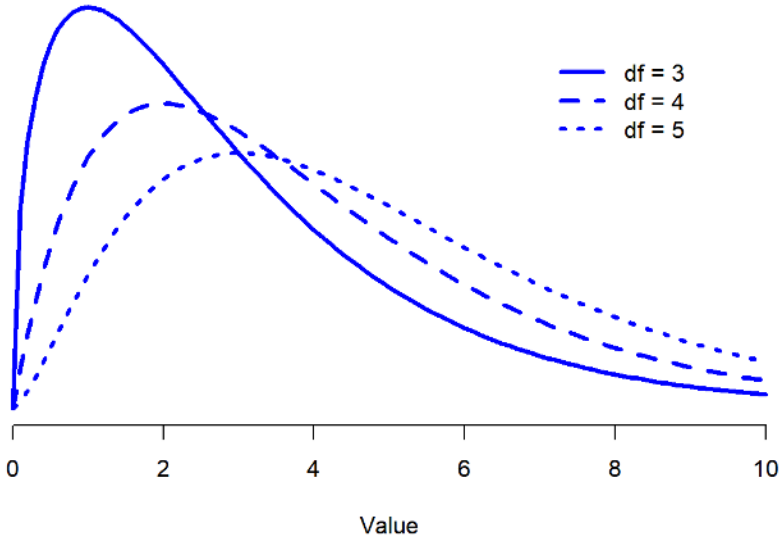


Figure 12.1: Chi-square distributions with different values for the "degrees of freedom".

When I introduced the chi-square distribution in Section 9.6, I was a bit vague about what "***degrees of freedom***" actually *means*. Obviously, it matters: looking Figure 12.1 you can see that if we change the degrees of freedom, then the chi-square distribution changes shape quite substantially. But what exactly *is* it? Again, when I introduced the distribution and explained its relationship to the normal distribution, I did offer an answer… it's the number of "normally distributed variables" that I'm squaring and adding together. But, for most people, that's kind of abstract, and not entirely helpful. What we really need to do is try to understand degrees of freedom in terms of our data. So here goes.

The basic idea behind degrees of freedom is quite simple: you calculate it by counting up the number of distinct "quantities" that are used to describe your data; and then subtracting off all of the "constraints" that those data must satisfy.[171] This is a bit vague, so let's use our cards data as a concrete example. We describe out data using four numbers, O_1, O_2, O_3 and O_4 corresponding to the observed frequencies of the four different categories (hearts, clubs, diamonds, spades). These four numbers are the *random outcomes* of our experiment. But, my experiment actually has a fixed constraint built into it: the sample size N.[172] That is, if we know how many people chose hearts, how many chose diamonds and how many chose clubs; then we'd be able to figure out exactly how many chose spades. In other words, although our data are described using four numbers, they only actually correspond to 4−1=3 degrees of freedom. A slightly different way of thinking about it is to notice that there are four *probabilities* that we're interested in (again, corresponding to the four different categories), but these probabilities must sum to one, which imposes a constraint. Therefore, the degrees of freedom is 4−1=3. Regardless of whether you want to think about it in terms of the observed frequencies or in terms of the probabilities, the answer is the same. In general, when running the chi-square goodness of fit test for an experiment involving k groups, then the degrees of freedom will be k−1.

10.1.6 Testing the null hypothesis

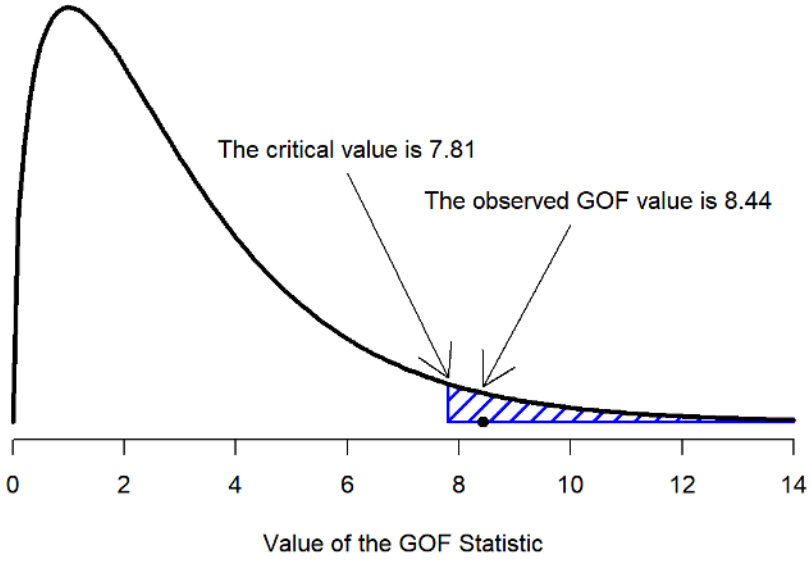


Figure 12.2: Illustration of how the hypothesis testing works for the chi-square goodness of fit test.

The final step in the process of constructing our hypothesis test is to figure out what the rejection region is. That is, what values of X^2 would lead is to reject the null hypothesis. As we saw earlier, large values of X^2 imply that the null hypothesis has done a poor job of predicting the data from our experiment, whereas small values of X^2 imply that it's actually done pretty well. Therefore, a pretty sensible strategy would be to say there is some critical value, such that if X^2 is bigger than the critical value we reject the null; but if X^2 is smaller than this value we retain the null. In other words, to use the language we introduced in Chapter @ref(hypothesistesting the chi-squared goodness of fit test is always a ***one-sided test***. Right, so all we have to do is figure out what this critical value is. And it's pretty straightforward. If we want our test to have significance level of α=.05 (that is, we are willing to tolerate a Type I error rate of 5%), then we have to choose our critical value so that there is only a 5% chance that X^2 could get to be that big if the null hypothesis is true. That is to say, we want the 95th percentile of the sampling distribution. This is illustrated in Figure 12.2.

Ah, but – I hear you ask – how do I calculate the 95th percentile of a chi-squared distribution with k−1 degrees of freedom? If only R had some function, called… oh, I don't know, `qchisq()` … that would let you calculate this percentile (see Chapter 9 if you've forgotten). Like this…

```
qchisq( p = .95, df = 3 )
```

```
## [1] 7.814728
```

So if our X^2 statistic is bigger than 7.81 or so, then we can reject the null hypothesis. Since we actually calculated that before (i.e., X^2=8.44) we can reject the null. If we want an exact p-value, we can calculate it using the `pchisq()` function:

```
pchisq( q = 8.44, df = 3, lower.tail = FALSE )
```

```
## [1] 0.03774185
```

This is hopefully pretty straightforward, as long as you recall that the "`p`" form of the probability distribution functions in R always calculates the probability of getting a value of *less* than the value you entered (in this case 8.44). We want the opposite: the probability of getting a value of 8.44 or *more*. That's why I told R to use the upper tail, not the lower tail. That said, it's usually easier to calculate the p-value this way:

```
1-pchisq( q = 8.44, df = 3 )
```

```
## [1] 0.03774185
```

So, in this case we would reject the null hypothesis, since p<.05. And that's it, basically. You now know "Pearson's χ^2 test for the goodness of fit". Lucky you.

10.1.7 Doing the test in R

Gosh darn it. Although we did manage to do everything in R as we were going through that little example, it does rather feel as if we're typing too many things into the magic computing box. And I *hate* typing. Not surprisingly, R provides a function that will do all of these calculations for you. In fact, there are several different ways of doing it. The one that most people use is the `chisq.test()` function, which comes with every installation of R. I'll show you how to use the `chisq.test()` function later on (in Section @ref(chisq.test), but to start out with I'm going to show you the `goodnessOfFitTest()` function in the `lsr` package, because it produces output that I think is easier for beginners to understand. It's pretty straightforward: our raw data are stored in the variable `cards$choice_1`, right? If you want to test the null hypothesis that all four suits are equally likely, then (assuming you have the `lsr` package loaded) all you have to do is type this:

```
goodnessOfFitTest( cards$choice_1 )
```

```
##
##      Chi-square test against specified probabilities
##
## Data variable:   cards$choice_1
##
## Hypotheses:
##    null:        true probabilities are as specified
##    alternative: true probabilities differ from those specified
##
## Descriptives:
##          observed freq. expected freq. specified prob.
## clubs               35             50            0.25
## diamonds            51             50            0.25
## hearts              64             50            0.25
## spades              50             50            0.25
##
## Test results:
##    X-squared statistic:  8.44
##    degrees of freedom:  3
##    p-value:  0.038
```

R then runs the test, and prints several lines of text. I'll go through the output line by line, so that you can make sure that you understand what you're looking at. The first two lines are just telling you things you already know:

```
Chi-square test against specified probabilities

Data variable:   cards$choice_1
```

The first line tells us what kind of hypothesis test we ran, and the second line tells us the name of the variable that we ran it on. After that comes a statement of what the null and alternative hypotheses are:

```
Hypotheses:
   null:        true probabilities are as specified
   alternative: true probabilities differ from those specified
```

For a beginner, it's kind of handy to have this as part of the output: it's a nice reminder of what your null and alternative hypotheses are. Don't get used to seeing this though. The vast majority of hypothesis tests in R aren't so kind to novices. Most R functions are written on the assumption that you already understand the statistical tool that you're using, so they don't bother to include an explicit statement of the null and alternative hypothesis. The only reason that `goodnessOfFitTest()` actually does give you this is that I wrote it with novices in mind.

The next part of the output shows you the comparison between the observed frequencies and the expected frequencies:

```
Descriptives:
         observed freq. expected freq. specified prob.
clubs               35             50            0.25
diamonds            51             50            0.25
hearts              64             50            0.25
spades              50             50            0.25
```

The first column shows what the observed frequencies were, the second column shows the expected frequencies according to the null hypothesis, and the third column shows you what the probabilities actually were according to the null. For novice users, I think this is helpful: you can look at this part of the output and check that it makes sense: if it doesn't you might have typed something incorrecrtly.

The last part of the output is the "important" stuff: it's the result of the hypothesis test itself. There are three key numbers that need to be reported: the value of the X^2 statistic, the degrees of freedom, and the p-value:

```
Test results:
   X-squared statistic:  8.44
   degrees of freedom:  3
   p-value:  0.038
```

Notice that these are the same numbers that we came up with when doing the calculations the long way.

10.1.8 Specifying a different null hypothesis

At this point you might be wondering what to do if you want to run a goodness of fit test, but your null hypothesis is *not* that all categories are equally likely. For instance, let's suppose that someone had made the theoretical prediction that people should choose red cards 60% of the time, and black cards 40% of the time (I've no idea why you'd predict that), but had no other preferences. If that were the case, the null hypothesis would be to expect 30% of the choices to be hearts, 30% to be diamonds, 20% to be spades and 20% to be clubs. This seems like a silly theory to me, and it's pretty easy to test it using our data. All we need to do is specify the probabilities associated with the null hypothesis. We create a vector like this:

```
nullProbs <- c(clubs = .2, diamonds = .3, hearts = .3, spades = .2)
nullProbs
```

```
##    clubs diamonds   hearts   spades
##      0.2      0.3      0.3      0.2
```

Now that we have an explicitly specified null hypothesis, we include it in our command. This time round I'll use the argument names properly. The data variable corresponds to the argument `x`, and the probabilities according to the null hypothesis correspond to the argument `p`. So our command is:

```
goodnessOfFitTest( x = cards$choice_1, p = nullProbs )
```

```
##
##      Chi-square test against specified probabilities
##
## Data variable:   cards$choice_1
##
## Hypotheses:
##    null:        true probabilities are as specified
##    alternative: true probabilities differ from those specified
##
## Descriptives:
##          observed freq. expected freq. specified prob.
## clubs               35             40             0.2
## diamonds            51             60             0.3
## hearts              64             60             0.3
## spades              50             40             0.2
##
## Test results:
##    X-squared statistic:  4.742
##    degrees of freedom:  3
##    p-value:  0.192
```

As you can see the null hypothesis and the expected frequencies are different to what they were last time. As a consequence our X^2 test statistic is different, and our p-value is different too. Annoyingly, the p-value is .192, so we can't reject the null hypothesis. Sadly, despite the fact that the null hypothesis corresponds to a very silly theory, these data don't provide enough evidence against it.

10.1.9 report the results of the test

So now you know how the test works, and you know how to do the test using a wonderful magic computing box. The next thing you need to know is how to write up the results. After all, there's no point in designing and running an experiment and then analysing the data if you don't tell anyone about it! So let's now talk about what you need to do when reporting your analysis. Let's stick with our card-suits example. If I wanted to write this result up for a paper or something, the conventional way to report this would be to write something like this:

Of the 200 participants in the experiment, 64 selected hearts for their first choice, 51 selected diamonds, 50 selected spades, and 35 selected clubs. A chi-square goodness of fit test was conducted to test whether the choice probabilities were identical for all four suits. The results were significant ($\chi^2(3)=8.44, p<.05$), suggesting that people did not select suits purely at random.

This is pretty straightforward, and hopefully it seems pretty unremarkable. That said, there's a few things that you should note about this description:

- *The statistical test is preceded by the descriptive statistics*. That is, I told the reader something about what the data look like before going on to do the test. In general, this is good practice: always remember that your reader doesn't know your data anywhere near as well as you do. So unless you describe it to them properly, the statistical tests won't make any sense to them, and they'll get frustrated and cry.

- *The description tells you what the null hypothesis being tested is*. To be honest, writers don't always do this, but it's often a good idea in those situations where some ambiguity exists; or when you can't rely on your readership being intimately familiar with the statistical tools that you're using. Quite often the reader might not know (or remember) all the details of the test that your using, so it's a kind of politeness to "remind" them! As far as the goodness of fit test goes, you can usually rely on a scientific audience knowing how it works (since it's covered in most intro stats classes). However, it's still a good idea to be explicit about stating the null hypothesis (briefly!) because the null hypothesis can be different depending on what you're using the test for. For instance, in the cards example my null hypothesis was that all the four suit probabilities were identical (i.e., $P_1=P_2=P_3=P_4=0.25$), but there's nothing special about that hypothesis. I could just as easily have tested the null hypothesis that $P_1=0.7$ and $P_2=P_3=P_4=0.1$ using a goodness of fit test. So it's helpful to the reader if you explain to them what your null hypothesis was. Also, notice that I described the null hypothesis in words, not in maths. That's perfectly acceptable. You can describe it in maths if you like, but since most readers find words easier to read than symbols, most writers tend to describe the null using words if they can.
- *A "stat block" is included*. When reporting the results of the test itself, I didn't just say that the result was significant, I included a "stat block" (i.e., the dense mathematical-looking part in the parentheses), which reports all the "raw" statistical data. For the chi-square goodness of fit test, the information that gets reported is the test statistic (that the goodness of fit statistic was 8.44), the information about the distribution used in the test (χ^2 with 3 degrees of freedom, which is usually shortened to $\chi^2(3)$), and then the information about whether the result was significant (in this case $p<.05$). The particular information that needs to go into the stat block is different for every test, and so each time I introduce a new test I'll show you what the stat block should look like.[173] However the general principle is that you should always provide enough information so that the reader could check the test results themselves if they really wanted to.
- *The results are interpreted*. In addition to indicating that the result was significant, I provided an interpretation of the result (i.e., that people didn't choose randomly). This is also a kindness to the reader, because it tells them something about what they should believe about what's going on in your data. If you don't include something like this, it's really hard for your reader to understand what's going on.[174]

As with everything else, your overriding concern should be that you *explain* things to your reader. Always remember that the point of reporting your results is to communicate to another human being. I cannot tell you just how many times I've seen the results section of a report or a thesis or even a scientific article that is just gibberish, because the writer has focused solely on making sure they've included all the numbers, and forgotten to actually communicate with the human reader.

10.1.10 comment on statistical notation (advanced)

> *Satan delights equally in statistics and in quoting scripture*
>
> *– H.G. Wells*

If you've been reading very closely, and are as much of a mathematical pedant as I am, there is one thing about the way I wrote up the chi-square test in the last section that might be bugging you a little bit. There's something that feels a bit wrong with writing "$\chi^2(3)=8.44$", you might be thinking. After all, it's the goodness of fit statistic that is equal to 8.44, so shouldn't I have written $X^2=8.44$ or maybe GOF=8.44? This seems to be conflating the *sampling distribution* (i.e., χ^2 with df=3) with the *test statistic* (i.e., X^2). Odds are you figured it was a typo, since χ and X look pretty similar. Oddly, it's not. Writing $\chi^2(3)=8.44$ is essentially a highly condensed way of writing "the sampling distribution of the test statistic is $\chi^2(3)$, and the value of the test statistic is 8.44".

In one sense, this is kind of stupid. There are *lots* of different test statistics out there that turn out to have a chi-square sampling distribution: the X^2 statistic that we've used for our goodness of fit test is only one of many (albeit one of the most commonly encountered ones). In a sensible, perfectly organised world, we'd *always* have a separate name for the test statistic and the sampling distribution: that way, the stat block itself would tell you exactly what it was that the researcher had calculated. Sometimes this happens. For instance, the test statistic used in the Pearson goodness of fit test is written X^2; but there's a closely related test known as the G-test[175] , in which the test statistic is written as G. As it happens, the Pearson goodness of fit test and the G-test both test the same null hypothesis; and the sampling distribution is exactly the same (i.e., chi-square with k−1 degrees of freedom). If I'd done a G-test for the cards data rather than a goodness of fit test, then I'd have ended up with a test statistic of G=8.65, which is slightly different from the $X^2=8.44$ value that I got earlier; and produces a slightly smaller p-value of p=.034. Suppose that the convention was to report the test statistic, then the sampling distribution, and then the p-value. If that were true,

then these two situations would produce different stat blocks: my original result would be written X^2=8.44, χ^2(3),p=.038, whereas the new version using the G-test would be written as G=8.65, χ^2(3), p=.034. However, using the condensed reporting standard, the original result is written χ^2(3)=8.44, p=.038, and the new one is written χ^2(3)=8.65, p=.034, and so it's actually unclear which test I actually ran.

So why don't we live in a world in which the contents of the stat block uniquely specifies what tests were ran? The deep reason is that life is messy. We (as users of statistical tools) want it to be nice and neat and organised… we want it to be *designed*, as if it were a product. But that's not how life works: statistics is an intellectual discipline just as much as any other one, and as such it's a massively distributed, partly-collaborative and partly-competitive project that no-one really understands completely. The things that you and I use as data analysis tools weren't created by an Act of the Gods of Statistics; they were invented by lots of different people, published as papers in academic journals, implemented, corrected and modified by lots of other people, and then explained to students in textbooks by someone else. As a consequence, there's a *lot* of test statistics that don't even have names; and as a consequence they're just given the same name as the corresponding sampling distribution. As we'll see later, any test statistic that follows a χ^2 distribution is commonly called a "chi-square statistic"; anything that follows a t-distribution is called a "t-statistic" and so on. But, as the X^2 versus G example illustrates, two different things with the same sampling distribution are still, well, different.

As a consequence, it's sometimes a good idea to be clear about what the actual test was that you ran, especially if you're doing something unusual. If you just say "chi-square test", it's not actually clear what test you're talking about. Although, since the two most common chi-square tests are the goodness of fit test and the independence test (Section 12.2), most readers with stats training can probably guess. Nevertheless, it's something to be aware of.

This page titled 10.1: The χ2 Goodness-of-fit Test is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **12.1: The χ2 Goodness-of-fit Test** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

10.2: The χ2 test of independence (or association)

GUARDBOT1:	Halt!
GUARDBOT2:	Be you robot or human?
LEELA:	Robot…we be.
FRY:	Uh, yup! Just two robots out roboting it up! Eh?
GUARDBOT1:	Administer the test.
GUARDBOT2:	Which of the following would you most prefer? A: A puppy, B: A pretty flower from your sweetie, or C: A large properly-formatted data file?
GUARDBOT1:	Choose!

– Futurama, "Fear of a Bot Planet

The other day I was watching an animated documentary examining the quaint customs of the natives of the planet *Chapek 9*. Apparently, in order to gain access to their capital city, a visitor must prove that they're a robot, not a human. In order to determine whether or not visitor is human, they ask whether the visitor prefers puppies, flowers or large, properly formatted data files. "Pretty clever," I thought to myself "but what if humans and robots have the same preferences? That probably wouldn't be a very good test then, would it?" As it happens, I got my hands on the testing data that the civil authorities of *Chapek 9* used to check this. It turns out that what they did was very simple… they found a bunch of robots and a bunch of humans and asked them what they preferred. I saved their data in a file called `chapek9.Rdata`, which I can now load and have a quick look at:

```
load( "./rbook-master/data/chapek9.Rdata" )
str(chapek9)
```

```
## 'data.frame':    180 obs. of  2 variables:
##  $ species: Factor w/ 2 levels "robot","human": 1 2 2 2 1 2 2 1 2 1 ...
##  $ choice : Factor w/ 3 levels "puppy","flower",..: 2 3 3 3 3 2 3 3 1 2 ...
```

Okay, so we have a single data frame called `chapek9`, which contains two factors, `species` and `choice`. As always, it's nice to have a quick look at the data,

```
head(chapek9)
```

```
##   species choice
## 1   robot flower
## 2   human   data
## 3   human   data
## 4   human   data
## 5   robot   data
## 6   human flower
```

and then take a `summary()`,

```
summary(chapek9)
```

```
##   species      choice
##  robot:87   puppy : 28
##  human:93   flower: 43
##             data  :109
```

In total there are 180 entries in the data frame, one for each person (counting both robots and humans as "people") who was asked to make a choice. Specifically, there's 93 humans and 87 robots; and overwhelmingly the preferred choice is the data file. However, these summaries don't address the question we're interested in. To do that, we need a more detailed description of the data. What we want to do is look at the `choices` broken down *by* `species`. That is, we need to cross-tabulate the data (see Section 7.1). There's quite a few ways to do this, as we've seen, but since our data are stored in a data frame, it's convenient to use the `xtabs()` function.

```
chapekFrequencies <- xtabs( ~ choice + species, data = chapek9)
chapekFrequencies
```

```
##         species
## choice   robot human
##   puppy     13    15
##   flower    30    13
##   data      44    65
```

That's more or less what we're after. So, if we add the row and column totals (which is convenient for the purposes of explaining the statistical tests), we would have a table like this,

	Robot	Human	Total
Puppy	13	15	28
Flower	30	13	43
Data file	44	65	109
Total	87	93	180
which actual	ly would	be a nice	way to report the descriptive statistics for this data set. In any case, it's quite clear that the vast majority of the humans chose the data file, whereas the robots tended to be a lot more even in their preferences. Leaving aside the question of *why* the humans might be more likely to choose the data file for the moment (which does seem quite odd, admittedly), our first order of business is to determine if the discrepancy between human choices and robot choices in the data set is statistically significant.

10.2.1 Constructing our hypothesis test

How do we analyse this data? Specifically, since my *research* hypothesis is that "humans and robots answer the question in different ways", how can I construct a test of the *null* hypothesis that "humans and robots answer the question the same way"? As before, we begin by establishing some notation to describe the data:

	Robot	Human	Total
Puppy	O_{11}	O_{12}	R_1
Flower	O_{21}	O_{22}	R_2
Data file	O_{31}	O_{32}	R_3
Total	C_1	C_2	N

In this notation we say that O_{ij} is a count (observed frequency) of the number of respondents that are of species j (robots or human) who gave answer i (puppy, flower or data) when asked to make a choice. The total number of observations is written N, as usual. Finally, I've used R_i to denote the row totals (e.g., R_1 is the total number of people who chose the flower), and C_j to denote the column totals (e.g., C_1 is the total number of robots).[176]

So now let's think about what the null hypothesis says. If robots and humans are responding in the same way to the question, it means that the probability that "a robot says puppy" is the same as the probability that "a human says puppy", and so on for the other two possibilities. So, if we use P_{ij} to denote "the probability that a member of species j gives response i" then our null hypothesis is that:

H0:	All of the following are true:
	$P_{11}=P_{12}$ (same probability of saying puppy)
	$P_{21}=P_{22}$ (same probability of saying flower) and
	$P_{31}=P_{32}$ (same probability of saying data).

And actually, since the null hypothesis is claiming that the true choice probabilities don't depend on the species of the person making the choice, we can let P_i refer to this probability: e.g., P_1 is the true probability of choosing the puppy.

Next, in much the same way that we did with the goodness of fit test, what we need to do is calculate the expected frequencies. That is, for each of the observed counts O_{ij}, we need to figure out what the null hypothesis would tell us to expect. Let's denote this expected frequency by E_{ij}. This time, it's a little bit trickier. If there are a total of C_j people that belong to species j, and the true probability of anyone (regardless of species) choosing option i is P_i, then the expected frequency is just:

$$E_{ij} = C_j \times P_i$$

Now, this is all very well and good, but we have a problem. Unlike the situation we had with the goodness of fit test, the null hypothesis doesn't actually specify a particular value for P_i. It's something we have to estimate (Chapter 10) from the data! Fortunately, this is pretty easy to do. If 28 out of 180 people selected the flowers, then a natural estimate for the probability of choosing flowers is 28/180, which is approximately .16. If we phrase this in mathematical terms, what we're saying is that our estimate for the probability of choosing option i is just the row total divided by the total sample size:

$$\hat{P}_i = \frac{R_i}{N}$$

Therefore, our expected frequency can be written as the product (i.e. multiplication) of the row total and the column total, divided by the total number of observations:[177]

$$E_{ij} = \frac{R_i \times C_j}{N}$$

Now that we've figured out how to calculate the expected frequencies, it's straightforward to define a test statistic; following the exact same strategy that we used in the goodness of fit test. In fact, it's pretty much the *same* statistic. For a contingency table with

r rows and c columns, the equation that defines our X_2 statistic is

$$X^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(E_{ij} - O_{ij})^2}{E_{ij}}$$

The only difference is that I have to include two summation sign (i.e., $\sum$) to indicate that we're summing over both rows and columns. As before, large values of X^2 indicate that the null hypothesis provides a poor description of the data, whereas small values of X^2 suggest that it does a good job of accounting for the data. Therefore, just like last time, we want to reject the null hypothesis if X^2 is too large.

Not surprisingly, this statistic is X^2 distributed. All we need to do is figure out how many degrees of freedom are involved, which actually isn't too hard. As I mentioned before, you can (usually) think of the degrees of freedom as being equal to the number of data points that you're analysing, minus the number of constraints. A contingency table with r rows and c columns contains a total of r×c observed frequencies, so that's the total number of observations. What about the constraints? Here, it's slightly trickier. The answer is always the same

$$\text{df} = (r-1)(c-1)$$

but the explanation for *why* the degrees of freedom takes this value is different depending on the experimental design. For the sake of argument, let's suppose that we had honestly intended to survey exactly 87 robots and 93 humans (column totals fixed by the experimenter), but left the row totals free to vary (row totals are random variables). Let's think about the constraints that apply here. Well, since we deliberately fixed the column totals by Act of Experimenter, we have c constraints right there. But, there's actually more to it than that. Remember how our null hypothesis had some free parameters (i.e., we had to estimate the P_i values)? Those matter too. I won't explain why in this book, but every free parameter in the null hypothesis is rather like an additional constraint. So, how many of those are there? Well, since these probabilities have to sum to 1, there's only r−1 of these. So our total degrees of freedom is:

$$\begin{aligned}
\text{df} &= (\text{number of observations}) - (\text{number of constraints}) \\
&= (rc) - (c + (r-1)) \\
&= rc - c - r + 1 \\
&= (r-1)(c-1)
\end{aligned}$$

Alternatively, suppose that the only thing that the experimenter fixed was the total sample size N. That is, we quizzed the first 180 people that we saw, and it just turned out that 87 were robots and 93 were humans. This time around our reasoning would be slightly different, but would still lead is to the same answer. Our null hypothesis still has r−1 free parameters corresponding to the choice probabilities, but it now *also* has c−1 free parameters corresponding to the species probabilities, because we'd also have to estimate the probability that a randomly sampled person turns out to be a robot.[178] Finally, since we did actually fix the total number of observations N, that's one more constraint. So now we have, rc observations, and (c−1)+(r−1)+1 constraints. What does that give?

$$\begin{aligned}
\text{df} &= (\text{number of observations}) - (\text{number of constraints}) \\
&= rc - ((c-1) + (r-1) + 1) \\
&= rc - c - r + 1 \\
&= (r-1)(c-1)
\end{aligned}$$

Amazing.

10.2.2 Doing the test in R

Okay, now that we know how the test works, let's have a look at how it's done in R. As tempting as it is to lead you through the tedious calculations so that you're forced to learn it the long way, I figure there's no point. I already showed you how to do it the long way for the goodness of fit test in the last section, and since the test of independence isn't conceptually any different, you won't learn anything new by doing it the long way. So instead, I'll go straight to showing you the easy way. As always, R lets you do it multiple ways. There's the `chisq.test()` function, which I'll talk about in Section @ref(chisq.test, but first I want to

use the `associationTest()` function in the `lsr` package, which I think is easier on beginners. It works in the exact same way as the `xtabs()` function. Recall that, in order to produce the contingency table, we used this command:

```
xtabs( formula = ~choice+species, data = chapek9 )
```

```
##         species
## choice   robot human
##   puppy     13    15
##   flower    30    13
##   data      44    65
```

The `associationTest()` function has exactly the same structure: it needs a `formula` that specifies which variables you're cross-tabulating, and the name of a `data` frame that contains those variables. So the command is just this:

```
associationTest( formula = ~choice+species, data = chapek9 )
```

```
##
##      Chi-square test of categorical association
##
## Variables:   choice, species
##
## Hypotheses:
##    null:        variables are independent of one another
##    alternative: some contingency exists between variables
##
## Observed contingency table:
##         species
## choice   robot human
##   puppy     13    15
##   flower    30    13
##   data      44    65
##
## Expected contingency table under the null hypothesis:
##         species
## choice   robot human
##   puppy   13.5  14.5
##   flower  20.8  22.2
##   data    52.7  56.3
##
## Test results:
##    X-squared statistic:  10.722
##    degrees of freedom:  2
##    p-value:  0.005
##
## Other information:
##    estimated effect size (Cramer's v):  0.244
```

Just like we did with the goodness of fit test, I'll go through it line by line. The first two lines are, once again, just reminding you what kind of test you ran and what variables were used:

```
Chi-square test of categorical association

Variables:   choice, species
```

Next, it tells you what the null and alternative hypotheses are (and again, I want to remind you not to get used to seeing these hypotheses written out so explicitly):

```
Hypotheses:
   null:        variables are independent of one another
   alternative: some contingency exists between variables
```

Next, it shows you the observed contingency table that is being tested:

```
Observed contingency table:
        species
choice   robot human
  puppy     13    15
  flower    30    13
  data      44    65
```

and it also shows you what the expected frequencies would be if the null hypothesis were true:

```
Expected contingency table under the null hypothesis:
        species
choice   robot human
  puppy   13.5  14.5
  flower  20.8  22.2
  data    52.7  56.3
```

The next part describes the results of the hypothesis test itself:

```
Test results:
   X-squared statistic:  10.722
   degrees of freedom:  2
   p-value:  0.005
```

And finally, it reports a measure of effect size:

```
Other information:
   estimated effect size (Cramer's v):  0.244
```

You can ignore this bit for now. I'll talk about it in just a moment.

This output gives us enough information to write up the result:

Pearson's $\chi 2$ revealed a significant association between species and choice ($\chi^2(2)=10.7, p<.01$): robots appeared to be more likely to say that they prefer flowers, but the humans were more likely to say they prefer data.

Notice that, once again, I provided a little bit of interpretation to help the human reader understand what's going on with the data. Later on in my discussion section, I'd provide a bit more context. To illustrate the difference, here's what I'd probably say later on:

The fact that humans appeared to have a stronger preference for raw data files than robots is somewhat counterintuitive. However, in context it makes some sense: the civil authority on Chapek 9 has an unfortunate tendency to kill and dissect humans when they are identified. As such it seems most likely that the human participants did not respond honestly to the question, so as to avoid potentially undesirable consequences. This should be considered to be a substantial methodological weakness.

This could be classified as a rather extreme example of a reactivity effect, I suppose. Obviously, in this case the problem is severe enough that the study is more or less worthless as a tool for understanding the difference preferences among humans and robots. However, I hope this illustrates the difference between getting a statistically significant result (our null hypothesis is rejected in favour of the alternative), and finding something of scientific value (the data tell us nothing of interest about our research hypothesis due to a big methodological flaw).

10.2.3 Postscript

I later found out the data were made up, and I'd been watching cartoons instead of doing work.

This page titled 10.2: The χ2 test of independence (or association) is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **12.2: The χ2 test of independence (or association)** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

10.3: The Continuity Correction

Okay, time for a little bit of a digression. I've been lying to you a little bit so far. There's a tiny change that you need to make to your calculations whenever you only have 1 degree of freedom. It's called the "continuity correction", or sometimes the ***Yates correction***. Remember what I pointed out earlier: the $\chi 2$ test is based on an approximation, specifically on the assumption that binomial distribution starts to look like a normal distribution for large N. One problem with this is that it often doesn't quite work, especially when you've only got 1 degree of freedom (e.g., when you're doing a test of independence on a 2×2 contingency table). The main reason for this is that the true sampling distribution for the X^2 statistic is actually discrete (because you're dealing with categorical data!) but the $\chi 2$ distribution is continuous. This can introduce systematic problems. Specifically, when N is small and when df=1, the goodness of fit statistic tends to be "too big", meaning that you actually have a bigger α value than you think (or, equivalently, the p values are a bit too small). Yates (1934) suggested a simple fix, in which you redefine the goodness of fit statistic as:

$$X^2 = \sum_i \frac{(|E_i - O_i| - 0.5)^2}{E_i}$$

Basically, he just subtracts off 0.5 everywhere. As far as I can tell from reading Yates' paper, the correction is basically a hack. It's not derived from any principled theory: rather, it's based on an examination of the behaviour of the test, and observing that the corrected version seems to work better. I feel obliged to explain this because you will sometimes see R (or any other software for that matter) introduce this correction, so it's kind of useful to know what they're about. You'll know when it happens, because the R output will explicitly say that it has used a "continuity correction" or "Yates' correction".

This page titled 10.3: The Continuity Correction is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **12.3: The Continuity Correction** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

10.4: Effect Size

As we discussed earlier (Section 11.8), it's becoming commonplace to ask researchers to report some measure of effect size. So, let's suppose that you've run your chi-square test, which turns out to be significant. So you now know that there is some association between your variables (independence test) or some deviation from the specified probabilities (goodness of fit test). Now you want to report a measure of effect size. That is, given that there is an association/deviation, how strong is it?

There are several different measures that you can choose to report, and several different tools that you can use to calculate them. I won't discuss all of them,[179] but will instead focus on the most commonly reported measures of effect size.

By default, the two measures that people tend to report most frequently are the ϕ statistic and the somewhat superior version, known as Cram'er's V. Mathematically, they're very simple. To calculate the ϕ statistic, you just divide your X^2 value by the sample size, and take the square root:

$$\phi = \sqrt{\frac{X^2}{N}}$$

The idea here is that the ϕ statistic is supposed to range between 0 (no at all association) and 1 (perfect association), but it doesn't always do this when your contingency table is bigger than 2×2, which is a total pain. For bigger tables it's actually possible to obtain ϕ>1, which is pretty unsatisfactory. So, to correct for this, people usually prefer to report the V statistic proposed by Cramér (1946). It's a pretty simple adjustment to ϕ. If you've got a contingency table with r rows and c columns, then define k=min(r,c) to be the smaller of the two values. If so, then ***Cram'er's V*** statistic is

$$V = \sqrt{\frac{X^2}{N(k-1)}}$$

And you're done. This seems to be a fairly popular measure, presumably because it's easy to calculate, and it gives answers that aren't completely silly: you know that V really does range from 0 (no at all association) to 1 (perfect association).

Calculating V or ϕ is obviously pretty straightforward. So much so that the core packages in R don't seem to have functions to do it, though other packages do. To save you the time and effort of finding one, I've included one in the `lsr` package, called `cramersV()`. It takes a contingency table as input, and prints out the measure of effect size:

```
cramersV( chapekFrequencies )
```

```
## [1] 0.244058
```

However, if you're using the `associationTest()` function to do your analysis, then you won't actually need to use this at all, because it reports the Cram'er's V statistic as part of the output.

This page titled 10.4: Effect Size is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **12.4: Effect Size** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

10.5: Assumptions of the Test(s)

All statistical tests make assumptions, and it's usually a good idea to check that those assumptions are met. For the chi-square tests discussed so far in this chapter, the assumptions are:

- *Expected frequencies are sufficiently large*. Remember how in the previous section we saw that the $\chi 2$ sampling distribution emerges because the binomial distribution is pretty similar to a normal distribution? Well, like we discussed in Chapter 9 this is only true when the number of observations is sufficiently large. What that means in practice is that all of the expected frequencies need to be reasonably big. How big is reasonably big? Opinions differ, but the default assumption seems to be that you generally would like to see all your expected frequencies larger than about 5, though for larger tables you would probably be okay if at least 80% of the the expected frequencies are above 5 and none of them are below 1. However, from what I've been able to discover , these seem to have been proposed as rough guidelines, not hard and fast rules; and they seem to be somewhat conservative [Larntz1978].
- *Data are independent of one another*. One somewhat hidden assumption of the chi-square test is that you have to genuinely believe that the observations are independent. Here's what I mean. Suppose I'm interested in proportion of babies born at a particular hospital that are boys. I walk around the maternity wards, and observe 20 girls and only 10 boys. Seems like a pretty convincing difference, right? But later on, it turns out that I'd actually walked into the same ward 10 times, and in fact I'd only seen 2 girls and 1 boy. Not as convincing, is it? My original 30 *observations* were massively non-independent… and were only in fact equivalent to 3 independent observations. Obviously this is an extreme (and extremely silly) example, but it illustrates the basic issue. Non-independence "stuffs things up". Sometimes it causes you to falsely reject the null, as the silly hospital example illustrats, but it can go the other way too. To give a slightly less stupid example, let's consider what would happen if I'd done the cards experiment slightly differently: instead of asking 200 people to try to imagine sampling one card at random, suppose I asked 50 people to select 4 cards. One possibility would be that *everyone* selects one heart, one club, one diamond and one spade (in keeping with the "representativeness heuristic"; Tversky & Kahneman 1974). This is highly non-random behaviour from people, but in this case, I would get an observed frequency of 50 four all four suits. For this example, the fact that the observations are non-independent (because the four cards that you pick will be related to each other) actually leads to the opposite effect… falsely retaining the null.

If you happen to find yourself in a situation where independence is violated, it may be possible to use the McNemar test (which we'll discuss) or the Cochran test (which we won't). Similarly, if your expected cell counts are too small, check out the Fisher exact test. It is to these topics that we now turn.

This page titled 10.5: Assumptions of the Test(s) is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **12.5: Assumptions of the Test(s)** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

10.6: The Most Typical Way to Do Chi-square Tests in R

When discussing how to do a chi-square goodness of fit test (Section 12.1.7) and the chi-square test of independence (Section 12.2.2), I introduced you to two separate functions in the `lsr` package. We ran our goodness of fit tests using the `goodnessOfFitTest()` function, and our tests of independence (or association) using the `associationTest()` function. And both of those functions produced quite detailed output, showing you the relevant descriptive statistics, printing out explicit reminders of what the hypotheses are, and so on. When you're first starting out, it can be very handy to be given this sort of guidance. However, once you start becoming a bit more proficient in statistics and in R it can start to get very tiresome. A real statistician hardly needs to be told what the null and alternative hypotheses for a chi-square test are, and if an advanced R user wants the descriptive statistics to be printed out, they know how to produce them!

For this reason, the basic `chisq.test()` function in R is a lot more terse in its output, and because the mathematics that underpin the goodness of fit test and the test of independence is basically the same in each case, it can run either test depending on what kind of input it is given. First, here's the goodness of fit test. Suppose you have the frequency table `observed` that we used earlier,

```
observed
```

```
##
##    clubs diamonds   hearts   spades
##       35       51       64       50
```

If you want to run the goodness of fit test against the hypothesis that all four suits are equally likely to appear, then all you need to do is input this frequenct table to the `chisq.test()` function:

```
chisq.test( x = observed )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  observed
## X-squared = 8.44, df = 3, p-value = 0.03774
```

Notice that the output is very compressed in comparison to the `goodnessOfFitTest()` function. It doesn't bother to give you any descriptive statistics, it doesn't tell you what null hypothesis is being tested, and so on. And as long as you already understand the test, that's not a problem. Once you start getting familiar with R and with statistics, you'll probably find that you prefer this simple output rather than the rather lengthy output that `goodnessOfFitTest()` produces. Anyway, if you want to change the null hypothesis, it's exactly the same as before, just specify the probabilities using the `p` argument. For instance:

```
chisq.test( x = observed, p = c(.2, .3, .3, .2) )
```

```
##
##  Chi-squared test for given probabilities
##
## data:  observed
## X-squared = 4.7417, df = 3, p-value = 0.1917
```

Again, these are the same numbers that the `goodnessOfFitTest()` function reports at the end of the output. It just hasn't included any of the other details.

What about a test of independence? As it turns out, the `chisq.test()` function is pretty clever.[180] If you input a *cross-tabulation* rather than a simple frequency table, it realises that you're asking for a test of independence and not a goodness of fit test. Recall that we already have this cross-tabulation stored as the `chapekFrequencies` variable:

```
chapekFrequencies
```

```
##         species
## choice   robot human
##   puppy     13    15
##   flower    30    13
##   data      44    65
```

To get the test of independence, all we have to do is feed this frequency table into the `chisq.test()` function like so:

```
chisq.test( chapekFrequencies )
```

```
##
##  Pearson's Chi-squared test
##
## data:  chapekFrequencies
## X-squared = 10.722, df = 2, p-value = 0.004697
```

Again, the numbers are the same as last time, it's just that the output is very terse and doesn't really explain what's going on in the rather tedious way that `associationTest()` does. As before, my intuition is that when you're just getting started it's easier to use something like `associationTest()` because it shows you more detail about what's going on, but later on you'll probably find that `chisq.test()` is more convenient.

This page titled 10.6: The Most Typical Way to Do Chi-square Tests in R is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **12.6: The Most Typical Way to Do Chi-square Tests in R** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

10.7: The Fisher Exact Test

What should you do if your cell counts are too small, but you'd still like to test the null hypothesis that the two variables are independent? One answer would be "collect more data", but that's far too glib: there are a lot of situations in which it would be either infeasible or unethical do that. If so, statisticians have a kind of moral obligation to provide scientists with better tests. In this instance, Fisher (1922) kindly provided the right answer to the question. To illustrate the basic idea, let's suppose that we're analysing data from a field experiment, looking at the emotional status of people who have been accused of witchcraft; some of whom are currently being burned at the stake.[181] Unfortunately for the scientist (but rather fortunately for the general populace), it's actually quite hard to find people in the process of being set on fire, so the cell counts are awfully small in some cases. The `salem.Rdata` file illustrates the point:

```
load("./rbook-master/data/salem.Rdata")

salem.tabs <- table( trial )
print( salem.tabs )
```

```
##        on.fire
## happy   FALSE TRUE
##   FALSE     3    3
##   TRUE     10    0
```

Looking at this data, you'd be hard pressed not to suspect that people not on fire are more likely to be happy than people on fire. However, the chi-square test makes this very hard to test because of the small sample size. If I try to do so, R gives me a warning message:

```
chisq.test( salem.tabs )
```

```
## Warning in chisq.test(salem.tabs): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  salem.tabs
## X-squared = 3.3094, df = 1, p-value = 0.06888
```

Speaking as someone who doesn't want to be set on fire, I'd *really* like to be able to get a better answer than this. This is where ***Fisher's exact test*** comes in very handy.

The Fisher exact test works somewhat differently to the chi-square test (or in fact any of the other hypothesis tests that I talk about in this book) insofar as it doesn't have a test statistic; it calculates the p-value "directly". I'll explain the basics of how the test works for a 2×2 contingency table, though the test works fine for larger tables. As before, let's have some notation:

	Happy	Sad	Total
Set on fire	O_{11}	O_{12}	R_1
Not set on fire	O_{21}	O_{22}	R_2
Total	C_1	C_2	N

In order to construct the test Fisher treats both the row and column totals (R_1, R_2, C_1 and C_2) are known, fixed quantities; and then calculates the probability that we would have obtained the observed frequencies that we did (O_{11}, O_{12}, O_{21} and O_{22}) given those totals. In the notation that we developed in Chapter 9 this is written:

$$P(O_{11},O_{12},O_{21},O_{22} \mid R_1,R_2,C_1,C_2)$$

and as you might imagine, it's a slightly tricky exercise to figure out what this probability is, but it turns out that this probability is described by a distribution known as the *hypergeometric distribution*.[182] Now that we know this, what we have to do to calculate our p-value is calculate the probability of observing this particular table *or a table that is "more extreme"*.[183] Back in the 1920s, computing this sum was daunting even in the simplest of situations, but these days it's pretty easy as long as the tables aren't too big and the sample size isn't too large. The conceptually tricky issue is to figure out what it means to say that one contingency table is more "extreme" than another. The easiest solution is to say that the table with the lowest probability is the most extreme. This then gives us the p-value.

The implementation of the test in R is via the `fisher.test()` function. Here's how it is used:

```
fisher.test( salem.tabs )
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  salem.tabs
## p-value = 0.03571
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.000000 1.202913
## sample estimates:
## odds ratio
##          0
```

This is a bit more output than we got from some of our earlier tests. The main thing we're interested in here is the p-value, which in this case is small enough (p=.036) to justify rejecting the null hypothesis that people on fire are just as happy as people not on fire.

This page titled 10.7: The Fisher Exact Test is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **12.7: The Fisher Exact Test** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

10.8: The McNemar Test

Suppose you've been hired to work for the *Australian Generic Political Party* (AGPP), and part of your job is to find out how effective the AGPP political advertisements are. So, what you do, is you put together a sample of N=100 people, and ask them to watch the AGPP ads. Before they see anything, you ask them if they intend to vote for the AGPP; and then after showing the ads, you ask them again, to see if anyone has changed their minds. Obviously, if you're any good at your job, you'd also do a whole lot of other things too, but let's consider just this one simple experiment. One way to describe your data is via the following contingency table:

	Before	After	Total
Yes	30	10	40
No	70	90	160
Total	100	100	200

At first pass, you might think that this situation lends itself to the Pearson $\chi 2$ test of independence (as per Section 12.2). However, a little bit of thought reveals that we've got a problem: we have 100 participants, but 200 observations. This is because each person has provided us with an answer in *both* the before column and the after column. What this means is that the 200 observations aren't independent of each other: if voter A says "yes" the first time and voter B says "no", then you'd expect that voter A is more likely to say "yes" the second time than voter B! The consequence of this is that the usual $\chi 2$ test won't give trustworthy answers due to the violation of the independence assumption. Now, if this were a really uncommon situation, I wouldn't be bothering to waste your time talking about it. But it's not uncommon at all: this is a *standard* repeated measures design, and none of the tests we've considered so far can handle it. Eek.

The solution to the problem was published by McNemar (1947). The trick is to start by tabulating your data in a slightly different way:

	Before: Yes	Before: No	Total
After: Yes	5	5	10
After: No	25	65	90
Total	30	70	100

This is exactly the same data, but it's been rewritten so that each of our 100 participants appears in only one cell. Because we've written our data this way, the independence assumption is now satisfied, and this is a contingency table that we *can* use to construct an X^2 goodness of fit statistic. However, as we'll see, we need to do it in a slightly nonstandard way. To see what's going on, it helps to label the entries in our table a little differently:

	Before: Yes	Before: No	Total
After: Yes	a	b	a+b
After: No	c	d	c+d
Total	a+c	b+d	n

Next, let's think about what our null hypothesis is: it's that the "before" test and the "after" test have the same proportion of people saying "Yes, I will vote for AGPP". Because of the way that we have rewritten the data, it means that we're now testing the hypothesis that the *row totals* and *column totals* come from the same distribution. Thus, the null hypothesis in McNemar's test is that we have "marginal homogeneity". That is, the row totals and column totals have the same distribution: $P_a+P_b=P_a+P_c$, and similarly that $P_c+P_d=P_b+P_d$. Notice that this means that the null hypothesis actually simplifies to $P_b=P_c$. In other words, as far as the McNemar test is concerned, it's only the off-diagonal entries in this table (i.e., b and c) that matter! After noticing this, the

McNemar test of marginal homogeneity is no different to a usual χ^2 test. After applying the Yates correction, our test statistic becomes:

$$X^2 = \frac{(|b-c| - 0.5)^2}{b+c}$$

or, to revert to the notation that we used earlier in this chapter:

$$X^2 = \frac{(|O_{12} - O_{21}| - 0.5)^2}{O_{12} + O_{21}}$$

and this statistic has an (approximately) χ^2 distribution with df=1. However, remember that – just like the other χ^2 tests – it's only an approximation, so you need to have reasonably large expected cell counts for it to work.

10.8.1 Doing the McNemar test in R

Now that you know what the McNemar test is all about, lets actually run one. The `agpp.Rdata` file contains the raw data that I discussed previously, so let's have a look at it:

```
load("./rbook-master/data/agpp.Rdata")
str(agpp)
```

```
## 'data.frame':    100 obs. of  3 variables:
##  $ id             : Factor w/ 100 levels "subj.1","subj.10",..: 1 13 24 35 46 57 6
##  $ response_before: Factor w/ 2 levels "no","yes": 1 2 2 2 1 1 1 1 1 1 ...
##  $ response_after : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 2 1 1 ...
```

The `agpp` data frame contains three variables, an `id` variable that labels each participant in the data set (we'll see why that's useful in a moment), a `response_before` variable that records the person's answer when they were asked the question the first time, and a `response_after` variable that shows the answer that they gave when asked the same question a second time. As usual, here's the first 6 entries:

```
head(agpp)
```

```
##       id response_before response_after
## 1 subj.1              no            yes
## 2 subj.2             yes             no
## 3 subj.3             yes             no
## 4 subj.4             yes             no
## 5 subj.5              no             no
## 6 subj.6              no             no
```

and here's a summary:

```
summary(agpp)
```

```
##         id     response_before response_after
##  subj.1  : 1   no :70          no :90
##  subj.10 : 1   yes:30          yes:10
##  subj.100: 1
##  subj.11 : 1
##  subj.12 : 1
##  subj.13 : 1
##  (Other) :94
```

Notice that each participant appears only once in this data frame. When we tabulate this data frame using `xtabs()`, we get the appropriate table:

```
right.table <- xtabs( ~ response_before + response_after, data = agpp)
print( right.table )
```

```
##                response_after
## response_before no yes
##             no  65   5
##             yes 25   5
```

and from there, we can run the McNemar test by using the `mcnemar.test()` function:

```
mcnemar.test( right.table )
```

```
##
##  McNemar's Chi-squared test with continuity correction
##
## data:  right.table
## McNemar's chi-squared = 12.033, df = 1, p-value = 0.0005226
```

And we're done. We've just run a McNemar's test to determine if people were just as likely to vote AGPP after the ads as they were before hand. The test was significant ($\chi 2(1)=12.04, p<.001$), suggesting that they were not. And in fact, it looks like the ads had a negative effect: people were less likely to vote AGPP after seeing the ads. Which makes a lot of sense when you consider the quality of a typical political advertisement.

This page titled 10.8: The McNemar Test is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **12.8: The McNemar Test** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

10.9: What's the Difference Between McNemar and Independence?

Let's go all the way back to the beginning of the chapter, and look at the `cards` data set again. If you recall, the actual experimental design that I described involved people making *two* choices. Because we have information about the first choice and the second choice that everyone made, we can construct the following contingency table that cross-tabulates the first choice against the second choice.

```
cardChoices <- xtabs( ~ choice_1 + choice_2, data = cards )
cardChoices
```

```
##           choice_2
## choice_1   clubs diamonds hearts spades
##   clubs       10        9     10      6
##   diamonds    20        4     13     14
##   hearts      20       18      3     23
##   spades      18       13     15      4
```

Suppose I wanted to know whether the choice you make the second time is dependent on the choice you made the first time. This is where a test of independence is useful, and what we're trying to do is see if there's some relationship between the rows and columns of this table. Here's the result:

```
chisq.test( cardChoices )
```

Alternatively, suppose I wanted to know if *on average*, the frequencies of suit choices were different the second time than the first time. In that situation, what I'm really trying to see if the row totals in `cardChoices` (i.e., the frequencies for `choice_1`) are different from the column totals (i.e., the frequencies for `choice_2`). That's when you use the McNemar test:

```
mcnemar.test( cardChoices )
```

```
##
##  McNemar's Chi-squared test
##
## data:  cardChoices
## McNemar's chi-squared = 16.033, df = 6, p-value = 0.01358
```

Notice that the results are different! These aren't the same test.

This page titled 10.9: What's the Difference Between McNemar and Independence? is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **12.9: What's the Difference Between McNemar and Independence?** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

10.10: Summary

The key ideas discussed in this chapter are:

- The chi-square goodness of fit test (Section 12.1) is used when you have a table of observed frequencies of different categories; and the null hypothesis gives you a set of "known" probabilities to compare them to. You can either use the `goodnessOfFitTest()` function in the `lsr` package to run this test, or the `chisq.test()` function.
- The chi-square test of independence (Section 12.2) is used when you have a contingency table (cross-tabulation) of two categorical variables. The null hypothesis is that there is no relationship/association between the variables. You can either use the `associationTest()` function in the `lsr` package, or you can use `chisq.test()`.
- Effect size for a contingency table can be measured in several ways (Section 12.4). In particular we noted the Cramer's V statistic, which can be calculated using `cramersV()`. This is also part of the output produced by `associationTest()`.
- Both versions of the Pearson test rely on two assumptions: that the expected frequencies are sufficiently large, and that the observations are independent (Section 12.5). The Fisher exact test (Section 12.7) can be used when the expected frequencies are small, `fisher.test(x = contingency.table)`. The McNemar test (Section 12.8) can be used for some kinds of violations of independence, `mcnemar.test(x = contingency.table)`.

If you're interested in learning more about categorical data analysis, a good first choice would be Agresti (1996) which, as the title suggests, provides an *Introduction to Categorical Data Analysis*. If the introductory book isn't enough for you (or can't solve the problem you're working on) you could consider Agresti (2002), *Categorical Data Analysis*. The latter is a more advanced text, so it's probably not wise to jump straight from this book to that one.

References

Pearson, K. 1900. "On the Criterion That a Given System of Deviations from the Probable in the Case of a Correlated System of Variables Is Such That It Can Be Reasonably Supposed to Have Arisen from Random Sampling." *Philosophical Magazine* 50: 157–75.

Fisher, R. A. 1922a. "On the Interpretation of $\chi 2$ from Contingency Tables, and the Calculation of p." *Journal of the Royal Statistical Society* 84: 87–94.

Yates, F. 1934. "Contingency Tables Involving Small Numbers and the $\chi 2$ Test." *Supplement to the Journal of the Royal Statistical Society* 1: 217–35.

Cramér, H. 1946. *Mathematical Methods of Statistics*. Princeton: Princeton University Press.

McNemar, Q. 1947. "Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages." *Psychometrika* 12: 153–57.

Agresti, A. 1996. *An Introduction to Categorical Data Analysis*. Hoboken, NJ: Wiley.

Agresti, A. 2002. *Categorical Data Analysis*. 2nd ed. Hoboken, NJ: Wiley.

---

170. I should point out that this issue does complicate the story somewhat: I'm not going to cover it in this book, but there's a sneaky trick that you can do to rewrite the equation for the goodness of fit statistic as a sum over k−1 independent things. When we do so we get the "proper" sampling distribution, which is chi-square with k−1 degrees of freedom. In fact, in order to get the maths to work out properly, you actually have to rewrite things that way. But it's beyond the scope of an introductory book to show the maths in that much detail: all I wanted to do is give you a sense of why the goodness of fit statistic is associated with the chi-squared distribution.
171. I feel obliged to point out that this is an over-simplification. It works nicely for quite a few situations; but every now and then we'll come across degrees of freedom values that aren't whole numbers. Don't let this worry you too much – when you come across this, just remind yourself that "degrees of freedom" is actually a bit of a messy concept, and that the nice simple story that I'm telling you here isn't the whole story. For an introductory class, it's usually best to stick to the simple story: but I figure it's best to warn you to expect this simple story to fall apart. If I didn't give you this warning, you might start getting confused

when you see df=3.4 or something; and (incorrectly) thinking that you had misunderstood something that I've taught you, rather than (correctly) realising that there's something that I haven't told you.

172. In practice, the sample size isn't always fixed… e.g., we might run the experiment over a fixed period of time, and the number of people participating depends on how many people show up. That doesn't matter for the current purposes.
173. Well, sort of. The conventions for how statistics should be reported tend to differ somewhat from discipline to discipline; I've tended to stick with how things are done in psychology, since that's what I do. But the general principle of providing enough information to the reader to allow them to check your results is pretty universal, I think.
174. To some people, this advice might sound odd, or at least in conflict with the "usual" advice on how to write a technical report. Very typically, students are told that the "results" section of a report is for describing the data and reporting statistical analysis; and the "discussion" section is for providing interpretation. That's true as far as it goes, but I think people often interpret it way too literally. The way I usually approach it is to provide a quick and simple interpretation of the data in the results section, so that my reader understands what the data are telling us. Then, in the discussion, I try to tell a bigger story; about how my results fit with the rest of the scientific literature. In short; don't let the "interpretation goes in the discussion" advice turn your results section into incomprehensible garbage. Being understood by your reader is *much* more important.
175. Complicating matters, the G-test is a special case of a whole class of tests that are known as *likelihood ratio tests*. I don't cover LRTs in this book, but they are quite handy things to know about.
176. A technical note. The way I've described the test pretends that the column totals are fixed (i.e., the researcher intended to survey 87 robots and 93 humans) and the row totals are random (i.e., it just turned out that 28 people chose the puppy). To use the terminology from my mathematical statistics textbook (Hogg, McKean, and Craig 2005) I should technically refer to this situation as a chi-square test of homogeneity; and reserve the term chi-square test of independence for the situation where both the row and column totals are random outcomes of the experiment. In the initial drafts of this book that's exactly what I did. However, it turns out that these two tests are identical; and so I've collapsed them together.
177. Technically, E_{ij} here is an estimate, so I should probably write it $\hat{E}_{ij}$. But since no-one else does, I won't either.
178. A problem many of us worry about in real life.
179. Though I do feel that it's worth mentioning the `assocstats()` function in the `vcd` package. If you install and load the `vcd` package, then a command like `assocstats( chapekFrequencies )` will run the $\chi 2$ test as well as the likelihood ratio test (not discussed here); and then report three different measures of effect size: ϕ^2, Cram'er's V, and the contingency coefficient (not discussed here)
180. Not really.
181. This example is based on a joke article published in the *Journal of Irreproducible Results*.
182. The R functions for this distribution are `dhyper()`, `phyper()`, `qhyper()` and `rhyper()`, though you don't need them for this book, and I haven't given you enough information to use these to perform the Fisher exact test the long way.
183. Not surprisingly, the Fisher exact test is motivated by Fisher's interpretation of a p-value, not Neyman's!

This page titled 10.10: Summary is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **12.10: Summary** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

10.11: Statistical Literacy

Learning Objectives

- A Spice Inhibits Liver Cancer

An experiment was conducted to test whether the spice saffron can inhibit liver cancer. Two groups of rats were tested. Both groups were injected with chemicals known to increase the chance of liver cancer. The experimental group was fed saffron ($n = 24$) whereas the control group was not ($n = 8$). The experiment is described here.

Only 4 of the 24 subjects in the saffron group developed cancer as compared to 6 of the 8 subjects in the control group.

Example 10.11.1: what do you think?

What method could be used to test whether this difference between the experimental and control groups is statistically significant? Use Analysis Lab to do the test.

Solution

The Chi Square test of contingency tables could be used. It yields a χ^2 ($df = 1$) of 9.50 which has an associated p of 0.002.

This page titled 10.11: Statistical Literacy is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

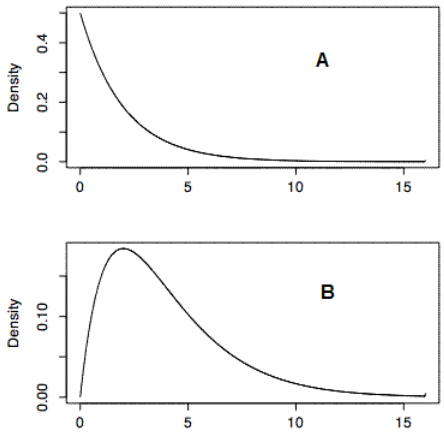
- **17.6: Statistical Literacy** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

10.12: Chi Square (Exercises)

General Questions

Q1

Which of the two Chi Square distributions shown below (*A* or *B*) has the larger degrees of freedom? How do you know? (relevant section)



Q2

Twelve subjects were each given two flavors of ice cream to taste and then were asked whether they liked them. Two of the subjects liked the first flavor and nine of them liked the second flavor. Is it valid to use the Chi Square test to determine whether this difference in proportions is significant? Why or why not? (relevant section)

Q3

A die is suspected of being biased. It is rolled 25 times with the following result:

Outcome	Frequency
1	9
2	4
3	1
4	8
5	3
6	0

Conduct a significance test to see if the die is biased.

a. What Chi Square value do you get and how many degrees of freedom does it have?
b. What is the p value? (relevant section)

Q4

A recent experiment investigated the relationship between smoking and urinary incontinence. Of the 322 subjects in the study who were incontinent, 113 were smokers, 51 were former smokers, and 158 had never smoked. Of the 284 control subjects who were not

incontinent, 68 were smokers, 23 were former smokers, and 193 had never smoked.

a. Create a table displaying this data.
b. What is the expected frequency in each cell?
c. Conduct a significance test to see if there is a relationship between smoking and incontinence. What Chi Square value do you get? What p value do you get?
d. What do you conclude? (relevant section)

Q5

At a school pep rally, a group of sophomore students organized a free raffle for prizes. They claim that they put the names of all of the students in the school in the basket and that they randomly drew 36 names out of this basket. Of the prize winners, 6 were freshmen, 14 were sophomores, 9 were juniors, and 7 were seniors. The results do not seem that random to you. You think it is a little fishy that sophomores organized the raffle and also won the most prizes. Your school is composed of 30% freshmen, 25% sophomores, 25% juniors, and 20% seniors.

a. What are the expected frequencies of winners from each class?
b. Conduct a significance test to determine whether the winners of the prizes were distributed throughout the classes as would be expected based on the percentage of students in each group. Report your Chi Square and p values.
c. What do you conclude? (relevant section)

Q6

Some parents of the West Bay little leaguers think that they are noticing a pattern. There seems to be a relationship between the number on the kids' jerseys and their position. These parents decide to record what they see. The hypothetical data appear below. Conduct a Chi Square test to determine if the parents' suspicion that there is a relationship between jersey number and position is right. Report your Chi Square and p values. (relevant section)

	Infield	Outfield	Pitcher	Total
0-9	12	5	5	22
10-19	5	10	2	17
20+	4	4	7	15
Total	21	19	14	54

Q7

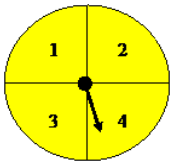
True/false: A Chi Square distribution with 2 df has a larger mean than a Chi Square distribution with 12 df. (relevant section)

Q8

True/false: A Chi Square test is often used to determine if there is a significant relationship between two continuous variables. (relevant section)

Q9

True/false: Imagine that you want to determine if the spinner shown below is biased. You spin it 50 times and write down how many times the arrow lands in each section. You will reject the null hypothesis at the 0.05 level and determine that this spinner is biased if you calculate a Chi Square value of 7.82 or higher. (relevant section)



Questions from Case Studies

The following question uses data from the SAT and GPA (SG) case study.

Q10

Answer these items to determine if the math SAT scores are normally distributed. You may want to first standardize the scores. (relevant section)

a. If these data were normally distributed, how many scores would you expect there to be in each of these brackets:
   i. smaller than 1 SD below the mean
   ii. in between the mean and 1 SD below the mean
   iii. in between the mean and 1 SD above the mean
   iv. greater than 1 SD above the mean?

b. How many scores are actually in each of these brackets?

c. Conduct a Chi Square test to determine if the math SAT scores are normally distributed based on these expected and observed frequencies. (relevant section)

The following questions are from the Diet and Health (DH) case study.

Q11

(DH#3) Conduct a Pearson Chi Square test to determine if there is any relationship between diet and outcome. Report the Chi Square and p values and state your conclusions. (relevant section)

The following questions are from ARTIST (reproduced with permission).



Q12

A study compared members of a medical clinic who filed complaints with a random sample of members who did not complain. The study divided the complainers into two subgroups: those who filed complaints about medical treatment and those who filed nonmedical complaints. Here are the data on the total number in each group and the number who voluntarily left the medical clinic. Set up a two-way table. Analyze these data to see if there is a relationship between complaint (no, yes - medical, yes - nonmedical) and leaving the clinic (yes or no).

	No Complaint	Medical Complaint	Non Medical Complaint
Total	743	199	440
Left	22	26	28

Q13

Imagine that you believe there is a relationship between a person's eye color and where he or she prefers to sit in a large lecture hall. You decide to collect data from a random sample of individuals and conduct a chi-square test of independence. What would your two-way table look like? Use the information to construct such a table, and be sure to label the different levels of each category.

Q14

A geologist collects hand-specimen sized pieces of limestone from a particular area. A qualitative assessment of both texture and color is made with the following results. Is there evidence of association between color and texture for these limestones? Explain

your answer.

	Color		
Texture	Light	Medium	Dark
Fine	4	20	8
Medium	5	23	12
Coarse	21	23	4

Q15

Suppose that college students are asked to identify their preferences in political affiliation (Democrat, Republican, or Independent) and in ice cream (chocolate, vanilla, or strawberry). Suppose that their responses are represented in the following two-way table (with some of the totals left for you to calculate).

	Chocolate	Vanilla	Strawberry	Total
Democrat	26	43	13	82
Republican	45	12	8	65
Independent	9	13	4	
Total		68	25	173

a. What proportion of the respondents prefer chocolate ice cream?
b. What proportion of the respondents are Independents?
c. What proportion of Independents prefer chocolate ice cream?
d. What proportion of those who prefer chocolate ice cream are Independents?
e. Analyze the data to determine if there is a relationship between political party preference and ice cream preference.

Q16

NCAA collected data on graduation rates of athletes in Division I in the mid $1980s$. Among $2,332$ men, $1,343$ had not graduated from college, and among 959 women, 441 had not graduated.

a. Set up a two-way table to examine the relationship between gender and graduation.
b. Identify a test procedure that would be appropriate for analyzing the relationship between gender and graduation. Carry out the procedure and state your conclusion.

Select Answers

S3

a. Chi Square $= 16.0$, $df = 5$

S4

b. Incontinent/Smoker cell: 96.2

S5

b. $p = 0.18$

S6

Chi Square $= 10.2$

S10

b. i. Scores smaller than 1 SD below the mean: 24

S11

Chi Square = 16.6

This page titled 10.12: Chi Square (Exercises) is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **17.7: Chi Square (Exercises)** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

CHAPTER OVERVIEW

11: Comparing Two Means

In the previous chapter we covered the situation when your outcome variable is nominal scale and your predictor variable[184] is also nominal scale. Lots of real world situations have that character, and so you'll find that chi-square tests in particular are quite widely used. However, you're much more likely to find yourself in a situation where your outcome variable is interval scale or higher, and what you're interested in is whether the average value of the outcome variable is higher in one group or another. For instance, a psychologist might want to know if anxiety levels are higher among parents than non-parents, or if working memory capacity is reduced by listening to music (relative to not listening to music). In a medical context, we might want to know if a new drug increases or decreases blood pressure. An agricultural scientist might want to know whether adding phosphorus to Australian native plants will kill them.[185] In all these situations, our outcome variable is a fairly continuous, interval or ratio scale variable; and our predictor is a binary "grouping" variable. In other words, we want to compare the means of the two groups.

The standard answer to the problem of comparing means is to use a t-test, of which there are several varieties depending on exactly what question you want to solve. As a consequence, the majority of this chapter focuses on different types of t-test: one sample t-tests are discussed in Section 13.2, independent samples t-tests are discussed in Sections 13.3 and 13.4, and paired samples t-tests are discussed in Section 13.5. After that, we'll talk a bit about Cohen's d, which is the standard measure of effect size for a t-test (Section 13.8). The later sections of the chapter focus on the assumptions of the t-tests, and possible remedies if they are violated. However, before discussing any of these useful things, we'll start with a discussion of the z-test.

11.1: The one-sample z-test
11.2: The One-sample t-test
11.3: The Independent Samples t-test (Student Test)
11.4: The Independent Samples t-test (Welch Test)
11.5: The Paired-samples t-test
11.6: One Sided Tests
11.7: Using the t.test() Function
11.8: Effect Size
11.9: Checking the Normality of a Sample
11.10: Testing Non-normal Data with Wilcoxon Tests
11.11: Summary
11.12: Statistical Literacy
11.E: Tests of Means (Exercises)

This page titled 11: Comparing Two Means is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

11.1: The one-sample z-test

In this section I'll describe one of the most useless tests in all of statistics: the ***z-test***. Seriously – this test is almost never used in real life. Its only real purpose is that, when teaching statistics, it's a very convenient stepping stone along the way towards the t-test, which is probably the most (over)used tool in all statistics.

11.1.1 inference problem that the test addresses

To introduce the idea behind the z-test, let's use a simple example. A friend of mine, Dr Zeppo, grades his introductory statistics class on a curve. Let's suppose that the average grade in his class is 67.5, and the standard deviation is 9.5. Of his many hundreds of students, it turns out that 20 of them also take psychology classes. Out of curiosity, I find myself wondering: do the psychology students tend to get the same grades as everyone else (i.e., mean 67.5) or do they tend to score higher or lower? He emails me the `zeppo.Rdata` file, which I use to pull up the `grades` of those students,

```
load( "./rbook-master/data/zeppo.Rdata" )
print( grades )
```

```
##  [1] 50 60 60 64 66 66 67 69 70 74 76 76 77 79 79 79 81 82 82 89
```

and calculate the mean:

```
mean( grades )
```

```
## [1] 72.3
```

Hm. It *might* be that the psychology students are scoring a bit higher than normal: that sample mean of $\bar{X}$ = 72.3 is a fair bit higher than the hypothesised population mean of μ=67.5, but on the other hand, a sample size of N=20 isn't all that big. Maybe it's pure chance.

To answer the question, it helps to be able to write down what it is that I think I know. Firstly, I know that the sample mean is $\bar{X}$ =72.3. If I'm willing to assume that the psychology students have the same standard deviation as the rest of the class then I can say that the population standard deviation is σ=9.5. I'll also assume that since Dr Zeppo is grading to a curve, the psychology student grades are normally distributed.

Next, it helps to be clear about what I want to learn from the data. In this case, my research hypothesis relates to the *population* mean μ for the psychology student grades, which is unknown. Specifically, I want to know if μ=67.5 or not. Given that this is what I know, can we devise a hypothesis test to solve our problem? The data, along with the hypothesised distribution from which they are thought to arise, are shown in Figure 13.1. Not entirely obvious what the right answer is, is it? For this, we are going to need some statistics.

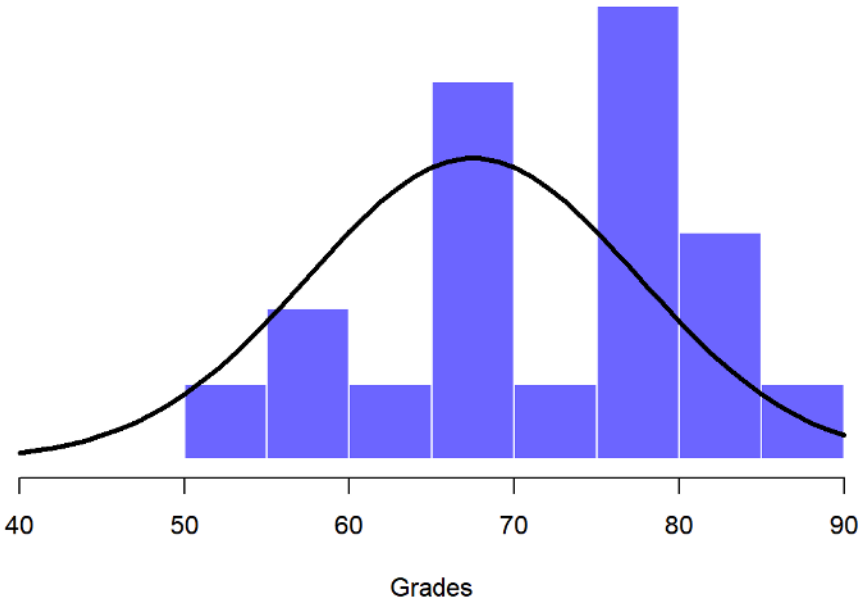


Figure 13.1: The theoretical distribution (solid line) from which the psychology student grades (grey bars) are supposed to have been generated.

11.1.2 Constructing the hypothesis test

The first step in constructing a hypothesis test is to be clear about what the null and alternative hypotheses are. This isn't too hard to do. Our null hypothesis, H_0, is that the true population mean μ for psychology student grades is 67.5%; and our alternative hypothesis is that the population mean *isn't* 67.5%. If we write this in mathematical notation, these hypotheses become,

$$H_0: \mu = 67.5$$

$$H_1: \mu \neq 67.5$$

though to be honest this notation doesn't add much to our understanding of the problem, it's just a compact way of writing down what we're trying to learn from the data. The null hypotheses H_0 and the alternative hypothesis H_1 for our test are both illustrated in Figure 13.2. In addition to providing us with these hypotheses, the scenario outlined above provides us with a fair amount of background knowledge that might be useful. Specifically, there are two special pieces of information that we can add:

1 The psychology grades are normally distributed. 1 The true standard deviation of these scores σ is known to be 9.5.

For the moment, we'll act as if these are absolutely trustworthy facts. In real life, this kind of absolutely trustworthy background knowledge doesn't exist, and so if we want to rely on these facts we'll just have make the *assumption* that these things are true. However, since these assumptions may or may not be warranted, we might need to check them. For now though, we'll keep things simple.

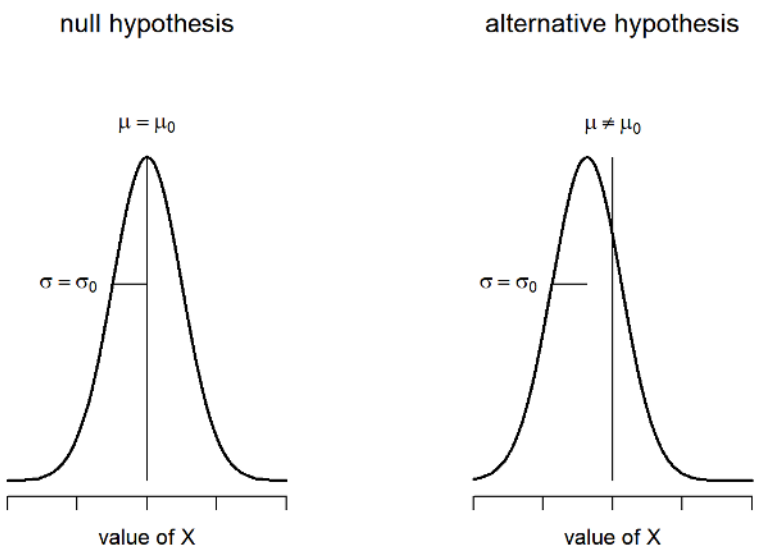


Figure 13.2: Graphical illustration of the null and alternative hypotheses assumed by the one sample z-test (the two sided version, that is). The null and alternative hypotheses both assume that the population distribution is normal, and additionally assumes that the population standard deviation is known (fixed at some value σ_0). The null hypothesis (left) is that the population mean μ is equal to some specified value μ_0. The alternative hypothesis is that the population mean differs from this value, $\mu \neq \mu_0$.

The next step is to figure out what we would be a good choice for a diagnostic test statistic; something that would help us discriminate between H_0 and H_1. Given that the hypotheses all refer to the population mean μ, you'd feel pretty confident that the sample mean $\bar{X}$ would be a pretty useful place to start. What we could do, is look at the difference between the sample mean $\bar{X}$ and the value that the null hypothesis predicts for the population mean. In our example, that would mean we calculate $\bar{X}$ - 67.5. More generally, if we let μ_0 refer to the value that the null hypothesis claims is our population mean, then we'd want to calculate

$$\bar{X} - \mu_0$$

If this quantity equals or is very close to 0, things are looking good for the null hypothesis. If this quantity is a long way away from 0, then it's looking less likely that the null hypothesis is worth retaining. But how far away from zero should it be for us to reject H_0?

To figure that out, we need to be a bit more sneaky, and we'll need to rely on those two pieces of background knowledge that I wrote down previously, namely that the raw data are normally distributed, and we know the value of the population standard deviation σ. If the null hypothesis is actually true, and the true mean is μ_0, then these facts together mean that we know the complete population distribution of the data: a normal distribution with mean μ_0 and standard deviation σ. Adopting the notation from Section 9.5, a statistician might write this as:

$$X \sim \text{Normal}(\mu_0, \sigma^2)$$

Okay, if that's true, then what can we say about the distribution of $\bar{X}$? Well, as we discussed earlier (see Section 10.3.3), the sampling distribution of the mean $\bar{X}$ is also normal, and has mean μ. But the standard deviation of this sampling distribution SE ($\bar{X}$), which is called the *standard error of the mean*, is

$$\text{SE}(\bar{X}) = \frac{\sigma}{\sqrt{N}}$$

In other words, if the null hypothesis is true then the sampling distribution of the mean can be written as follows:

$$\bar{X} \sim \text{Normal}(\mu_0, \text{SE}(\bar{X}))$$

Now comes the trick. What we can do is convert the sample mean $\bar{X}$ into a standard score (Section 5.6). This is conventionally written as z, but for now I'm going to refer to it as $z_{\bar{X}}$. (The reason for using this expanded notation is to help you remember that we're calculating standardised version of a sample mean, *not* a standardised version of a single observation, which is what a z-score usually refers to). When we do so, the z-score for our sample mean is

$$z_{\bar{X}} = \frac{\bar{X} - \mu_0}{SE(\bar{X})}$$

or, equivalently

$$z_{\bar{X}} = \frac{\bar{X} - \mu_0}{\sigma/\sqrt{N}}$$

This z-score is our test statistic. The nice thing about using this as our test statistic is that like all z-scores, it has a standard normal distribution:

$$z_{\bar{X}} \sim \text{Normal}(0,1)$$

(again, see Section 5.6 if you've forgotten why this is true). In other words, regardless of what scale the original data are on, the z-statistic iteself always has the same interpretation: it's equal to the number of standard errors that separate the observed sample mean $\bar{X}$ from the population mean μ_0 predicted by the null hypothesis. Better yet, regardless of what the population parameters for the raw scores actually are, the 5% critical regions for z-test are always the same, as illustrated in Figures 13.4 and 13.3. And what this meant, way back in the days where people did all their statistics by hand, is that someone could publish a table like this:

desired α level	two-sided test	one-sided test
.1	1.644854	1.281552
.05	1.959964	1.644854
.01	2.575829	2.326348
.001	3.290527	3.090232

which in turn meant that researchers could calculate their z-statistic by hand, and then look up the critical value in a text book. That was an incredibly handy thing to be able to do back then, but it's kind of unnecessary these days, since it's trivially easy to do it with software like R.

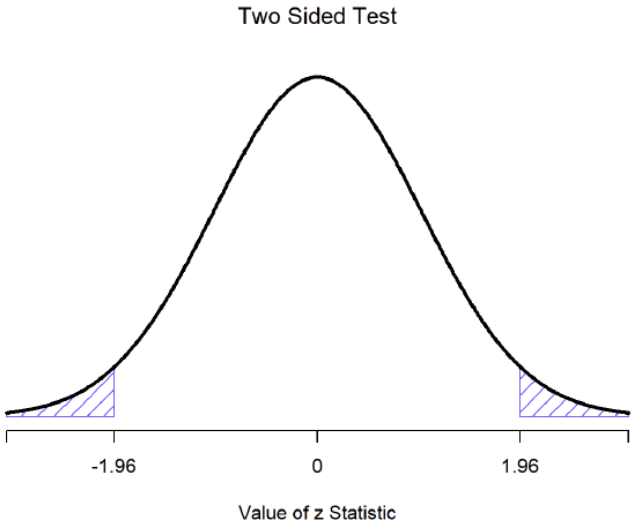


Figure 13.3: Rejection regions for the two-sided z-test

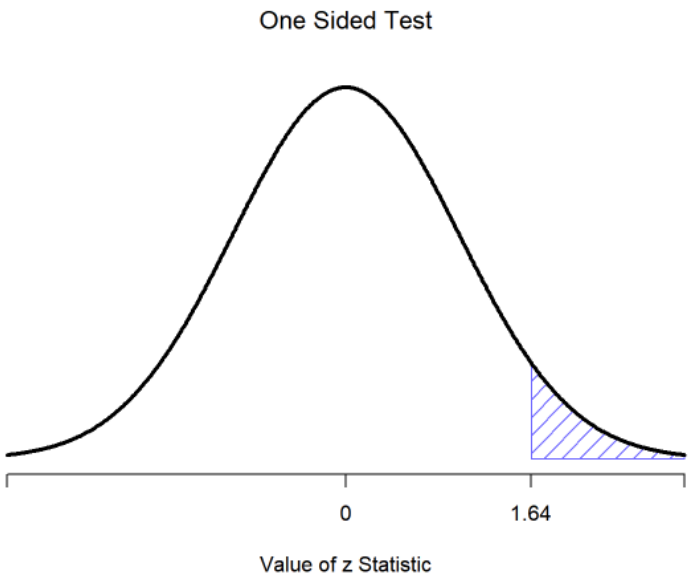


Figure 13.4: Rejection regions for the one-sided z-test

11.1.3 worked example using R

Now, as I mentioned earlier, the z-test is almost never used in practice. It's so rarely used in real life that the basic installation of R doesn't have a built in function for it. However, the test is so incredibly simple that it's really easy to do one manually. Let's go back to the data from Dr Zeppo's class. Having loaded the `grades` data, the first thing I need to do is calculate the sample mean:

```
sample.mean <- mean( grades )
print( sample.mean )
```

```
## [1] 72.3
```

Then, I create variables corresponding to known population standard deviation (σ=9.5), and the value of the population mean that the null hypothesis specifies (μ_0=67.5):

```
mu.null <- 67.5
sd.true <- 9.5
```

Let's also create a variable for the sample size. We could count up the number of observations ourselves, and type `N <- 20` at the command prompt, but counting is tedious and repetitive. Let's get R to do the tedious repetitive bit by using the `length()` function, which tells us how many elements there are in a vector:

```
N <- length( grades )
print( N )
```

```
## [1] 20
```

Next, let's calculate the (true) standard error of the mean:

```
sem.true <- sd.true / sqrt(N)
print(sem.true)
```

```
## [1] 2.124265
```

And finally, we calculate our z-score:

```
z.score <- (sample.mean - mu.null) / sem.true
print( z.score )
```

```
## [1] 2.259606
```

At this point, we would traditionally look up the value 2.26 in our table of critical values. Our original hypothesis was two-sided (we didn't really have any theory about whether psych students would be better or worse at statistics than other students) so our hypothesis test is two-sided (or two-tailed) also. Looking at the little table that I showed earlier, we can see that 2.26 is bigger than the critical value of 1.96 that would be required to be significant at α=.05, but smaller than the value of 2.58 that would be required to be significant at a level of α=.01. Therefore, we can conclude that we have a significant effect, which we might write up by saying something like this:

With a mean grade of 73.2 in the sample of psychology students, and assuming a true population standard deviation of 9.5, we can conclude that the psychology students have significantly different statistics scores to the class average (z=2.26, N=20, p<.05).

However, what if want an exact p-value? Well, back in the day, the tables of critical values were huge, and so you could look up your actual z-value, and find the smallest value of α for which your data would be significant (which, as discussed earlier, is the very definition of a p-value). However, looking things up in books is tedious, and typing things into computers is awesome. So let's do it using R instead. Now, notice that the α level of a z-test (or any other test, for that matter) defines the total area "under the curve" for the critical region, right? That is, if we set α=.05 for a two-sided test, then the critical region is set up such that the area under the curve for the critical region is .05. And, for the z-test, the critical value of 1.96 is chosen that way because the area in the lower tail (i.e., below −1.96) is exactly .025 and the area under the upper tail (i.e., above 1.96) is exactly .025. So, since our observed z-statistic is 2.26, why not calculate the area under the curve below −2.26 or above 2.26? In R we can calculate this using the `pnorm()` function. For the upper tail:

```
upper.area <- pnorm( q = z.score, lower.tail = FALSE )
print( upper.area )
```

```
## [1] 0.01192287
```

The `lower.tail = FALSE` is me telling R to calculate the area under the curve from 2.26 *and upwards*. If I'd told it that `lower.tail = TRUE`, then R would calculate the area from 2.26 *and below*, and it would give me an answer 0.9880771. Alternatively, to calculate the area from −2.26 and below, we get

```
lower.area <- pnorm( q = -z.score, lower.tail = TRUE )
print( lower.area )
```

```
## [1] 0.01192287
```

Thus we get our p-value:

```
p.value <- lower.area + upper.area
print( p.value )
```

```
## [1] 0.02384574
```

11.1.4 Assumptions of the z-test

As I've said before, all statistical tests make assumptions. Some tests make reasonable assumptions, while other tests do not. The test I've just described – the one sample z-test – makes three basic assumptions. These are:

- *Normality*. As usually described, the z-test assumes that the true population distribution is normal.[186] is often pretty reasonable, and not only that, it's an assumption that we can check if we feel worried about it (see Section 13.9).
- *Independence*. The second assumption of the test is that the observations in your data set are not correlated with each other, or related to each other in some funny way. This isn't as easy to check statistically: it relies a bit on good experimetal design. An obvious (and stupid) example of something that violates this assumption is a data set where you "copy" the same observation over and over again in your data file: so you end up with a massive "sample size", consisting of only one genuine observation. More realistically, you have to ask yourself if it's really plausible to imagine that each observation is a completely random sample from the population that you're interested in. In practice, this assumption is never met; but we try our best to design studies that minimise the problems of correlated data.
- *Known standard deviation*. The third assumption of the z-test is that the true standard deviation of the population is known to the researcher. This is just stupid. In no real world data analysis problem do you know the standard deviation σ of some population, but are completely ignorant about the mean μ. In other words, this assumption is *always* wrong.

In view of the stupidity of assuming that σ is known, let's see if we can live without it. This takes us out of the dreary domain of the z-test, and into the magical kingdom of the t-test, with unicorns and fairies and leprechauns, and um…

This page titled 11.1: The one-sample z-test is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **13.1: The one-sample z-test** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

11.2: The One-sample t-test

After some thought, I decided that it might not be safe to assume that the psychology student grades necessarily have the same standard deviation as the other students in Dr Zeppo's class. After all, if I'm entertaining the hypothesis that they don't have the same mean, then why should I believe that they absolutely have the same standard deviation? In view of this, I should really stop assuming that I know the true value of σ. This violates the assumptions of my z-test, so in one sense I'm back to square one. However, it's not like I'm completely bereft of options. After all, I've still got my raw data, and those raw data give me an *estimate* of the population standard deviation:

```
sd( grades )
```

```
## [1] 9.520615
```

In other words, while I can't say that I know that σ=9.5, I *can* say that $\hat{\sigma}$=9.52.

Okay, cool. The obvious thing that you might think to do is run a z-test, but using the estimated standard deviation of 9.52 instead of relying on my assumption that the true standard deviation is 9.5. So, we could just type this new number into R and out would come the answer. And you probably wouldn't be surprised to hear that this would still give us a significant result. This approach is close, but it's not *quite* correct. Because we are now relying on an *estimate* of the population standard deviation, we need to make some adjustment for the fact that we have some uncertainty about what the true population standard deviation actually is. Maybe our data are just a fluke … maybe the true population standard deviation is 11, for instance. But if that were actually true, and we ran the z-test assuming σ=11, then the result would end up being *non-significant*. That's a problem, and it's one we're going to have to address.

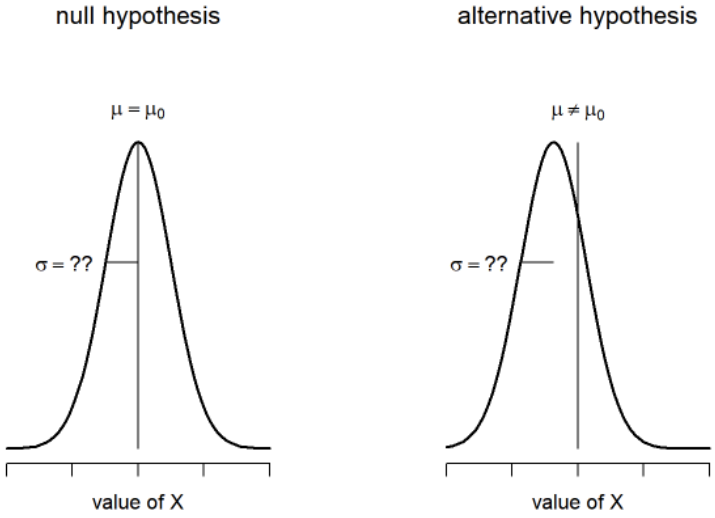


Figure 13.4: Graphical illustration of the null and alternative hypotheses assumed by the (two sided) one sample t-test. Note the similarity to the z-test. The null hypothesis is that the population mean μ is equal to some specified value μ_0, and the alternative hypothesis is that it is not. Like the z-test, we assume that the data are normally distributed; but we do not assume that the population standard deviation σ is known in advance.

11.2.1 Introducing the t-test

This ambiguity is annoying, and it was resolved in 1908 by a guy called William Sealy Gosset (Student 1908), who was working as a chemist for the Guinness brewery at the time (see Box 1987). Because Guinness took a dim view of its employees publishing statistical analysis (apparently they felt it was a trade secret), he published the work under the pseudonym "A Student", and to this day, the full name of the t-test is actually ***Student's t-test***. The key thing that Gosset figured out is how we should accommodate the fact that we aren't completely sure what the true standard deviation is.[187] The answer is that it subtly changes the sampling distribution. In the t-test, our test statistic (now called a t-statistic) is calculated in exactly the same way I mentioned above. If our

null hypothesis is that the true mean is μ, but our sample has mean $\bar{X}$ and our estimate of the population standard deviation is $\hat{\sigma}$, then our t statistic is:

$$t = \frac{\bar{X} - \mu}{\hat{\sigma}/\sqrt{N}}$$

The only thing that has changed in the equation is that instead of using the known true value σ, we use the estimate $\hat{\sigma}$ And if this estimate has been constructed from N observations, then the sampling distribution turns into a t-distribution with N−1 ***degrees of freedom*** (df). The t distribution is very similar to the normal distribution, but has "heavier" tails, as discussed earlier in Section 9.6 and illustrated in Figure 13.5. Notice, though, that as df gets larger, the t-distribution starts to look identical to the standard normal distribution. This is as it should be: if you have a sample size of N=70,000,000 then your "estimate" of the standard deviation would be pretty much perfect, right? So, you should expect that for large N, the t-test would behave exactly the same way as a z-test. And that's exactly what happens!

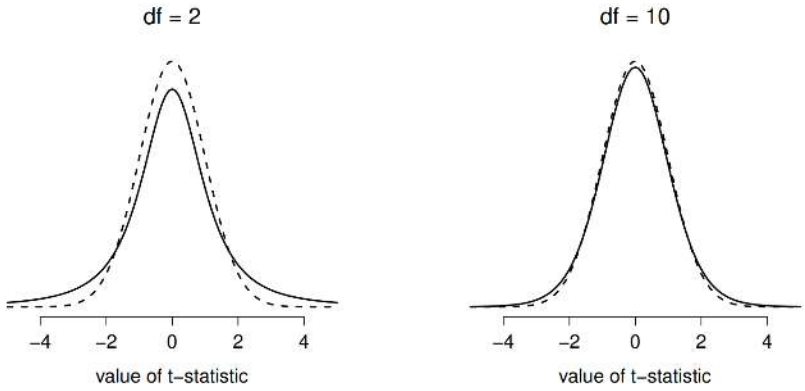


Figure 13.5: The t distribution with 2 degrees of freedom (left) and 10 degrees of freedom (right), with a standard normal distribution (i.e., mean 0 and std dev 1) plotted as dotted lines for comparison purposes. Notice that the t distribution has heavier tails (higher kurtosis) than the normal distribution; this effect is quite exaggerated when the degrees of freedom are very small, but negligible for larger values. In other words, for large df the t distribution is essentially identical to a normal distribution.

11.2.2 Doing the test in R

As you might expect, the mechanics of the t-test are almost identical to the mechanics of the z-test. So there's not much point in going through the tedious exercise of showing you how to do the calculations using low level commands: it's pretty much identical to the calculations that we did earlier, except that we use the estimated standard deviation (i.e., something like `se.est <- sd(grades)`), and then we test our hypothesis using the t distribution rather than the normal distribution (i.e. we use `pt()` rather than `pnorm()`. And so instead of going through the calculations in tedious detail for a second time, I'll jump straight to showing you how t-tests are actually done in practice.

The situation with t-tests is very similar to the one we encountered with chi-squared tests in Chapter 12. R comes with one function called `t.test()` that is very flexible (it can run lots of different kinds of t-tests) and is somewhat terse (the output is quite compressed). Later on in the chapter I'll show you how to use the `t.test()` function (Section 13.7), but to start out with I'm going to rely on some simpler functions in the `lsr` package. Just like last time, what I've done is written a few simpler functions, each of which does only one thing. So, if you want to run a one-sample t-test, use the `oneSampleTTest()` function! It's pretty straightforward to use: all you need to do is specify `x`, the variable containing the data, and `mu`, the true population mean according to the null hypothesis. All you need to type is this:

```
library(lsr)

oneSampleTTest( x=grades, mu=67.5 )
```

```
##
##    One sample t-test
##
## Data variable:   grades
##
## Descriptive statistics:
##             grades
##    mean     72.300
##    std dev.  9.521
##
## Hypotheses:
##    null:        population mean equals 67.5
##    alternative: population mean not equal to 67.5
##
## Test results:
##    t-statistic:  2.255
##    degrees of freedom:  19
##    p-value:  0.036
##
## Other information:
##    two-sided 95% confidence interval:  [67.844, 76.756]
##    estimated effect size (Cohen's d):  0.504
```

Easy enough. Now lets go through the output. Just like we saw in the last chapter, I've written the functions so that the output is pretty verbose. It tries to describe in a lot of detail what its actually done:

```
 One sample t-test

Data variable:   grades

Descriptive statistics:
            grades
   mean     72.300
   std dev.  9.521

Hypotheses:
   null:        population mean equals 67.5
   alternative: population mean not equal to 67.5

Test results:
   t-statistic:  2.255
   degrees of freedom:  19
   p-value:  0.036

Other information:
   two-sided 95% confidence interval:  [67.844, 76.756]
   estimated effect size (Cohen's d):  0.504
```

Reading this output from top to bottom, you can see it's trying to lead you through the data analysis process. The first two lines tell you what kind of test was run and what data were used. It then gives you some basic information about the sample: specifically, the sample mean and standard deviation of the data. It then moves towards the inferential statistics part. It starts by telling you what the null and alternative hypotheses were, and then it reports the results of the test: the t-statistic, the degrees of freedom, and the p-value. Finally, it reports two other things you might care about: the confidence interval for the mean, and a measure of effect size (we'll talk more about effect sizes later).

So that seems straightforward enough. Now what do we *do* with this output? Well, since we're pretending that we actually care about my toy example, we're overjoyed to discover that the result is statistically significant (i.e. p value below 0.05). We could report the result by saying something like this:

With a mean grade of 72.3, the psychology students scored slightly higher than the average grade of 67.5 (t(19)=2.25, p<.05); the 95% confidence interval is [67.8, 76.8].

where t(19) is shorthand notation for a t-statistic that has 19 degrees of freedom. That said, it's often the case that people don't report the confidence interval, or do so using a much more compressed form than I've done here. For instance, it's not uncommon to see the confidence interval included as part of the stat block, like this:

t(19)=2.25, p<.05, CI95=[67.8,76.8]

With that much jargon crammed into half a line, you know it must be really smart.[188]

11.2.3 Assumptions of the one sample t-test

Okay, so what assumptions does the one-sample t-test make? Well, since the t-test is basically a z-test with the assumption of known standard deviation removed, you shouldn't be surprised to see that it makes the same assumptions as the z-test, minus the one about the known standard deviation. That is

- *Normality*. We're still assuming that the the population distribution is normal^[A technical comment… in the same way that we can weaken the assumptions of the z-test so that we're only talking about the sampling distribution, we *can* weaken the t test assumptions so that we don't have to assume normality of the population. However, for the t-test, it's trickier to do this. As before, we can replace the assumption of population normality with an assumption that the sampling distribution of $\bar{X}$ is normal. However, remember that we're also relying on a sample estimate of the standard deviation; and so we also require the sampling distribution of ^σ to be chi-square. That makes things nastier, and this version is rarely used in practice: fortunately, if the population is normal, then both of these two assumptions are met., and as noted earlier, there are standard tools that you can use to check to see if this assumption is met (Section 13.9), and other tests you can do in it's place if this assumption is violated (Section 13.10).
- *Independence*. Once again, we have to assume that the observations in our sample are generated independently of one another. See the earlier discussion about the z-test for specifics (Section 13.1.4).

Overall, these two assumptions aren't terribly unreasonable, and as a consequence the one-sample t-test is pretty widely used in practice as a way of comparing a sample mean against a hypothesised population mean.

This page titled 11.2: The One-sample t-test is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **13.2: The One-sample t-test** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

11.3: The Independent Samples t-test (Student Test)

Although the one sample t-test has its uses, it's not the most typical example of a t-test[189]. A much more common situation arises when you've got two different groups of observations. In psychology, this tends to correspond to two different groups of participants, where each group corresponds to a different condition in your study. For each person in the study, you measure some outcome variable of interest, and the research question that you're asking is whether or not the two groups have the same population mean. This is the situation that the independent samples t-test is designed for.

11.3.1 data

Suppose we have 33 students taking Dr Harpo's statistics lectures, and Dr Harpo doesn't grade to a curve. Actually, Dr Harpo's grading is a bit of a mystery, so we don't really know anything about what the average grade is for the class as a whole. There are two tutors for the class, Anastasia and Bernadette. There are N_1=15 students in Anastasia's tutorials, and N_2=18 in Bernadette's tutorials. The research question I'm interested in is whether Anastasia or Bernadette is a better tutor, or if it doesn't make much of a difference. Dr Harpo emails me the course grades, in the `harpo.Rdata` file. As usual, I'll load the file and have a look at what variables it contains:

```
load( "./rbook-master/data/harpo.Rdata" )
str(harpo)
```

```
## 'data.frame':    33 obs. of  2 variables:
##  $ grade: num  65 72 66 74 73 71 66 76 69 79 ...
##  $ tutor: Factor w/ 2 levels "Anastasia","Bernadette": 1 2 2 1 1 2 2 2 2 2 ...
```

As we can see, there's a single data frame with two variables, `grade` and `tutor`. The `grade` variable is a numeric vector, containing the grades for all N=33 students taking Dr Harpo's class; the `tutor` variable is a factor that indicates who each student's tutor was. The first six observations in this data set are shown below:

```
head( harpo )
```

```
##   grade      tutor
## 1    65  Anastasia
## 2    72 Bernadette
## 3    66 Bernadette
## 4    74  Anastasia
## 5    73  Anastasia
## 6    71 Bernadette
```

We can calculate means and standard deviations, using the `mean()` and `sd()` functions. Rather than show the R output, here's a nice little summary table:

	mean	std dev	N
Anastasia's students	74.53	9.00	15
Bernadette's students	69.06	5.77	18

To give you a more detailed sense of what's going on here, I've plotted histograms showing the distribution of grades for both tutors (Figure 13.6 and 13.7). Inspection of these histograms suggests that the students in Anastasia's class may be getting slightly better grades on average, though they also seem a little more variable.

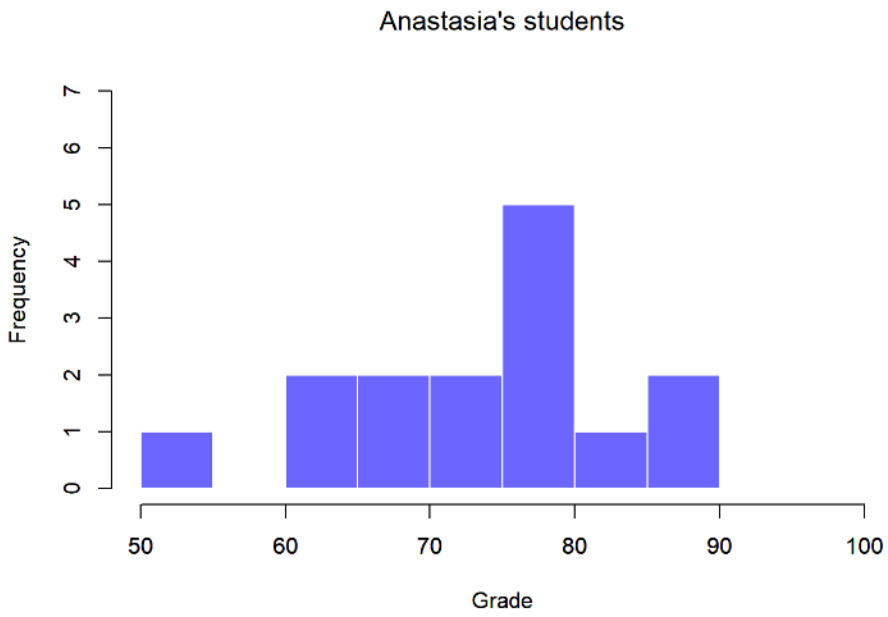


Figure 13.6: Histogram showing the overall distribution of grades for students in Anastasia's class

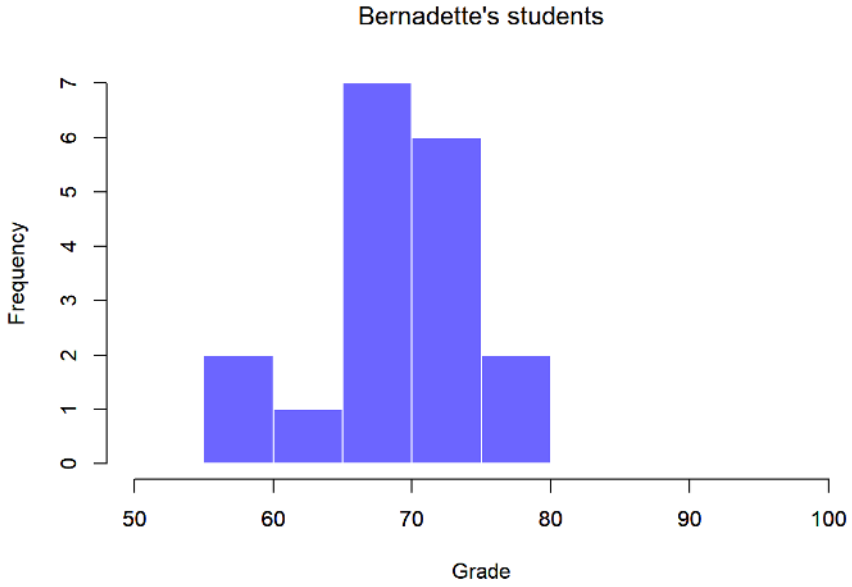


Figure 13.7: Histogram showing the overall distribution of grades for students in Bernadette's class

Here is a simpler plot showing the means and corresponding confidence intervals for both groups of students (Figure 13.8).

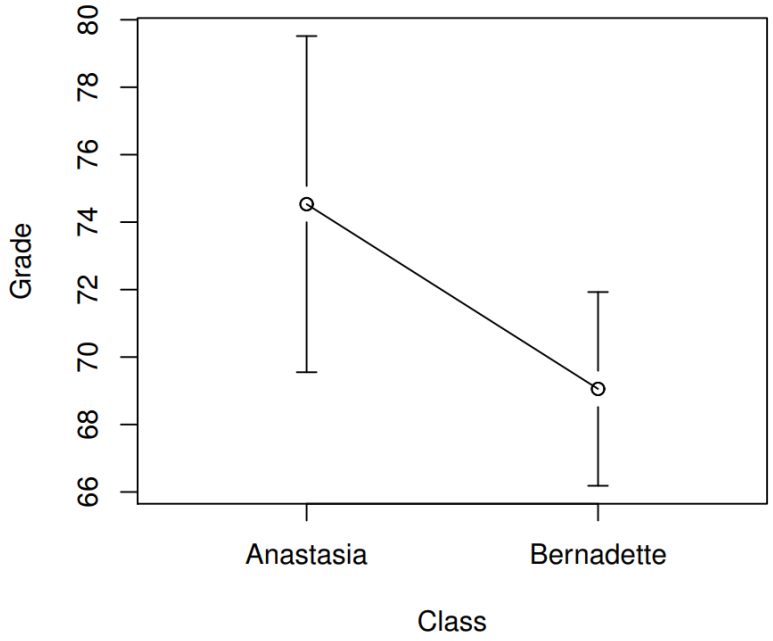


Figure 13.8: Plots showing the mean grade for the students in Anastasia's and Bernadette's tutorials. Error bars depict 95% confidence intervals around the mean. On the basis of visual inspection, it does look like there's a real difference between the groups, though it's hard to say for sure.

11.3.2 Introducing the test

The ***independent samples t-test*** comes in two different forms, Student's and Welch's. The original Student t-test – which is the one I'll describe in this section – is the simpler of the two, but relies on much more restrictive assumptions than the Welch t-test. Assuming for the moment that you want to run a two-sided test, the goal is to determine whether two "independent samples" of data are drawn from populations with the same mean (the null hypothesis) or different means (the alternative hypothesis). When we say "independent" samples, what we really mean here is that there's no special relationship between observations in the two samples. This probably doesn't make a lot of sense right now, but it will be clearer when we come to talk about the paired samples t-test later on. For now, let's just point out that if we have an experimental design where participants are randomly allocated to one of two groups, and we want to compare the two groups' mean performance on some outcome measure, then an independent samples t-test (rather than a paired samples t-test) is what we're after.

Okay, so let's let μ_1 denote the true population mean for group 1 (e.g., Anastasia's students), and μ_2 will be the true population mean for group 2 (e.g., Bernadette's students),[190] and as usual we'll let $\bar{X}_1$ and $\bar{X}_2$ denote the observed sample means for both of these groups. Our null hypothesis states that the two population means are identical ($\mu_1=\mu_2$) and the alternative to this is that they are not ($\mu_1\neq\mu_2$). Written in mathematical-ese, this is…

$$H_0:\mu_1=\mu_2$$

$$H_1:\mu_1\neq\mu_2$$

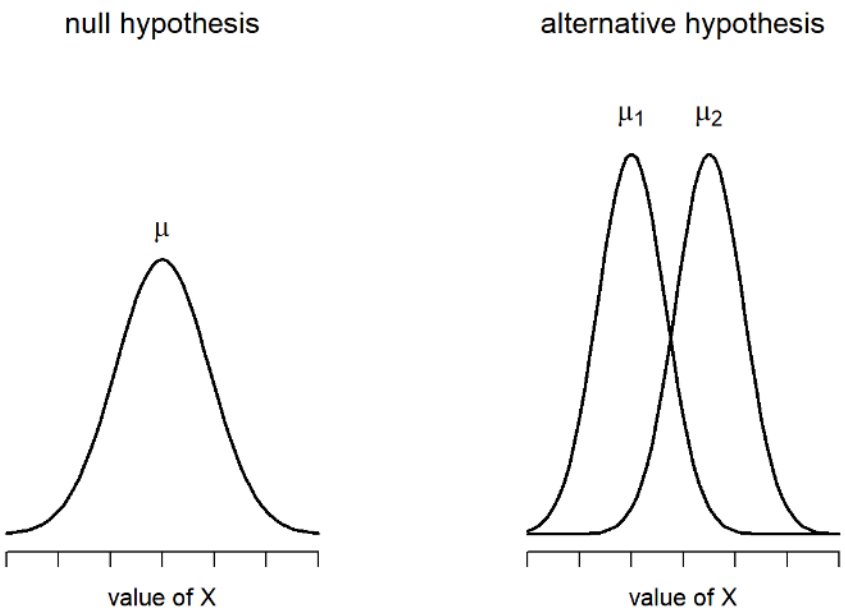


Figure 13.9: Graphical illustration of the null and alternative hypotheses assumed by the Student t-test. The null hypothesis assumes that both groups have the same mean μ, whereas the alternative assumes that they have different means μ_1 and μ_2. Notice that it is assumed that the population distributions are normal, and that, although the alternative hypothesis allows the group to have different means, it assumes they have the same standard deviation

To construct a hypothesis test that handles this scenario, we start by noting that if the null hypothesis is true, then the difference between the population means is *exactly* zero, $\mu_1 - \mu_2 = 0$ As a consequence, a diagnostic test statistic will be based on the difference between the two sample means. Because if the null hypothesis is true, then we'd expect

$$\bar{X}_1 - \bar{X}_2$$

to be *pretty close* to zero. However, just like we saw with our one-sample tests (i.e., the one-sample z-test and the one-sample t-test) we have to be precise about exactly *how close* to zero this difference

$$t = \frac{\bar{X}_1 - \bar{X}_2}{SE}$$

We just need to figure out what this standard error estimate actually is. This is a bit trickier than was the case for either of the two tests we've looked at so far, so we need to go through it a lot more carefully to understand how it works.

11.3.3 "pooled estimate" of the standard deviation

In the original "Student t-test", we make the assumption that the two groups have the same population standard deviation: that is, regardless of whether the population means are the same, we assume that the population standard deviations are identical, $\sigma_1 = \sigma_2$. Since we're assuming that the two standard deviations are the same, we drop the subscripts and refer to both of them as σ. How should we estimate this? How should we construct a single estimate of a standard deviation when we have two samples? The answer is, basically, we average them. Well, sort of. Actually, what we do is take a *weighed* average of the *variance* estimates, which we use as our ***pooled estimate of the variance***. The weight assigned to each sample is equal to the number of observations in that sample, minus 1. Mathematically, we can write this as

$$\omega_1 = N_1 - 1$$

$$\omega_2 = N_2 - 1$$

Now that we've assigned weights to each sample, we calculate the pooled estimate of the variance by taking the weighted average of the two variance estimates, $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$

$$\hat{\sigma}_p^2 = \frac{\omega_1 \hat{\sigma}_1^2 + \omega_2 \hat{\sigma}_2^2}{\omega_1 + \omega_2}$$

Finally, we convert the pooled variance estimate to a pooled standard deviation estimate, by taking the square root. This gives us the following formula for $\hat{\sigma}_p$,

$$\hat{\sigma}_p = \sqrt{\frac{\omega_1 \hat{\sigma}_1^2 + \omega_2 \hat{\sigma}_2^2}{\omega_1 + \omega_2}}$$

And if you mentally substitute ω_1=N1−1 and ω_2=N2−1 into this equation you get a very ugly looking formula; a very ugly formula that actually seems to be the "standard" way of describing the pooled standard deviation estimate. It's not my favourite way of thinking about pooled standard deviations, however.[191]

11.3.4 same pooled estimate, described differently

I prefer to think about it like this. Our data set actually corresponds to a set of N observations, which are sorted into two groups. So let's use the notation X_{ik} to refer to the grade received by the i-th student in the k-th tutorial group: that is, X_{11} is the grade received by the first student in Anastasia's class, X_{21} is her second student, and so on. And we have two separate group means $\bar{X}_1$ and $\bar{X}_2$, which we could "generically" refer to using the notation $\bar{X}_k$, i.e., the mean grade for the k-th tutorial group. So far, so good. Now, since every single student falls into one of the two tutorials, and so we can describe their deviation from the group mean as the difference

$$X_{ik} - \bar{X}_k$$

So why not just use these deviations (i.e., the extent to which each student's grade differs from the mean grade in their tutorial?) Remember, a variance is just the average of a bunch of squared deviations, so let's do that. Mathematically, we could write it like this:

$$\frac{\sum_{ik} (X_{ik} - \bar{X}_k)^2}{N}$$

where the notation "$\sum_{ik}$" is a lazy way of saying "calculate a sum by looking at all students in all tutorials", since each "ik" corresponds to one student.[192] But, as we saw in Chapter 10, calculating the variance by dividing by N produces a biased estimate of the population variance. And previously, we needed to divide by N−1 to fix this. However, as I mentioned at the time, the reason why this bias exists is because the variance estimate relies on the sample mean; and to the extent that the sample mean isn't equal to the population mean, it can systematically bias our estimate of the variance. But this time we're relying on *two* sample means! Does this mean that we've got more bias? Yes, yes it does. And does this mean we now need to divide by N−2 instead of N−1, in order to calculate our pooled variance estimate? Why, yes…

$$\hat{\sigma}_p^2 = \frac{\sum_{ik} (X_{ik} - X_k)^2}{N - 2}$$

Oh, and if you take the square root of this then you get $\hat{\sigma}_P$, the pooled standard deviation estimate. In other words, the pooled standard deviation calculation is nothing special: it's not terribly different to the regular standard deviation calculation.

11.3.5 Completing the test

Regardless of which way you want to think about it, we now have our pooled estimate of the standard deviation. From now on, I'll drop the silly p subscript, and just refer to this estimate as $\hat{\sigma}$. Great. Let's now go back to thinking about the bloody hypothesis test, shall we? Our whole reason for calculating this pooled estimate was that we knew it would be helpful when calculating our *standard error* estimate. But, standard error of *what*? In the one-sample t-test, it was the standard error of the sample mean, SE ($\bar{X}$), and since SE ($\bar{X} = \sigma/\sqrt{N}$ that's what the denominator of our t-statistic looked like. This time around, however, we have *two* sample means. And what we're interested in, specifically, is the the difference between the two $\bar{X}_1$ - $\bar{X}_2$. As a consequence, the standard error that we need to divide by is in fact the ***standard error of the difference*** between means. As long as the two variables really do have the same standard deviation, then our estimate for the standard error is

$$\text{SE}\left(\bar{X}_1 - \bar{X}_2\right) = \hat{\sigma} \sqrt{\frac{1}{N_1} + \frac{1}{N_2}}$$

and our t-statistic is therefore

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\text{SE}\left(\bar{X}_1 - \bar{X}_2\right)}$$

(shocking, isn't it?) as long as the null hypothesis is true, and all of the assumptions of the test are met. The degrees of freedom, however, is slightly different. As usual, we can think of the degrees of freedom to be equal to the number of data points minus the

number of constraints. In this case, we have N observations (N1 in sample 1, and N2 in sample 2), and 2 constraints (the sample means). So the total degrees of freedom for this test are N−2.

11.3.6 Doing the test in R

Not surprisingly, you can run an independent samples t-test using the `t.test()` function (Section 13.7), but once again I'm going to start with a somewhat simpler function in the `lsr` package. That function is unimaginatively called `independentSamplesTTest()`. First, recall that our data look like this:

```
head( harpo )
```

```
##   grade     tutor
## 1    65  Anastasia
## 2    72 Bernadette
## 3    66 Bernadette
## 4    74  Anastasia
## 5    73  Anastasia
## 6    71 Bernadette
```

The outcome variable for our test is the student `grade`, and the groups are defined in terms of the `tutor` for each class. So you probably won't be too surprised to see that we're going to describe the test that we want in terms of an R formula that reads like this `grade ~ tutor`. The specific command that we need is:

```
independentSamplesTTest(
      formula = grade ~ tutor,  # formula specifying outcome and group variables
      data = harpo,             # data frame that contains the variables
      var.equal = TRUE          # assume that the two groups have the same variance
  )
```

```
##
##    Student's independent samples t-test
##
## Outcome variable:   grade
## Grouping variable:  tutor
##
## Descriptive statistics:
##             Anastasia Bernadette
##    mean        74.533     69.056
##    std dev.     8.999      5.775
##
## Hypotheses:
##    null:        population means equal for both groups
##    alternative: different population means in each group
##
## Test results:
##    t-statistic:  2.115
##    degrees of freedom:  31
##    p-value:  0.043
##
## Other information:
##    two-sided 95% confidence interval:  [0.197, 10.759]
##    estimated effect size (Cohen's d):  0.74
```

The first two arguments should be familiar to you. The first one is the formula that tells R what variables to use and the second one tells R the name of the data frame that stores those variables. The third argument is not so obvious. By saying `var.equal = TRUE` , what we're really doing is telling R to use the *Student* independent samples t-test. More on this later. For now, lets ignore that bit and look at the output:

The output has a very familiar form. First, it tells you what test was run, and it tells you the names of the variables that you used. The second part of the output reports the sample means and standard deviations for both groups (i.e., both tutorial groups). The third section of the output states the null hypothesis and the alternative hypothesis in a fairly explicit form. It then reports the test results: just like last time, the test results consist of a t-statistic, the degrees of freedom, and the p-value. The final section reports two things: it gives you a confidence interval, and an effect size. I'll talk about effect sizes later. The confidence interval, however, I should talk about now.

It's pretty important to be clear on what this confidence interval actually refers to: it is a confidence interval for the *difference* between the group means. In our example, Anastasia's students had an average grade of 74.5, and Bernadette's students had an average grade of 69.1, so the difference between the two sample means is 5.4. But of course the difference between population means might be bigger or smaller than this. The confidence interval reported by the `independentSamplesTTest()` function tells you that there's a 95% chance that the true difference between means lies between 0.2 and 10.8.

In any case, the difference between the two groups is significant (just barely), so we might write up the result using text like this:

The mean grade in Anastasia's class was 74.5% (std dev = 9.0), whereas the mean in Bernadette's class was 69.1% (std dev = 5.8). A Student's independent samples t-test showed that this 5.4% difference was significant ($t(31)=2.1$, $p<.05$, $CI_{95}=[0.2,10.8]$, $d=.74$), suggesting that a genuine difference in learning outcomes has occurred.

Notice that I've included the confidence interval and the effect size in the stat block. People don't always do this. At a bare minimum, you'd expect to see the t-statistic, the degrees of freedom and the p value. So you should include something like this at a minimum: $t(31)=2.1$, $p<.05$. If statisticians had their way, everyone would also report the confidence interval and probably the effect size measure too, because they are useful things to know. But real life doesn't always work the way statisticians want it to:

you should make a judgment based on whether you think it will help your readers, and (if you're writing a scientific paper) the editorial standard for the journal in question. Some journals expect you to report effect sizes, others don't. Within some scientific communities it is standard practice to report confidence intervals, in other it is not. You'll need to figure out what your audience expects. But, just for the sake of clarity, if you're taking my class: my default position is that it's usually worth includng the effect size, but don't worry about the confidence interval unless the assignment asks you to or implies that you should.

11.3.7 Positive and negative t values

Before moving on to talk about the assumptions of the t-test, there's one additional point I want to make about the use of t-tests in practice. The first one relates to the sign of the t-statistic (that is, whether it is a positive number or a negative one). One very common worry that students have when they start running their first t-test is that they often end up with negative values for the t-statistic, and don't know how to interpret it. In fact, it's not at all uncommon for two people working independently to end up with R outputs that are almost identical, except that one person has a negative t values and the other one has a positive t value. Assuming that you're running a two-sided test, then the p-values will be identical. On closer inspection, the students will notice that the confidence intervals also have the opposite signs. This is perfectly okay: whenever this happens, what you'll find is that the two versions of the R output arise from slightly different ways of running the t-test. What's happening here is very simple. The t-statistic that R is calculating here is always of the form

$$t = \frac{(\text{mean 1}) - (\text{mean 2})}{(\text{SE})}$$

If "mean 1" is larger than "mean 2" the t statistic will be positive, whereas if "mean 2" is larger then the t statistic will be negative. Similarly, the confidence interval that R reports is the confidence interval for the difference "(mean 1) minus (mean 2)", which will be the reverse of what you'd get if you were calculating the confidence interval for the difference "(mean 2) minus (mean 1)".

Okay, that's pretty straightforward when you think about it, but now consider our t-test comparing Anastasia's class to Bernadette's class. Which one should we call "mean 1" and which one should we call "mean 2". It's arbitrary. However, you really do need to designate one of them as "mean 1" and the other one as "mean 2". Not surprisingly, the way that R handles this is also pretty arbitrary. In earlier versions of the book I used to try to explain it, but after a while I gave up, because it's not really all that important, and to be honest I can never remember myself. Whenever I get a significant t-test result, and I want to figure out which mean is the larger one, I don't try to figure it out by looking at the t-statistic. Why would I bother doing that? It's foolish. It's easier just look at the actual group means, since the R output actually shows them!

Here's the important thing. Because it really doesn't matter what R printed out, I usually try to *report* the t-statistic in such a way that the numbers match up with the text. Here's what I mean… suppose that what I want to write in my report is "Anastasia's class had higher grades than Bernadette's class". The phrasing here implies that Anastasia's group comes first, so it makes sense to report the t-statistic as if Anastasia's class corresponded to group 1. If so, I would write

Anastasia's class had higher grades than Bernadette's class ($t(31) = 2.1, p = .04$).

(I wouldn't actually emphasise the word "higher" in real life, I'm just doing it to emphasise the point that "higher" corresponds to positive t values). On the other hand, suppose the phrasing I wanted to use has Bernadette's class listed first. If so, it makes more sense to treat her class as group 1, and if so, the write up looks like this:

Bernadette's class had lower grades than Anastasia's class ($t(31) = -2.1, p = .04$).

Because I'm talking about one group having "lower" scores this time around, it is more sensible to use the negative form of the t-statistic. It just makes it read more cleanly.

One last thing: please note that you *can't* do this for other types of test statistics. It works for t-tests, but it wouldn't be meaningful for chi-square testsm F-tests or indeed for most of the tests I talk about in this book. So don't overgeneralise this advice! I'm really just talking about t-tests here and nothing else!

11.3.8 Assumptions of the test

As always, our hypothesis test relies on some assumptions. So what are they? For the Student t-test there are three assumptions, some of which we saw previously in the context of the one sample t-test (see Section 13.2.3):

- *Normality*. Like the one-sample t-test, it is assumed that the data are normally distributed. Specifically, we assume that both groups are normally distributed. In Section 13.9 we'll discuss how to test for normality, and in Section 13.10 we'll discuss possible solutions.
- *Independence*. Once again, it is assumed that the observations are independently sampled. In the context of the Student test this has two aspects to it. Firstly, we assume that the observations within each sample are independent of one another (exactly the same as for the one-sample test). However, we also assume that there are no cross-sample dependencies. If, for instance, it turns out that you included some participants in both experimental conditions of your study (e.g., by accidentally allowing the same person to sign up to different conditions), then there are some cross sample dependencies that you'd need to take into account.
- *Homogeneity of variance* (also called "homoscedasticity"). The third assumption is that the population standard deviation is the same in both groups. You can test this assumption using the Levene test, which I'll talk about later on in the book (Section 14.7). However, there's a very simple remedy for this assumption, which I'll talk about in the next section.

This page titled 11.3: The Independent Samples t-test (Student Test) is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **13.3: The Independent Samples t-test (Student Test)** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

11.4: The Independent Samples t-test (Welch Test)

The biggest problem with using the Student test in practice is the third assumption listed in the previous section: it assumes that both groups have the same standard deviation. This is rarely true in real life: if two samples don't have the same means, why should we expect them to have the same standard deviation? There's really no reason to expect this assumption to be true. We'll talk a little bit about how you can check this assumption later on because it does crop up in a few different places, not just the t-test. But right now I'll talk about a different form of the t-test (Welch 1947) that does not rely on this assumption. A graphical illustration of what the ***Welch t test*** assumes about the data is shown in Figure 13.10, to provide a contrast with the Student test version in Figure 13.9. I'll admit it's a bit odd to talk about the cure before talking about the diagnosis, but as it happens the Welch test is the default t-test in R, so this is probably the best place to discuss it.

The Welch test is very similar to the Student test. For example, the t-statistic that we use in the Welch test is calculated in much the same way as it is for the Student test. That is, we take the difference between the sample means, and then divide it by some estimate of the standard error of that difference:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\mathrm{SE}\left(\bar{X}_1 - \bar{X}_2\right)}$$

The main difference is that the standard error calculations are different. If the two populations have different standard deviations, then it's a complete nonsense to try to calculate a pooled standard deviation estimate, because you're averaging apples and oranges.[193] But you can still estimate the standard error of the difference between sample means; it just ends up looking different:

$$\mathrm{SE}\left(\bar{X}_1 - \bar{X}_2\right) = \sqrt{\frac{\hat{\sigma}_1^2}{N_1} + \frac{\hat{\sigma}_2^2}{N_2}}$$

The reason why it's calculated this way is beyond the scope of this book. What matters for our purposes is that the t-statistic that comes out of the Welch test is actually somewhat different to the one that comes from the Student test.

The second difference between Welch and Student is that the degrees of freedom are calculated in a very different way. In the Welch test, the "degrees of freedom" doesn't have to be a whole number any more, and it doesn't correspond all that closely to the "number of data points minus the number of constraints" heuristic that I've been using up to this point. The degrees of freedom are, in fact…

$$\mathrm{df} = \frac{\left(\hat{\sigma}_1^2/N_1 + \hat{\sigma}_2^2/N_2\right)^2}{\left(\hat{\sigma}_1^2/N_1\right)^2/(N_1-1) + \left(\hat{\sigma}_2^2/N_2\right)^2/(N_2-1)}$$

… which is all pretty straightforward and obvious, right? Well, perhaps not. It doesn't really matter for out purposes. What matters is that you'll see that the "df" value that pops out of a Welch test tends to be a little bit smaller than the one used for the Student test, and it doesn't have to be a whole number.

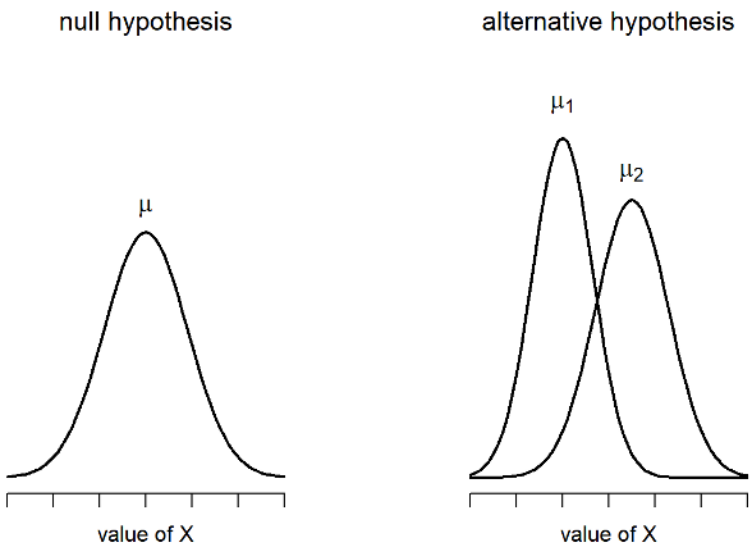


Figure 13.10: Graphical illustration of the null and alternative hypotheses assumed by the Welch t-test. Like the Student test we assume that both samples are drawn from a normal population; but the alternative hypothesis no longer requires the two populations to have equal variance.

11.4.1 Doing the test in R

To run a Welch test in R is pretty easy. All you have to do is not bother telling R to assume equal variances. That is, you take the command we used to run a Student's t-test and drop the `var.equal = TRUE` bit. So the command for a Welch test becomes:

```
independentSamplesTTest(
      formula = grade ~ tutor,  # formula specifying outcome and group variables
      data = harpo              # data frame that contains the variables
  )
```

```
##
##    Welch's independent samples t-test
##
## Outcome variable:   grade
## Grouping variable:  tutor
##
## Descriptive statistics:
##             Anastasia Bernadette
##    mean        74.533     69.056
##    std dev.     8.999      5.775
##
## Hypotheses:
##    null:        population means equal for both groups
##    alternative: different population means in each group
##
## Test results:
##    t-statistic:  2.034
##    degrees of freedom:  23.025
##    p-value:  0.054
##
## Other information:
##    two-sided 95% confidence interval:  [-0.092, 11.048]
##    estimated effect size (Cohen's d):  0.724
```

Not too difficult, right? Not surprisingly, the output has exactly the same format as it did last time too:

The very first line is different, because it's telling you that its run a Welch test rather than a Student test, and of course all the numbers are a bit different. But I hope that the interpretation of this output should be fairly obvious. You read the output in the same way that you would for the Student test. You've got your descriptive statistics, the hypotheses, the test results and some other information. So that's all pretty easy.

Except, except… our result isn't significant anymore. When we ran the Student test, we did get a significant effect; but the Welch test on the same data set is not ($t(23.03)=2.03$, $p=.054$). What does this mean? Should we panic? Is the sky burning? Probably not. The fact that one test is significant and the other isn't doesn't itself mean very much, especially since I kind of rigged the data so that this would happen. As a general rule, it's not a good idea to go out of your way to try to interpret or explain the difference between a p-value of .049 and a p-value of .051. If this sort of thing happens in real life, the *difference* in these p-values is almost certainly due to chance. What does matter is that you take a little bit of care in thinking about what test you use. The Student test and the Welch test have different strengths and weaknesses. If the two populations really do have equal variances, then the Student test is slightly more powerful (lower Type II error rate) than the Welch test. However, if they *don't* have the same variances, then the assumptions of the Student test are violated and you may not be able to trust it: you might end up with a higher Type I error rate. So it's a trade off. However, in real life, I tend to prefer the Welch test; because almost no-one *actually* believes that the population variances are identical.

11.4.2 Assumptions of the test

The assumptions of the Welch test are very similar to those made by the Student t-test (see Section 13.3.8), except that the Welch test does not assume homogeneity of variance. This leaves only the assumption of normality, and the assumption of independence. The specifics of these assumptions are the same for the Welch test as for the Student test.

This page titled 11.4: The Independent Samples t-test (Welch Test) is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **13.4: The Independent Samples t-test (Welch Test)** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

11.5: The Paired-samples t-test

Regardless of whether we're talking about the Student test or the Welch test, an independent samples t-test is intended to be used in a situation where you have two samples that are, well, independent of one another. This situation arises naturally when participants are assigned randomly to one of two experimental conditions, but it provides a very poor approximation to other sorts of research designs. In particular, a repeated measures design – in which each participant is measured (with respect to the same outcome variable) in both experimental conditions – is not suited for analysis using independent samples t-tests. For example, we might be interested in whether listening to music reduces people's working memory capacity. To that end, we could measure each person's working memory capacity in two conditions: with music, and without music. In an experimental design such as this one,[194] each participant appears in *both* groups. This requires us to approach the problem in a different way; by using the ***paired samples t-test***.

11.5.1 data

The data set that we'll use this time comes from Dr Chico's class.[195] In her class, students take two major tests, one early in the semester and one later in the semester. To hear her tell it, she runs a very hard class, one that most students find very challenging; but she argues that by setting hard assessments, students are encouraged to work harder. Her theory is that the first test is a bit of a "wake up call" for students: when they realise how hard her class really is, they'll work harder for the second test and get a better mark. Is she right? To test this, let's have a look at the `chico.Rdata` file:

```
load( "./rbook-master/data/chico.Rdata" )
str(chico)
```

```
## 'data.frame':    20 obs. of  3 variables:
##  $ id         : Factor w/ 20 levels "student1","student10",..: 1 12 14 15 16 17 18
##  $ grade_test1: num  42.9 51.8 71.7 51.6 63.5 58 59.8 50.8 62.5 61.9 ...
##  $ grade_test2: num  44.6 54 72.3 53.4 63.8 59.3 60.8 51.6 64.3 63.2 ...
```

The data frame `chico` contains three variables: an `id` variable that identifies each student in the class, the `grade_test1` variable that records the student grade for the first test, and the `grade_test2` variable that has the grades for the second test. Here's the first six students:

```
head( chico )
```

```
##         id grade_test1 grade_test2
## 1 student1        42.9        44.6
## 2 student2        51.8        54.0
## 3 student3        71.7        72.3
## 4 student4        51.6        53.4
## 5 student5        63.5        63.8
## 6 student6        58.0        59.3
```

At a glance, it does seem like the class is a hard one (most grades are between 50% and 60%), but it does look like there's an improvement from the first test to the second one. If we take a quick look at the descriptive statistics

```
library( psych )
describe( chico )
```

```
##             vars  n  mean   sd median trimmed  mad  min  max range  skew
## id*            1 20 10.50 5.92   10.5   10.50 7.41  1.0 20.0  19.0  0.00
## grade_test1    2 20 56.98 6.62   57.7   56.92 7.71 42.9 71.7  28.8  0.05
## grade_test2    3 20 58.38 6.41   59.7   58.35 6.45 44.6 72.3  27.7 -0.05
##             kurtosis   se
## id*            -1.38 1.32
## grade_test1    -0.35 1.48
## grade_test2    -0.39 1.43
```

we see that this impression seems to be supported. Across all 20 students[196] the mean grade for the first test is 57%, but this rises to 58% for the second test. Although, given that the standard deviations are 6.6% and 6.4% respectively, it's starting to feel like maybe the improvement is just illusory; maybe just random variation. This impression is reinforced when you see the means and confidence intervals plotted in Figure 13.11. If we were to rely on this plot alone, we'd come to the same conclusion that we got from looking at the descriptive statistics that the `describe()` function produced. Looking at how wide those confidence intervals are, we'd be tempted to think that the apparent improvement in student performance is pure chance.

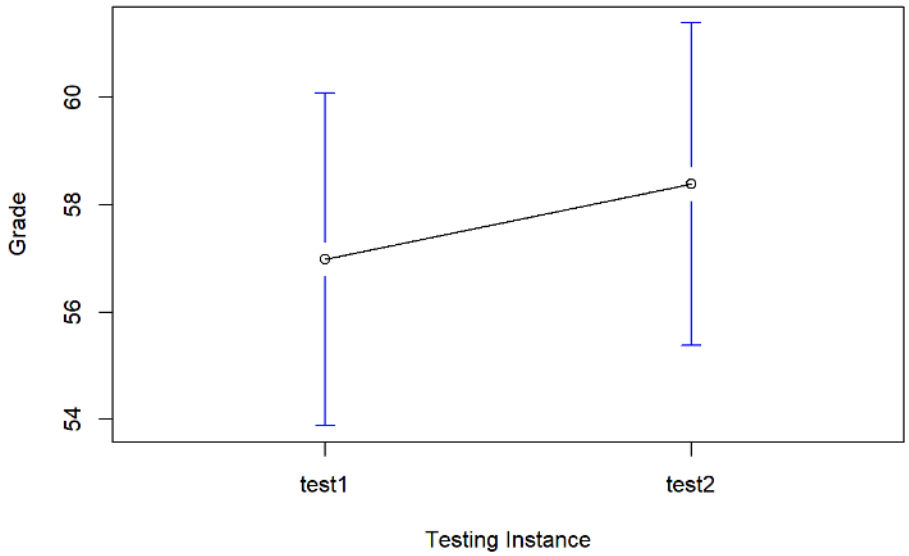


Figure 13.11: Mean grade for test 1 and test 2, with associated 95% confidence intervals

Nevertheless, this impression is wrong. To see why, take a look at the scatterplot of the grades for test 1 against the grades for test 2. shown in Figure 13.12.

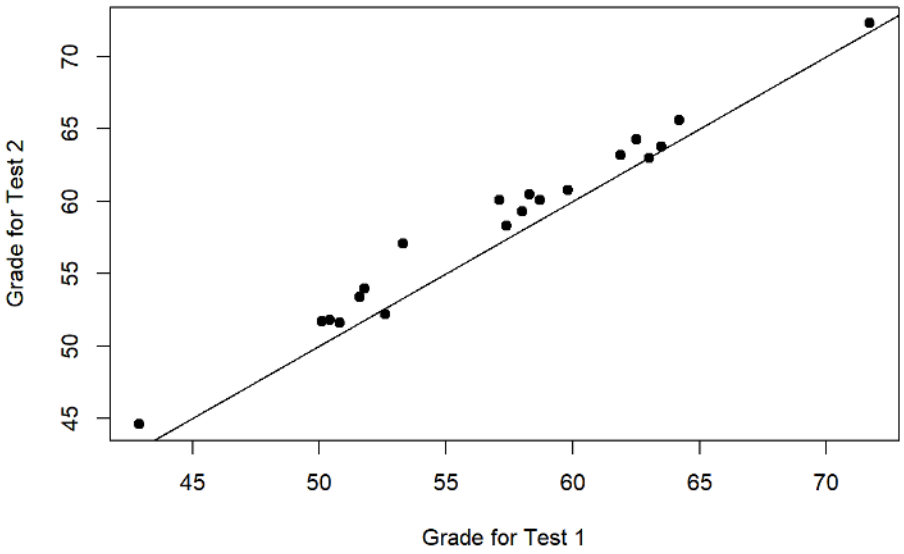


Figure 13.12: Scatterplot showing the individual grades for test 1 and test 2

In this plot, each dot corresponds to the two grades for a given student: if their grade for test 1 (x co-ordinate) equals their grade for test 2 (y co-ordinate), then the dot falls on the line. Points falling above the line are the students that performed better on the second test. Critically, almost all of the data points fall above the diagonal line: almost all of the students *do* seem to have improved their grade, if only by a small amount. This suggests that we should be looking at the *improvement* made by each student from one test to the next, and treating that as our raw data. To do this, we'll need to create a new variable for the `improvement` that each student makes, and add it to the `chico` data frame. The easiest way to do this is as follows:

```
chico$improvement <- chico$grade_test2 - chico$grade_test1
```

Notice that I assigned the output to a variable called `chico$improvement` . That has the effect of creating a new variable called `improvement` inside the `chico` data frame. So now when I look at the `chico` data frame, I get an output that looks like this:

```
head( chico )
```

```
##         id grade_test1 grade_test2 improvement
## 1 student1        42.9        44.6         1.7
## 2 student2        51.8        54.0         2.2
## 3 student3        71.7        72.3         0.6
## 4 student4        51.6        53.4         1.8
## 5 student5        63.5        63.8         0.3
## 6 student6        58.0        59.3         1.3
```

Now that we've created and stored this `improvement` variable, we can draw a histogram showing the distribution of these improvement scores (using the `hist()` function), shown in Figure 13.13.

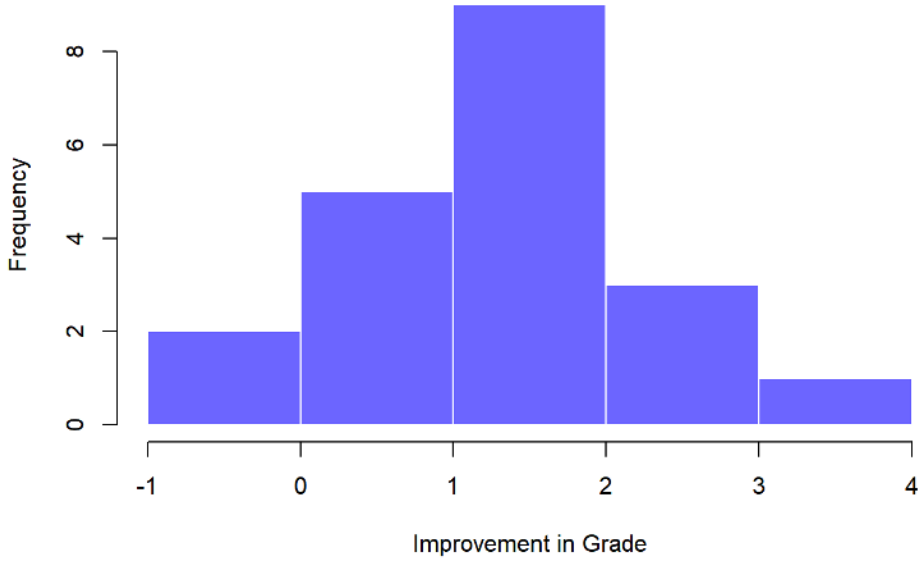


Figure 13.13: Histogram showing the improvement made by each student in Dr Chico's class. Notice that almost the entire distribution is above zero: the vast majority of students did improve their performance from the first test to the second one

When we look at histogram, it's very clear that there *is* a real improvement here. The vast majority of the students scored higher on the test 2 than on test 1, reflected in the fact that almost the entire histogram is above zero. In fact, if we use `ciMean()` to compute a confidence interval for the population mean of this new variable,

```
ciMean( x = chico$improvement )
```

```
##           2.5%    97.5%
## [1,] 0.9508686 1.859131
```

we see that it is 95% certain that the true (population-wide) average improvement would lie between 0.95% and 1.86%. So you can see, qualitatively, what's going on: there is a real "within student" improvement (everyone improves by about 1%), but it is very small when set against the quite large "between student" differences (student grades vary by about 20% or so).

11.5.2 What is the paired samples t-test?

In light of the previous exploration, let's think about how to construct an appropriate t test. One possibility would be to try to run an independent samples t-test using `grade_test1` and `grade_test2` as the variables of interest. However, this is clearly the wrong thing to do: the independent samples t-test assumes that there is no particular relationship between the two samples. Yet clearly that's not true in this case, because of the repeated measures structure to the data. To use the language that I introduced in the last section, if we were to try to do an independent samples t-test, we would be conflating the ***within subject*** differences (which is what we're interested in testing) with the ***between subject*** variability (which we are not).

The solution to the problem is obvious, I hope, since we already did all the hard work in the previous section. Instead of running an independent samples t-test on `grade_test1` and `grade_test2`, we run a *one-sample* t-test on the within-subject difference variable, `improvement`. To formalise this slightly, if X_{i1} is the score that the i-th participant obtained on the first variable, and X_{i2} is the score that the same person obtained on the second one, then the difference score is:

$$D_i = X_{i1} - X_{i2}$$

Notice that the difference scores is *variable 1 minus variable 2* and not the other way around, so if we want improvement to correspond to a positive valued difference, we actually want "test 2" to be our "variable 1". Equally, we would say that $\mu_D = \mu_1 - \mu_2$ is the population mean for this difference variable. So, to convert this to a hypothesis test, our null hypothesis is that this mean difference is zero; the alternative hypothesis is that it is not:

$$H_0: \mu_D = 0$$

$$H_1: \mu_D \neq 0$$

(this is assuming we're talking about a two-sided test here). This is more or less identical to the way we described the hypotheses for the one-sample t-test: the only difference is that the specific value that the null hypothesis predicts is 0. And so our t-statistic is defined in more or less the same way too. If we let $\bar{D}$ denote the mean of the difference scores, then

$$t = \frac{\bar{D}}{\mathrm{SE}(\bar{D})}$$

which is

$$t = \frac{\bar{D}}{\hat{\sigma}_D / \sqrt{N}}$$

where $\hat{\sigma}_D$ is the standard deviation of the difference scores. Since this is just an ordinary, one-sample t-test, with nothing special about it, the degrees of freedom are still N−1. And that's it: the paired samples t-test really isn't a new test at all: it's a one-sample t-test, but applied to the difference between two variables. It's actually very simple; the only reason it merits a discussion as long as the one we've just gone through is that you need to be able to recognise *when* a paired samples test is appropriate, and to understand *why* it's better than an independent samples t test.

11.5.3 Doing the test in R, part 1

How do you do a paired samples t-test in R. One possibility is to follow the process I outlined above: create a "difference" variable and then run a one sample t-test on that. Since we've already created a variable called `chico$improvement`, let's do that:

```
oneSampleTTest( chico$improvement, mu=0 )
```

```
##
##    One sample t-test
##
## Data variable:   chico$improvement
##
## Descriptive statistics:
##             improvement
##    mean           1.405
##    std dev.       0.970
##
## Hypotheses:
##    null:        population mean equals 0
##    alternative: population mean not equal to 0
##
## Test results:
##    t-statistic:  6.475
##    degrees of freedom:  19
##    p-value:  <.001
##
## Other information:
##    two-sided 95% confidence interval:  [0.951, 1.859]
##    estimated effect size (Cohen's d):  1.448
```

The output here is (obviously) formatted exactly the same was as it was the last time we used the `oneSampleTTest()` function (Section 13.2), and it confirms our intuition. There's an average improvement of 1.4% from test 1 to test 2, and this is significantly different from 0 (t(19)=6.48, p<.001).

However, suppose you're lazy and you don't want to go to all the effort of creating a new variable. Or perhaps you just want to keep the difference between one-sample and paired-samples tests clear in your head. If so, you can use the `pairedSamplesTTest()` function, also in the `lsr` package. Let's assume that your data organised like they are in the `chico` data frame, where there are two separate variables, one for each measurement. The way to run the test is to input a *one-sided* formula, just like you did when running a test of association using the `associationTest()` function in Chapter 12. For the `chico` data frame, the formula that you need would be `~ grade_time2 + grade_time1`. As usual, you'll also need to input the name of the data frame too. So the command just looks like this:

```
pairedSamplesTTest(
     formula = ~ grade_test2 + grade_test1, # one-sided formula listing the two varia
     data = chico                           # data frame containing the two variables
  )
```

```
##
##    Paired samples t-test
##
## Variables:  grade_test2 , grade_test1
##
## Descriptive statistics:
##             grade_test2 grade_test1 difference
##    mean          58.385      56.980      1.405
##    std dev.       6.406       6.616      0.970
##
## Hypotheses:
##    null:        population means equal for both measurements
##    alternative: different population means for each measurement
##
## Test results:
##    t-statistic:  6.475
##    degrees of freedom:  19
##    p-value:  <.001
##
## Other information:
##    two-sided 95% confidence interval:  [0.951, 1.859]
##    estimated effect size (Cohen's d):  1.448
```

The numbers are identical to those that come from the one sample test, which of course they have to be given that the paired samples t-test is just a one sample test under the hood. However, the output is a bit more detailed:

This time around the descriptive statistics block shows you the means and standard deviations for the original variables, as well as for the difference variable (notice that it always defines the difference as the first listed variable mines the second listed one). The null hypothesis and the alternative hypothesis are now framed in terms of the original variables rather than the difference score, but you should keep in mind that in a paired samples test it's still the difference score being tested. The statistical information at the bottom about the test result is of course the same as before.

11.5.4 Doing the test in R, part 2

The paired samples t-test is a little different from the other t-tests, because it is used in repeated measures designs. For the `chico` data, every student is "measured" twice, once for the first test, and again for the second test. Back in Section 7.7 I talked about the fact that repeated measures data can be expressed in two standard ways, known as *wide form* and *long form*. The `chico` data frame is in wide form: every row corresponds to a unique *person*. I've shown you the data in that form first because

that's the form that you're most used to seeing, and it's also the format that you're most likely to receive data in. However, the majority of tools in R for dealing with repeated measures data expect to receive data in long form. The paired samples t-test is a bit of an exception that way.

As you make the transition from a novice user to an advanced one, you're going to have to get comfortable with long form data, and switching between the two forms. To that end, I want to show you how to apply the `pairedSamplesTTest()` function to long form data. First, let's use the `wideToLong()` function to create a long form version of the `chico` data frame. If you've forgotten how the `wideToLong()` function works, it might be worth your while quickly re-reading Section 7.7. Assuming that you've done so, or that you're already comfortable with data reshaping, I'll use it to create a new data frame called `chico2`:

```
chico2 <- wideToLong( chico, within="time" )
head( chico2 )
```

```
##         id improvement  time grade
## 1 student1         1.7 test1  42.9
## 2 student2         2.2 test1  51.8
## 3 student3         0.6 test1  71.7
## 4 student4         1.8 test1  51.6
## 5 student5         0.3 test1  63.5
## 6 student6         1.3 test1  58.0
```

As you can see, this has created a new data frame containing three variables: an `id` variable indicating which person provided the data, a `time` variable indicating which test the data refers to (i.e., test 1 or test 2), and a `grade` variable that records what score the person got on that test. Notice that this data frame is in long form: every row corresponds to a unique *measurement*. Because every person provides two observations (test 1 and test 2), there are two rows for every person. To see this a little more clearly, I'll use the `sortFrame()` function to sort the rows of `chico2` by `id` variable (see Section 7.6.3).

```
chico2 <- sortFrame( chico2, id )
head( chico2 )
```

```
##           id improvement  time grade
## 1   student1         1.7 test1  42.9
## 21  student1         1.7 test2  44.6
## 10 student10         1.3 test1  61.9
## 30 student10         1.3 test2  63.2
## 11 student11         1.4 test1  50.4
## 31 student11         1.4 test2  51.8
```

As you can see, there are two rows for "student1": one showing their grade on the first test, the other showing their grade on the second test.[197]

Okay, suppose that we were given the `chico2` data frame to analyse. How would we run our paired samples t-test now? One possibility would be to use the `longToWide()` function (Section 7.7) to force the data back into wide form, and do the same thing that we did previously. But that's sort of defeating the point, and besides, there's an easier way. Let's think about what how the `chico2` data frame is structured: there are three variables here, and they all matter. The outcome measure is stored as the `grade`, and we effectively have two "groups" of measurements (test 1 and test 2) that are defined by the `time` points at which a test is given. Finally, because we want to keep track of which measurements should be paired together, we need to know which student obtained each grade, which is what the `id` variable gives us. So, when your data are presented to you in long form, we would want specify a *two-sided* formula and a data frame, in the same way that we do for an independent samples t-test:

the formula specifies the outcome variable and the groups, so in this case it would be `grade ~ time`, and the data frame is `chico2`. However, we also need to tell it the id variable, which in this case is boringly called `id`. So our command is:

```
pairedSamplesTTest(
     formula = grade ~ time,  # two sided formula: outcome ~ group
     data = chico2,           # data frame
     id = "id"                # name of the id variable
  )
```

```
##
##    Paired samples t-test
##
## Outcome variable:   grade
## Grouping variable:  time
## ID variable:        id
##
## Descriptive statistics:
##              test1  test2 difference
##    mean     56.980 58.385     -1.405
##    std dev.  6.616  6.406      0.970
##
## Hypotheses:
##    null:        population means equal for both measurements
##    alternative: different population means for each measurement
##
## Test results:
##    t-statistic:  -6.475
##    degrees of freedom:  19
##    p-value:  <.001
##
## Other information:
##    two-sided 95% confidence interval:  [-1.859, -0.951]
##    estimated effect size (Cohen's d):  1.448
```

Note that the name of the id variable is `"id"` and not `id`. Note that the `id` variable must be a factor. As of the current writing, you do need to include the quote marks, because the `pairedSamplesTTest()` function is expecting a *character string* that specifies the name of a variable. If I ever find the time I'll try to relax this constraint.

As you can see, it's a bit more detailed than the output from `oneSampleTTest()`. It gives you the descriptive statistics for the original variables, states the null hypothesis in a fashion that is a bit more appropriate for a repeated measures design, and then reports all the nuts and bolts from the hypothesis test itself. Not surprisingly the numbers the same as the ones that we saw last time.

One final comment about the `pairedSamplesTTest()` function. One of the reasons I designed it to be able handle long form and wide form data is that I want you to be get comfortable thinking about repeated measures data in both formats, and also to become familiar with the different ways in which R functions tend to specify models and tests for repeated measures data. With that last point in mind, I want to highlight a slightly different way of thinking about what the paired samples t-test is doing. There's a sense in which what you're really trying to do is look at how the outcome variable (`grade`) is related to the grouping variable (`time`), after taking account of the fact that there are individual differences between people (`id`). So there's a sense in which `id` is actually a *second* predictor: you're trying to predict the `grade` on the basis of the `time` and the `id`. With that in mind, the `pairedSamplesTTest()` function lets you specify a formula like this one

```
grade ~ time + (id)
```

This formula tells R everything it needs to know: the variable on the left (`grade`) is the outcome variable, the bracketed term on the right (`id`) is the id variable, and the other term on the right is the grouping variable (`time`). If you specify your formula that way, then you only need to specify the `formula` and the `data` frame, and so you can get away with using a command as simple as this one:

```
pairedSamplesTTest(
     formula = grade ~ time + (id),
     data = chico2
  )
```

or you can drop the argument names and just do this:

```
> pairedSamplesTTest( grade ~ time + (id), chico2 )
```

These commands will produce the same output as the last one, I personally find this format a lot more elegant. That being said, the main reason for allowing you to write your formulas that way is that they're quite similar to the way that mixed models (fancy pants repeated measures analyses) are specified in the `lme4` package. This book doesn't talk about mixed models (yet!), but if you go on to learn more statistics you'll find them pretty hard to avoid, so I've tried to lay a little bit of the groundwork here.

This page titled 11.5: The Paired-samples t-test is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **13.5: The Paired-samples t-test** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

11.6: One Sided Tests

When introducing the theory of null hypothesis tests, I mentioned that there are some situations when it's appropriate to specify a *one-sided* test (see Section 11.4.3). So far, all of the t-tests have been two-sided tests. For instance, when we specified a one sample t-test for the grades in Dr Zeppo's class, the null hypothesis was that the true mean was 67.5%. The alternative hypothesis was that the true mean was greater than *or* less than 67.5%. Suppose we were only interested in finding out if the true mean is greater than 67.5%, and have no interest whatsoever in testing to find out if the true mean is lower than 67.5%. If so, our null hypothesis would be that the true mean is 67.5% or less, and the alternative hypothesis would be that the true mean is greater than 67.5%. The `oneSampleTTest()` function lets you do this, by specifying the `one.sided` argument. If you set `one.sided="greater"`, it means that you're testing to see if the true mean is larger than `mu`. If you set `one.sided="less"`, then you're testing to see if the true mean is smaller than `mu`. Here's how it would work for Dr Zeppo's class:

```
oneSampleTTest( x=grades, mu=67.5, one.sided="greater" )
```

```
##
##    One sample t-test
##
## Data variable:   grades
##
## Descriptive statistics:
##             grades
##    mean     72.300
##    std dev.  9.521
##
## Hypotheses:
##    null:        population mean less than or equal to 67.5
##    alternative: population mean greater than 67.5
##
## Test results:
##    t-statistic:  2.255
##    degrees of freedom:  19
##    p-value:  0.018
##
## Other information:
##    one-sided 95% confidence interval:  [68.619, Inf]
##    estimated effect size (Cohen's d):  0.504
```

Notice that there are a few changes from the output that we saw last time. Most important is the fact that the null and alternative hypotheses have changed, to reflect the different test. The second thing to note is that, although the t-statistic and degrees of freedom have not changed, the p-value has. This is because the one-sided test has a different rejection region from the two-sided test. If you've forgotten why this is and what it means, you may find it helpful to read back over Chapter 11, and Section 11.4.3 in particular. The third thing to note is that the confidence interval is different too: it now reports a "one-sided" confidence interval rather than a two-sided one. In a two-sided confidence interval, we're trying to find numbers a and b such that we're 95% confident that the true mean lies *between* a and b. In a one-sided confidence interval, we're trying to find a single number a such that we're 95% confident that the true mean is *greater than* a (or less than a if you set `one.sided="less"`).

So that's how to do a one-sided one sample t-test. However, all versions of the t-test can be one-sided. For an independent samples t test, you could have a one-sided test if you're only interestd in testing to see if group A has *higher* scores than group B, but have no interest in finding out if group B has higher scores than group A. Let's suppose that, for Dr Harpo's class, you wanted to see if

Anastasia's students had higher grades than Bernadette's. The `independentSamplesTTest()` function lets you do this, again by specifying the `one.sided` argument. However, this time around you need to specify the name of the group that you're expecting to have the higher score. In our case, we'd write `one.sided = "Anastasia"`. So the command would be:

```
independentSamplesTTest(
    formula = grade ~ tutor,
    data = harpo,
    one.sided = "Anastasia"
  )
```

```
##
##    Welch's independent samples t-test
##
## Outcome variable:   grade
## Grouping variable:  tutor
##
## Descriptive statistics:
##             Anastasia Bernadette
##    mean        74.533     69.056
##    std dev.     8.999      5.775
##
## Hypotheses:
##    null:        population means are equal, or smaller for group 'Anastasia'
##    alternative: population mean is larger for group 'Anastasia'
##
## Test results:
##    t-statistic:  2.034
##    degrees of freedom:  23.025
##    p-value:  0.027
##
## Other information:
##    one-sided 95% confidence interval:  [0.863, Inf]
##    estimated effect size (Cohen's d):  0.724
```

Again, the output changes in a predictable way. The definition of the null and alternative hypotheses has changed, the p-value has changed, and it now reports a one-sided confidence interval rather than a two-sided one.

What about the paired samples t-test? Suppose we wanted to test the hypothesis that grades go *up* from test 1 to test 2 in Dr Zeppo's class, and are not prepared to consider the idea that the grades go down. Again, we can use the `one.sided` argument to specify the one-sided test, and it works the same way it does for the independent samples t-test. You need to specify the name of the group whose scores are expected to be larger under the alternative hypothesis. If your data are in wide form, as they are in the `chico` data frame, you'd use this command:

```
pairedSamplesTTest(
     formula = ~ grade_test2 + grade_test1,
     data = chico,
     one.sided = "grade_test2"
  )
```

```
##
##    Paired samples t-test
##
## Variables:  grade_test2 , grade_test1
##
## Descriptive statistics:
##             grade_test2 grade_test1 difference
##    mean          58.385      56.980      1.405
##    std dev.       6.406       6.616      0.970
##
## Hypotheses:
##    null:        population means are equal, or smaller for measurement 'grade_test
##    alternative: population mean is larger for measurement 'grade_test2'
##
## Test results:
##    t-statistic:  6.475
##    degrees of freedom:  19
##    p-value:  <.001
##
## Other information:
##    one-sided 95% confidence interval:  [1.03, Inf]
##    estimated effect size (Cohen's d):  1.448
```

Yet again, the output changes in a predictable way. The hypotheses have changed, the p-value has changed, and the confidence interval is now one-sided. If your data are in long form, as they are in the `chico2` data frame, it still works the same way. Either of the following commands would work,

```
> pairedSamplesTTest(
    formula = grade ~ time,
    data = chico2,
    id = "id",
    one.sided = "test2"
  )

> pairedSamplesTTest(
    formula = grade ~ time + (id),
    data = chico2,
    one.sided = "test2"
  )
```

and would produce the same answer as the output shown above.

This page titled 11.6: One Sided Tests is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **13.6: One Sided Tests** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

11.7: Using the t.test() Function

In this chapter, we've talked about three different kinds of t-test: the one sample test, the independent samples test (Student's and Welch's), and the paired samples test. In order to run these different tests, I've shown you three different functions: `oneSampleTTest()`, `independentSamplesTTest()` and `pairedSamplesTTest()`. I wrote these as three different functions for two reasons. Firstly, I thought it made sense to have separate functions for each test, in order to help make it clear to beginners that there *are* different tests. Secondly, I wanted to show you some functions that produced "verbose" output, to help you see what hypotheses are being tested and so on.

However, once you've started to become familiar with t-tests and with using R, you might find it easier to use the `t.test()` function. It's one function, but it can run all four of the different t-tests that we've talked about. Here's how it works. Firstly, suppose you want to run a one sample t-test. To run the test on the `grades` data from Dr Zeppo's class (Section 13.2), we'd use a command like this:

```
t.test( x = grades, mu = 67.5 )
```

```
##
##  One Sample t-test
##
## data:  grades
## t = 2.2547, df = 19, p-value = 0.03615
## alternative hypothesis: true mean is not equal to 67.5
## 95 percent confidence interval:
##  67.84422 76.75578
## sample estimates:
## mean of x
##      72.3
```

The input is the same as for the `oneSampleTTest()`: we specify the sample data using the argument `x`, and the value against which it is to be tested using the argument `mu`. The output is a lot more compressed.

As you can see, it still has all the information you need. It tells you what type of test it ran and the data it tested it on. It gives you the t-statistic, the degrees of freedom and the p-value. And so on. There's nothing wrong with this output, but in my experience it can be a little confusing when you're just starting to learn statistics, because it's a little disorganised. Once you know what you're looking at though, it's pretty easy to read off the relevant information.

What about independent samples t-tests? As it happens, the `t.test()` function can be used in much the same way as the `independentSamplesTTest()` function, by specifying a formula, a data frame, and using `var.equal` to indicate whether you want a Student test or a Welch test. If you want to run the Welch test from Section 13.4, then you'd use this command:

```
t.test( formula = grade ~ tutor, data = harpo )
```

```
##
##  Welch Two Sample t-test
##
## data:  grade by tutor
## t = 2.0342, df = 23.025, p-value = 0.05361
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.09249349 11.04804904
## sample estimates:
##  mean in group Anastasia mean in group Bernadette
##                 74.53333                 69.05556
```

If you want to do the Student test, it's exactly the same except that you need to add an additional argument indicating that `var.equal = TRUE`. This is no different to how it worked in the `independentSamplesTTest()` function.

Finally, we come to the paired samples t-test. Somewhat surprisingly, given that most R functions for dealing with repeated measures data require data to be in long form, the `t.test()` function isn't really set up to handle data in long form. Instead it expects to be given two separate variables, `x` and `y`, and you need to specify `paired=TRUE`. And on top of that, you'd better make sure that the first element of `x` and the first element of `y` actually correspond to the same person! Because it doesn't ask for an "id" variable. I don't know why. So, in order to run the paired samples t test on the data from Dr Chico's class, we'd use this command:

```
t.test( x = chico$grade_test2,   # variable 1 is the "test2" scores
        y = chico$grade_test1,   # variable 2 is the "test1" scores
        paired = TRUE            # paired test
 )
```

```
##
##  Paired t-test
##
## data:  chico$grade_test2 and chico$grade_test1
## t = 6.4754, df = 19, p-value = 3.321e-06
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.9508686 1.8591314
## sample estimates:
## mean of the differences
##                   1.405
```

Yet again, these are the same numbers that we saw in Section 13.5. Feel free to check.

This page titled 11.7: Using the t.test() Function is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **13.7: Using the t.test() Function** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

11.8: Effect Size

The most commonly used measure of effect size for a t-test is ***Cohen's d*** (Cohen 1988). It's a very simple measure in principle, with quite a few wrinkles when you start digging into the details. Cohen himself defined it primarily in the context of an independent samples t-test, specifically the Student test. In that context, a natural way of defining the effect size is to divide the difference between the means by an estimate of the standard deviation. In other words, we're looking to calculate *something* along the lines of this:

$$d = \frac{(\text{mean } 1) - (\text{mean } 2)}{\text{std dev}}$$

and he suggested a rough guide for interpreting d in Table **??**. You'd think that this would be pretty unambiguous, but it's not; largely because Cohen wasn't too specific on what he thought should be used as the measure of the standard deviation (in his defence, he was trying to make a broader point in his book, not nitpick about tiny details). As discussed by McGrath and Meyer (2006), there are several different version in common usage, and each author tends to adopt slightly different notation. For the sake of simplicity (as opposed to accuracy) I'll use d to refer to any statistic that you calculate from the sample, and use δ to refer to a theoretical population effect. Obviously, that does mean that there are several different things all called d. The `cohensD()` function in the `lsr` package uses the `method` argument to distinguish between them, so that's what I'll do in the text.

My suspicion is that the only time that you would want Cohen's d is when you're running a t-test, and if you're using the `oneSampleTTest`, `independentSamplesTTest` and `pairedSamplesTTest()` functions to run your t-tests, then you don't need to learn any new commands, because they automatically produce an estimate of Cohen's d as part of the output. However, if you're using `t.test()` then you'll need to use the `cohensD()` function (also in the `lsr` package) to do the calculations.

d-value	rough interpretation
about 0.2	small effect
about 0.5	moderate effect
about 0.8	large effect

11.8.1 Cohen's d from one sample

The simplest situation to consider is the one corresponding to a one-sample t-test. In this case, the one sample mean $\bar{X}$ and one (hypothesised) population mean μ_o to compare it to. Not only that, there's really only one sensible way to estimate the population standard deviation: we just use our usual estimate $\hat{\sigma}$. Therefore, we end up with the following as the only way to calculate d,

$$d = \frac{\bar{X} - \mu_0}{\hat{\sigma}}$$

When writing the `cohensD()` function, I've made some attempt to make it work in a similar way to `t.test()`. As a consequence, `cohensD()` can calculate your effect size regardless of which type of t-test you performed. If what you want is a measure of Cohen's d to accompany a one-sample t-test, there's only two arguments that you need to care about. These are:

- `x`. A numeric vector containing the sample data.
- `mu`. The mean against which the mean of `x` is compared (default value is `mu = 0`).

We don't need to specify what `method` to use, because there's only one version of d that makes sense in this context. So, in order to compute an effect size for the data from Dr Zeppo's class (Section 13.2), we'd type something like this:

```
cohensD( x = grades,    # data are stored in the grades vector
         mu = 67.5      # compare students to a mean of 67.5
 )
```

```
## [1] 0.5041691
```

and, just so that you can see that there's nothing fancy going on, the command below shows you how to calculate it if there weren't no fancypants `cohensD()` function available:

```
( mean(grades) - 67.5 ) / sd(grades)
```

```
## [1] 0.5041691
```

Yep, same number. Overall, then, the psychology students in Dr Zeppo's class are achieving grades (mean = 72.3%) that are about .5 standard deviations higher than the level that you'd expect (67.5%) if they were performing at the same level as other students. Judged against Cohen's rough guide, this is a moderate effect size.

11.8.2 Cohen's d from a Student t test

The majority of discussions of Cohen's d focus on a situation that is analogous to Student's independent samples t test, and it's in this context that the story becomes messier, since there are several different versions of d that you might want to use in this situation, and you can use the `method` argument to the `cohensD()` function to pick the one you want. To understand why there are multiple versions of d, it helps to take the time to write down a formula that corresponds to the true population effect size δ. It's pretty straightforward,

$$\delta = \frac{\mu_1 - \mu_2}{\sigma}$$

where, as usual, μ1 and μ2 are the population means corresponding to group 1 and group 2 respectively, and σ is the standard deviation (the same for both populations). The obvious way to estimate δ is to do exactly the same thing that we did in the t-test itself: use the sample means as the top line, and a pooled standard deviation estimate for the bottom line:

$$d = \frac{\bar{X}_1 - \bar{X}_2}{\hat{\sigma}_p}$$

where $\hat{\sigma}_p$ is the exact same pooled standard deviation measure that appears in the t-test. This is the most commonly used version of Cohen's d when applied to the outcome of a Student t-test ,and is sometimes referred to as Hedges' g statistic (Hedges 1981). It corresponds to `method = "pooled"` in the `cohensD()` function, and it's the default.

However, there are other possibilities, which I'll briefly describe. Firstly, you may have reason to want to use only one of the two groups as the basis for calculating the standard deviation. This approach (often called Glass' Δ) only makes most sense when you have good reason to treat one of the two groups as a purer reflection of "natural variation" than the other. This can happen if, for instance, one of the two groups is a control group. If that's what you want, then use `method = "x.sd"` or `method = "y.sd"` when using `cohensD()` . Secondly, recall that in the usual calculation of the pooled standard deviation we divide by N−2 to correct for the bias in the sample variance; in one version of Cohen's d this correction is omitted. Instead, we divide by N. This version (`method = "raw"`) makes sense primarily when you're trying to calculate the effect size in the sample; rather than estimating an effect size in the population. Finally, there is a version based on Hedges and Olkin (1985), who point out there is a small bias in the usual (pooled) estimation for Cohen's d. Thus they introduce a small correction (`method = "corrected"`), by multiplying the usual value of d by (N−3)/(N−2.25).

In any case, ignoring all those variations that you could make use of if you wanted, let's have a look at how to calculate the default version. In particular, suppose we look at the data from Dr Harpo's class (the `harpo` data frame). The command that we want to use is very similar to the relevant `t.test()` command, but also specifies a `method`

```
cohensD( formula = grade ~ tutor,  # outcome ~ group
         data = harpo,             # data frame
         method = "pooled"         # which version to calculate?
)
```

```
## [1] 0.7395614
```

This is the version of Cohen's d that gets reported by the `independentSamplesTTest()` function whenever it runs a Student t-test.

11.8.3 Cohen's d from a Welch test

Suppose the situation you're in is more like the Welch test: you still have two independent samples, but you no longer believe that the corresponding populations have equal variances. When this happens, we have to redefine what we mean by the population effect size. I'll refer to this new measure as δ', so as to keep it distinct from the measure δ which we defined previously. What Cohen (1988) suggests is that we could define our new population effect size by averaging the two population variances. What this means is that we get:

$$\delta' = \frac{\mu_1 - \mu_2}{\sigma'}$$

where

$$\sigma' = \sqrt{\frac{\sigma_1^2 + \sigma_2^2}{2}}$$

This seems quite reasonable, but notice that none of the measures that we've discussed so far are attempting to estimate this new quantity. It might just be my own ignorance of the topic, but I'm only aware of one version of Cohen's d that actually estimates the unequal-variance effect size δ' rather than the equal-variance effect size δ. All we do to calculate d for this version (`method = "unequal"`) is substitute the sample means $\bar{X}_1$ and $\bar{X}_2$ and the corrected sample standard deviations $\hat{\sigma}_1$ and $\hat{\sigma}_2$ into the equation for δ'. This gives us the following equation for d,

$$d = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{\hat{\sigma}_1^{\,2} + \hat{\sigma}_2^{\,2}}{2}}}$$

as our estimate of the effect size. There's nothing particularly difficult about calculating this version in R, since all we have to do is change the `method` argument:

```
cohensD( formula = grade ~ tutor,
         data = harpo,
         method = "unequal"
 )
```

```
## [1] 0.7244995
```

This is the version of Cohen's d that gets reported by the `independentSamplesTTest()` function whenever it runs a Welch t-test.

11.8.4 Cohen's d from a paired-samples test

Finally, what should we do for a paired samples t-test? In this case, the answer depends on what it is you're trying to do. *If* you want to measure your effect sizes relative to the distribution of difference scores, the measure of d that you calculate is just (`method = "paired"`)

$$d = \frac{\bar{D}}{\hat{\sigma}_D}$$

where $\hat{\sigma_D}$ is the estimate of the standard deviation of the differences. The calculation here is pretty straightforward

```
cohensD( x = chico$grade_test2,
         y = chico$grade_test1,
         method = "paired"
 )
```

```
## [1] 1.447952
```

This is the version of Cohen's d that gets reported by the `pairedSamplesTTest()` function. The only wrinkle is figuring out whether this is the measure you want or not. To the extent that you care about the practical consequences of your research, you often want to measure the effect size relative to the *original* variables, not the *difference* scores (e.g., the 1% improvement in Dr Chico's class is pretty small when measured against the amount of between-student variation in grades), in which case you use the same versions of Cohen's d that you would use for a Student or Welch test. For instance, when we do that for Dr Chico's class,

```
cohensD( x = chico$grade_test2,
         y = chico$grade_test1,
         method = "pooled"
 )
```

```
## [1] 0.2157646
```

what we see is that the overall effect size is quite small, when assessed on the scale of the original variables.

This page titled 11.8: Effect Size is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **13.8: Effect Size** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

11.9: Checking the Normality of a Sample

All of the tests that we have discussed so far in this chapter have assumed that the data are normally distributed. This assumption is often quite reasonable, because the central limit theorem (Section 10.3.3) does tend to ensure that many real world quantities are normally distributed: any time that you suspect that your variable is *actually* an average of lots of different things, there's a pretty good chance that it will be normally distributed; or at least close enough to normal that you can get away with using t-tests. However, life doesn't come with guarantees; and besides, there are lots of ways in which you can end up with variables that are highly non-normal. For example, any time you think that your variable is actually the minimum of lots of different things, there's a very good chance it will end up quite skewed. In psychology, response time (RT) data is a good example of this. If you suppose that there are lots of things that could trigger a response from a human participant, then the actual response will occur the first time one of these trigger events occurs.[198] This means that RT data are systematically non-normal. Okay, so if normality is assumed by all the tests, and is mostly but not always satisfied (at least approximately) by real world data, how can we check the normality of a sample? In this section I discuss two methods: QQ plots, and the Shapiro-Wilk test.

11.9.1 plots

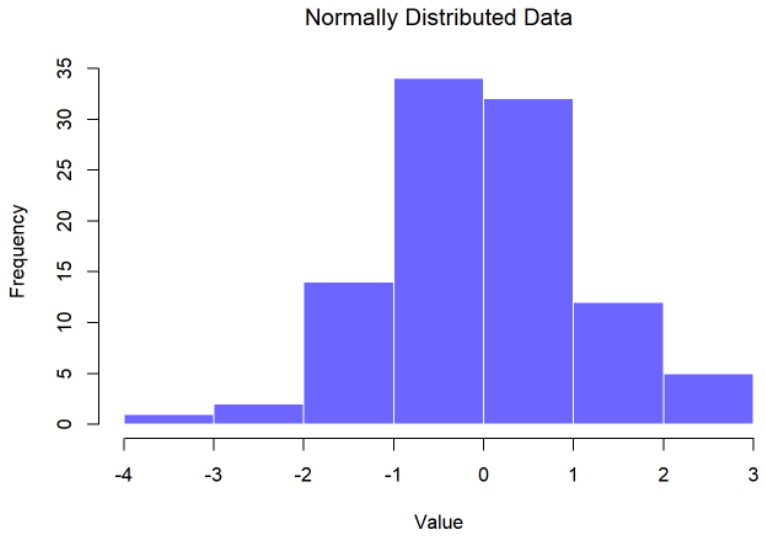


Figure 13.14: Histogram of `normal.data`, a normally distributed sample with 100 observations.

```
## Normally Distributed Data
## skew= -0.02936155
## kurtosis= -0.06035938
##
##  Shapiro-Wilk normality test
##
## data:  data
## W = 0.99108, p-value = 0.7515
```

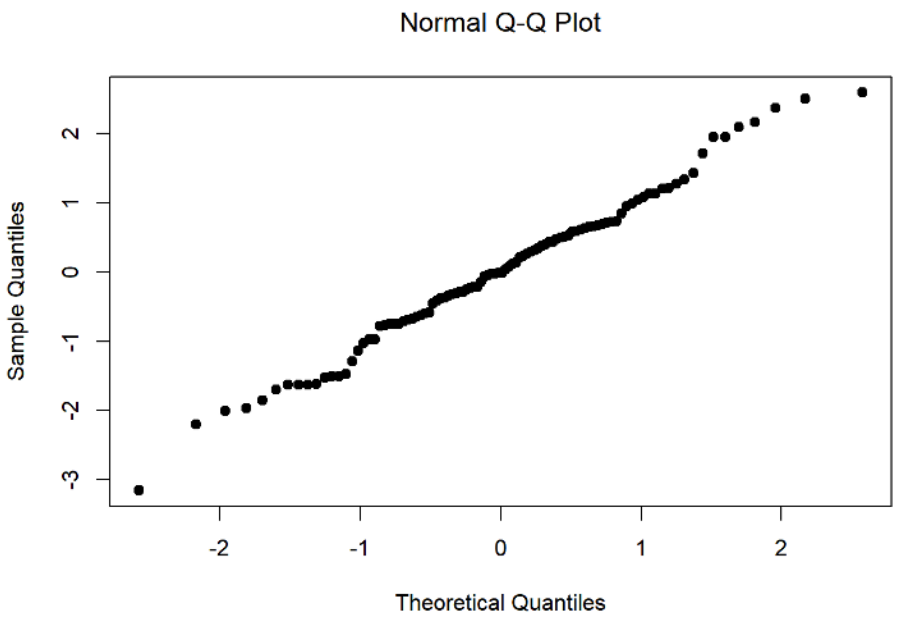


Figure 13.15: Normal QQ plot of `normal.data` , a normally distributed sample with 100 observations.

The Shapiro-Wilk statistic associated with the data in Figures 13.14 and 13.15 is W=.99, indicating that no significant departures from normality were detected (p=.73). As you can see, these data form a pretty straight line; which is no surprise given that we sampled them from a normal distribution! In contrast, have a look at the two data sets shown in Figures 13.16, 13.17, 13.18, 13.19. Figures 13.16 and 13.17 show the histogram and a QQ plot for a data set that is highly skewed: the QQ plot curves upwards. Figures 13.18 and 13.19 show the same plots for a heavy tailed (i.e., high kurtosis) data set: in this case, the QQ plot flattens in the middle and curves sharply at either end.

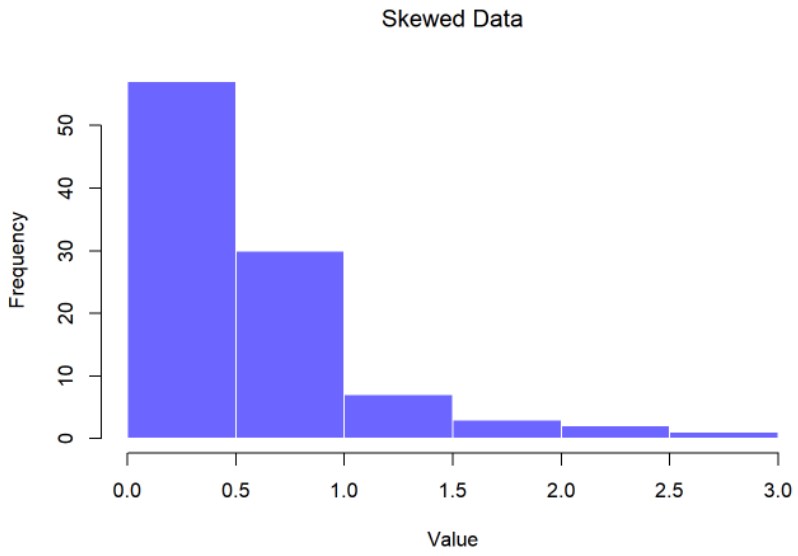


Figure 13.16: A histogram of the 100 observations in a `skewed.data` set

```
## Skewed Data
## skew= 1.889475
## kurtosis= 4.4396
##
##  Shapiro-Wilk normality test
##
## data:  data
## W = 0.81758, p-value = 8.908e-10
```

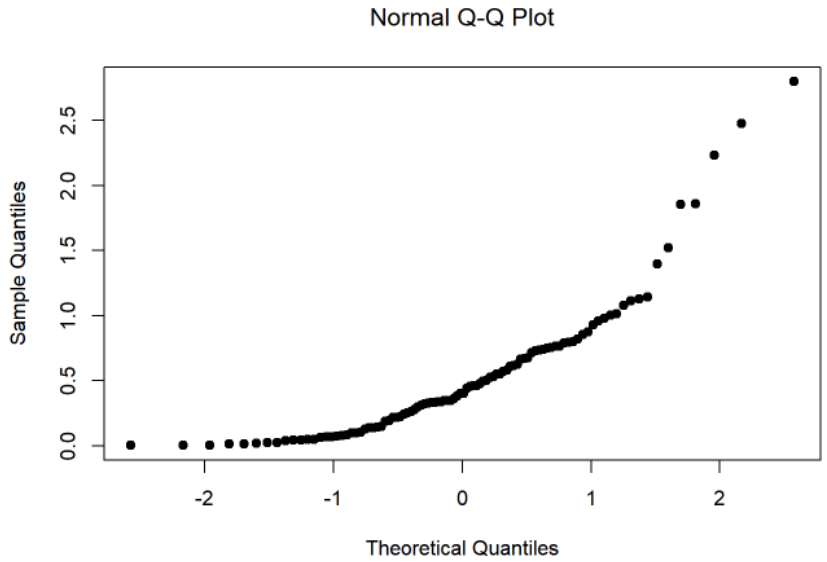


Figure 13.17: A normal QQ plot of the 100 observations in a `skewed.data` set

The skewness of the data in Figures 13.16 and 13.17 is 1.94, and is reflected in a QQ plot that curves upwards. As a consequence, the Shapiro-Wilk statistic is W=.80, reflecting a significant departure from normality (p<.001).

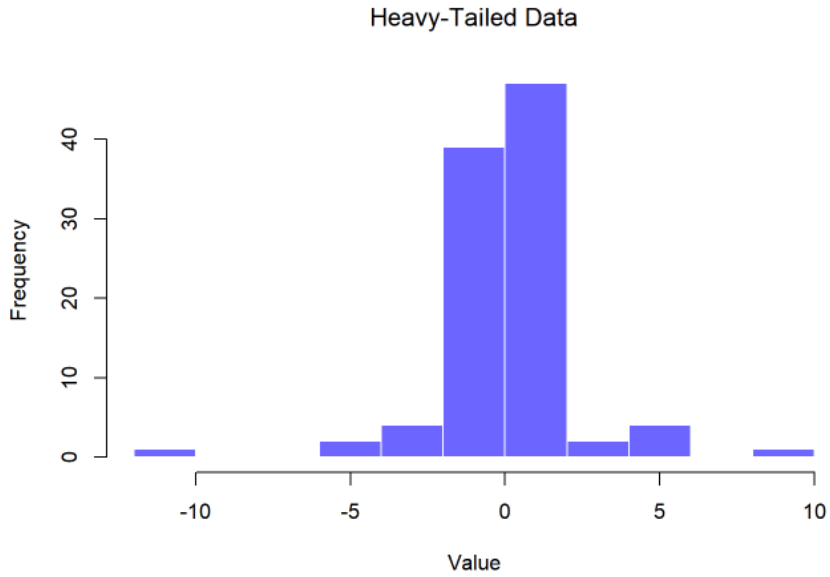


Figure 13.18: A histogram of the 100 observations in a *heavy tailed*` data set, again consisting of 100 observations.

```
## Heavy-Tailed Data
## skew= -0.05308273
## kurtosis= 7.508765
##
##  Shapiro-Wilk normality test
##
## data:  data
## W = 0.83892, p-value = 4.718e-09
```

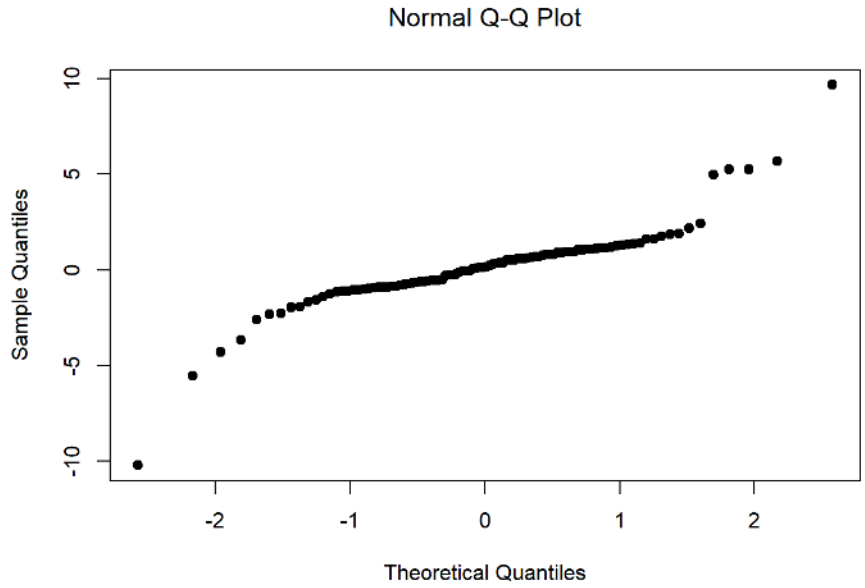
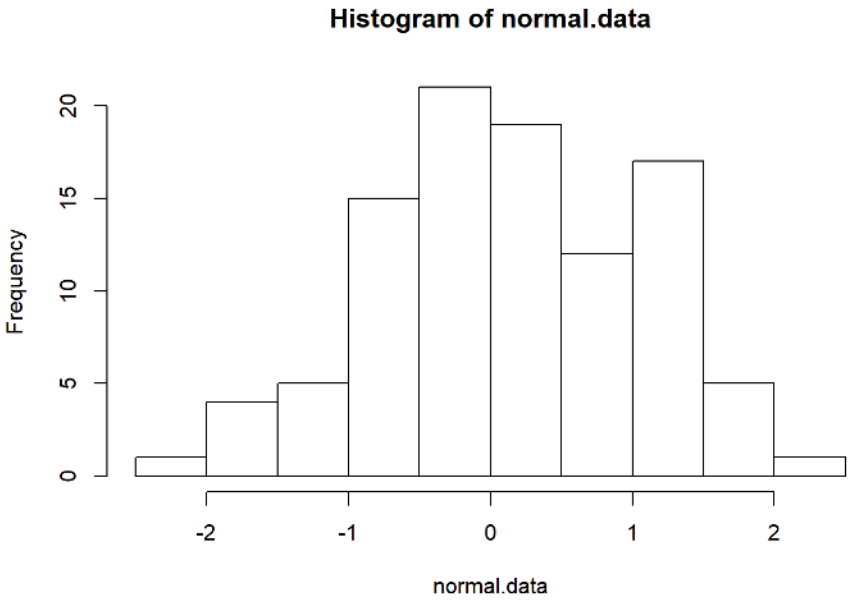


Figure 13.19: A histogram of the 100 observations in a *heavy tailed*` data set, again consisting of 100 observations.

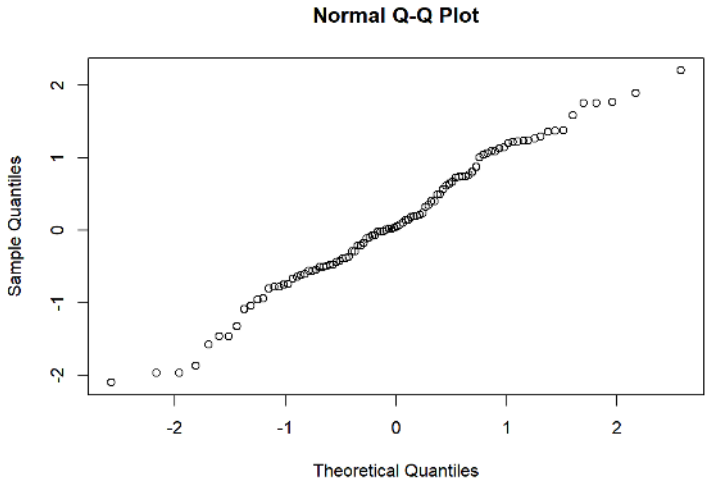
Figures 13.18 and 13.19 shows the same plots for a heavy tailed data set, again consisting of 100 observations. In this case, the heavy tails in the data produce a high kurtosis (2.80), and cause the QQ plot to flatten in the middle, and curve away sharply on either side. The resulting Shapiro-Wilk statistic is W=.93, again reflecting significant non-normality (p<.001).

One way to check whether a sample violates the normality assumption is to draw a ***"quantile-quantile" plot*** (QQ plot). This allows you to visually check whether you're seeing any systematic violations. In a QQ plot, each observation is plotted as a single dot. The x co-ordinate is the theoretical quantile that the observation should fall in, if the data were normally distributed (with mean and variance estimated from the sample) and on the y co-ordinate is the actual quantile of the data within the sample. If the data are normal, the dots should form a straight line. For instance, lets see what happens if we generate data by sampling from a normal distribution, and then drawing a QQ plot using the R function `qqnorm()` . The `qqnorm()` function has a few arguments, but the only one we really need to care about here is `y` , a vector specifying the data whose normality we're interested in checking. Here's the R commands:

```
normal.data <- rnorm( n = 100 )  # generate N = 100 normally distributed numbers
hist( x = normal.data )          # draw a histogram of these numbers
```




```
qqnorm( y = normal.data )        # draw the QQ plot
```



11.9.2 Shapiro-Wilk tests

Although QQ plots provide a nice way to informally check the normality of your data, sometimes you'll want to do something a bit more formal. And when that moment comes, the ***Shapiro-Wilk test*** (Shapiro and Wilk 1965) is probably what you're looking for.[199] As you'd expect, the null hypothesis being tested is that a set of N observations is normally distributed. The test statistic that it calculates is conventionally denoted as W, and it's calculated as follows. First, we sort the observations in order of increasing size, and let X1 be the smallest value in the sample, X2 be the second smallest and so on. Then the value of W is given by

$$W = \frac{\left(\sum_{i=1}^{N} a_i X_i\right)^2}{\sum_{i=1}^{N} \left(X_i - \bar{X}\right)^2}$$

where $\bar{X}$ is the mean of the observations, and the ai values are … mumble, mumble … something complicated that is a bit beyond the scope of an introductory text.

Because it's a little hard to explain the maths behind the W statistic, a better idea is to give a broad brush description of how it behaves. Unlike most of the test statistics that we'll encounter in this book, it's actually *small* values of W that indicated departure from normality. The W statistic has a maximum value of 1, which arises when the data look "perfectly normal". The smaller the value of W, the less normal the data are. However, the sampling distribution for W – which is not one of the standard ones that I

discussed in Chapter 9 and is in fact a complete pain in the arse to work with – does depend on the sample size N. To give you a feel for what these sampling distributions look like, I've plotted three of them in Figure 13.20. Notice that, as the sample size starts to get large, the sampling distribution becomes very tightly clumped up near W=1, and as a consequence, for larger samples W doesn't have to be very much smaller than 1 in order for the test to be significant.

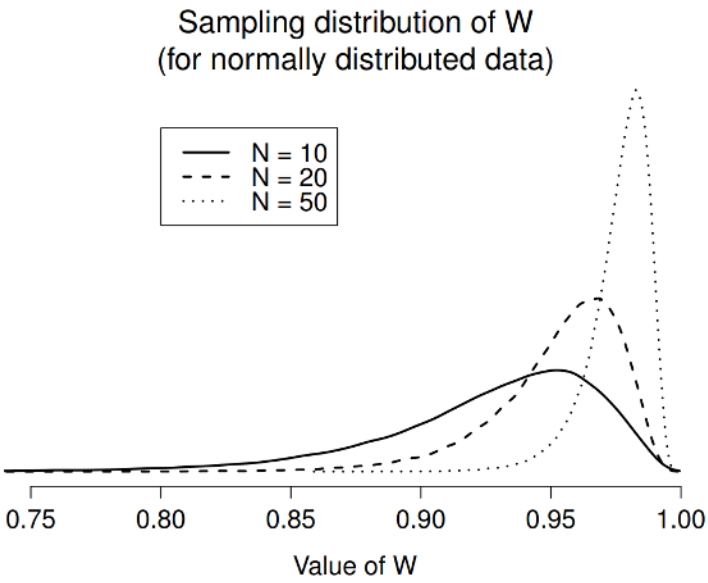


Figure 13.20: Sampling distribution of the Shapiro-Wilk W statistic, under the null hypothesis that the data are normally distributed, for samples of size 10, 20 and 50. Note that *small* values of W indicate departure from normality.

To run the test in R, we use the `shapiro.test()` function. It has only a single argument `x`, which is a numeric vector containing the data whose normality needs to be tested. For example, when we apply this function to our `normal.data`, we get the following:

```
shapiro.test( x = normal.data )
```

```
##
##  Shapiro-Wilk normality test
##
## data:  normal.data
## W = 0.98654, p-value = 0.4076
```

So, not surprisingly, we have no evidence that these data depart from normality. When reporting the results for a Shapiro-Wilk test, you should (as usual) make sure to include the test statistic W and the p value, though given that the sampling distribution depends so heavily on N it would probably be a politeness to include N as well.

This page titled 11.9: Checking the Normality of a Sample is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **13.9: Checking the Normality of a Sample** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

11.10: Testing Non-normal Data with Wilcoxon Tests

Okay, suppose your data turn out to be pretty substantially non-normal, but you still want to run something like a t-test? This situation occurs a lot in real life: for the AFL winning margins data, for instance, the Shapiro-Wilk test made it very clear that the normality assumption is violated. This is the situation where you want to use Wilcoxon tests.

Like the t-test, the Wilcoxon test comes in two forms, one-sample and two-sample, and they're used in more or less the exact same situations as the corresponding t-tests. Unlike the t-test, the Wilcoxon test doesn't assume normality, which is nice. In fact, they don't make any assumptions about what kind of distribution is involved: in statistical jargon, this makes them ***nonparametric tests***. While avoiding the normality assumption is nice, there's a drawback: the Wilcoxon test is usually less powerful than the t-test (i.e., higher Type II error rate). I won't discuss the Wilcoxon tests in as much detail as the t-tests, but I'll give you a brief overview.

11.10.1 sample Wilcoxon test

I'll start by describing the ***two sample Wilcoxon test*** (also known as the Mann-Whitney test), since it's actually simpler than the one sample version. Suppose we're looking at the scores of 10 people on some test. Since my imagination has now failed me completely, let's pretend it's a "test of awesomeness", and there are two groups of people, "A" and "B". I'm curious to know which group is more awesome. The data are included in the file `awesome.Rdata`, and like many of the data sets I've been using, it contains only a single data frame, in this case called `awesome`. Here's the data:

```
load("./rbook-master/data/awesome.Rdata")
print( awesome )
```

```
##    scores group
## 1     6.4     A
## 2    10.7     A
## 3    11.9     A
## 4     7.3     A
## 5    10.0     A
## 6    14.5     B
## 7    10.4     B
## 8    12.9     B
## 9    11.7     B
## 10   13.0     B
```

As long as there are no ties (i.e., people with the exact same awesomeness score), then the test that we want to do is surprisingly simple. All we have to do is construct a table that compares every observation in group A against every observation in group B. Whenever the group A datum is larger, we place a check mark in the table:

		group B				
		14.5	10.4	12.4	11.7	13.0
group A	6.4	.	.	.	.	.
	10.7	.	✓	.	.	.
	11.9	.	✓	.	✓	.
	7.3	.	.	.	.	.
	10.0	.	.	.	.	.

We then count up the number of checkmarks. This is our test statistic, W.[200] The actual sampling distribution for W is somewhat complicated, and I'll skip the details. For our purposes, it's sufficient to note that the interpretation of W is qualitatively the same as the interpretation of t or z. That is, if we want a two-sided test, then we reject the null hypothesis when W is very large or very small; but if we have a directional (i.e., one-sided) hypothesis, then we only use one or the other.

The structure of the `wilcox.test()` function should feel very familiar to you by now. When you have your data organised in terms of an outcome variable and a grouping variable, then you use the `formula` and `data` arguments, so your command looks like this:

```
wilcox.test( formula = scores ~ group, data = awesome)
```

```
##
##  Wilcoxon rank sum test
##
## data:  scores by group
## W = 3, p-value = 0.05556
## alternative hypothesis: true location shift is not equal to 0
```

Just like we saw with the `t.test()` function, there is an `alternative` argument that you can use to switch between two-sided tests and one-sided tests, plus a few other arguments that we don't need to worry too much about at an introductory level. Similarly, the `wilcox.test()` function allows you to use the `x` and `y` arguments when you have your data stored separately for each group. For instance, suppose we use the data from the `awesome2.Rdata` file:

```
load( "./rbook-master/data/awesome2.Rdata" )
score.A
```

```
## [1]  6.4 10.7 11.9  7.3 10.0
```

```
score.B
```

```
## [1] 14.5 10.4 12.9 11.7 13.0
```

When your data are organised like this, then you would use a command like this:

```
wilcox.test( x = score.A, y = score.B )
```

```
##
##  Wilcoxon rank sum test
##
## data:  score.A and score.B
## W = 3, p-value = 0.05556
## alternative hypothesis: true location shift is not equal to 0
```

The output that R produces is pretty much the same as last time.

11.10.2 sample Wilcoxon test

What about the ***one sample Wilcoxon test*** (or equivalently, the paired samples Wilcoxon test)? Suppose I'm interested in finding out whether taking a statistics class has any effect on the happiness of students. Here's my data:

```
load( "./rbook-master/data/happy.Rdata" )
print( happiness )
```

```
##    before after change
## 1      30     6    -24
## 2      43    29    -14
## 3      21    11    -10
## 4      24    31      7
## 5      23    17     -6
## 6      40     2    -38
## 7      29    31      2
## 8      56    21    -35
## 9      38     8    -30
## 10     16    21      5
```

What I've measured here is the happiness of each student `before` taking the class and `after` taking the class; the `change` score is the difference between the two. Just like we saw with the t-test, there's no fundamental difference between doing a paired-samples test using `before` and `after`, versus doing a one-sample test using the `change` scores. As before, the simplest way to think about the test is to construct a tabulation. The way to do it this time is to take those change scores that are positive valued, and tabulate them against all the complete sample. What you end up with is a table that looks like this:

		all differences									
		-24	-14	-10	7	-6	-38	2	-35	-30	5
positive differences	7	.	.	.	✓	✓	.	✓	.	.	✓
	2	.	.	.	.	.	.	✓	.	.	.
	5	.	.	.	.	.	.	✓	.	.	✓

Counting up the tick marks this time, we get a test statistic of V=7. As before, if our test is two sided, then we reject the null hypothesis when V is very large or very small. As far of running it in R goes, it's pretty much what you'd expect. For the one-sample version, the command you would use is

```
wilcox.test( x = happiness$change,
             mu = 0
)
```

```
##
##  Wilcoxon signed rank test
##
## data:  happiness$change
## V = 7, p-value = 0.03711
## alternative hypothesis: true location is not equal to 0
```

As this shows, we have a significant effect. Evidently, taking a statistics class does have an effect on your happiness. Switching to a paired samples version of the test won't give us different answers, of course; but here's the command to do it:

```
wilcox.test( x = happiness$after,
             y = happiness$before,
             paired = TRUE
)
```

```
##
##  Wilcoxon signed rank test
##
## data:  happiness$after and happiness$before
## V = 7, p-value = 0.03711
## alternative hypothesis: true location shift is not equal to 0
```

This page titled 11.10: Testing Non-normal Data with Wilcoxon Tests is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **13.10: Testing Non-normal Data with Wilcoxon Tests** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

11.11: Summary

- A one sample t-test is used to compare a single sample mean against a hypothesised value for the population mean. (Section 13.2)
- An independent samples t-test is used to compare the means of two groups, and tests the null hypothesis that they have the same mean. It comes in two forms: the Student test (Section 13.3 assumes that the groups have the same standard deviation, the Welch test (Section 13.4) does not.
- A paired samples t-test is used when you have two scores from each person, and you want to test the null hypothesis that the two scores have the same mean. It is equivalent to taking the difference between the two scores for each person, and then running a one sample t-test on the difference scores. (Section 13.5)
- Effect size calculations for the difference between means can be calculated via the Cohen's d statistic. (Section 13.8).
- You can check the normality of a sample using QQ plots and the Shapiro-Wilk test. (Section 13.9)
- If your data are non-normal, you can use Wilcoxon tests instead of t-tests. (Section 13.10)

References

Student, A. 1908. "The Probable Error of a Mean." *Biometrika* 6: 1–2.

Box, J. F. 1987. "Guinness, Gosset, Fisher, and Small Samples." *Statistical Science* 2: 45–52.

Welch, B. L. 1947. "The Generalization of 'Student's' Problem When Several Different Population Variances Are Involved." *Biometrika* 34: 28–35.

Cohen, J. 1988. *Statistical Power Analysis for the Behavioral Sciences*. 2nd ed. Lawrence Erlbaum.

McGrath, R. E., and G. J. Meyer. 2006. "When Effect Sizes Disagree: The Case of r and d." *Psychological Methods* 11: 386–401.

Hedges, L. V. 1981. "Distribution Theory for Glass's Estimator of Effect Size and Related Estimators." *Journal of Educational Statistics* 6: 107–28.

Hedges, L. V., and I. Olkin. 1985. *Statistical Methods for Meta-Analysis*. New York: Academic Press.

Shapiro, S. S., and M. B. Wilk. 1965. "An Analysis of Variance Test for Normality (Complete Samples)." *Biometrika* 52: 591–611.

184. We won't cover multiple predictors until Chapter 15
185. Informal experimentation in my garden suggests that yes, it does. Australian natives are adapted to low phosphorus levels relative to everywhere else on Earth, apparently, so if you've bought a house with a bunch of exotics and you want to plant natives, don't follow my example: keep them separate. Nutrients to European plants are poison to Australian ones. There's probably a joke in that, but I can't figure out what it is.
186. Actually this is too strong. Strictly speaking the z test only requires that the sampling distribution of the mean be normally distributed; if the population is normal then it necessarily follows that the sampling distribution of the mean is also normal. However, as we saw when talking about the central limit theorem, it's quite possible (even commonplace) for the sampling distribution to be normal even if the population distribution itself is non-normal. However, in light of the sheer ridiculousness of the assumption that the true standard deviation is known, there really isn't much point in going into details on this front!
187. Well, sort of. As I understand the history, Gosset only provided a partial solution: the general solution to the problem was provided by Sir Ronald Fisher.
188. More seriously, I tend to think the reverse is true: I get very suspicious of technical reports that fill their results sections with nothing except the numbers. It might just be that I'm an arrogant jerk, but I often feel like an author that makes no attempt to explain and interpret their analysis to the reader either doesn't understand it themselves, or is being a bit lazy. Your readers are smart, but not infinitely patient. Don't annoy them if you can help it.
189. Although it is the simplest, which is why I started with it.
190. A funny question almost always pops up at this point: what the heck *is* the population being referred to in this case? Is it the set of students actually taking Dr Harpo's class (all 33 of them)? The set of people who might take the class (an unknown number) of them? Or something else? Does it matter which of these we pick? It's traditional in an introductory behavioural stats class to mumble a lot at this point, but since I get asked this question every year by my students, I'll give a brief answer. Technically

yes, it does matter: if you change your definition of what the "real world" population actually is, then the sampling distribution of your observed mean $\bar{X}$ changes too. The t-test relies on an assumption that the observations are sampled at random from an infinitely large population; and to the extent that real life isn't like that, then the t-test can be wrong. In practice, however, this isn't usually a big deal: even though the assumption is almost always wrong, it doesn't lead to a lot of pathological behaviour from the test, so we tend to just ignore it.

191. Yes, I have a "favourite" way of thinking about pooled standard deviation estimates. So what?
192. A more correct notation will be introduced in Chapter 14.
193. Well, I guess you can average apples and oranges, and what you end up with is a delicious fruit smoothie. But no one really thinks that a fruit smoothie is a very good way to describe the original fruits, do they?
194. This design is very similar to the one in Section 12.8 that motivated the McNemar test. This should be no surprise. Both are standard repeated measures designs involving two measurements. The only difference is that this time our outcome variable is interval scale (working memory capacity) rather than a binary, nominal scale variable (a yes-or-no question).
195. At this point we have Drs Harpo, Chico and Zeppo. No prizes for guessing who Dr Groucho is.
196. This is obviously a class being taught at a very small or very expensive university, or else is a postgraduate class. *I've* never taught an intro stats class with less than 350 students.
197. The `sortFrame()` function sorts factor variables like `id` in alphabetical order, which is why it jumps from "student1" to "student10"
198. This is a massive oversimplification.
199. Either that, or the Kolmogorov-Smirnov test, which is probably more traditional than the Shapiro-Wilk, though most things I've read seem to suggest Shapiro-Wilk is the better test of normality; although Kolomogorov-Smirnov is a general purpose test of distributional equivalence, so it can be adapted to handle other kinds of distribution tests; in R it's implemented via the `ks.test()` function.
200. Actually, there are two different versions of the test statistic; they differ from each other by a constant value. The version that I've described is the one that R calculates.

This page titled 11.11: Summary is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **13.11: Summary** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

11.12: Statistical Literacy

Learning Objectives

- Effectiveness of Surgery for Weight Loss

Research on the effectiveness of surgery for weight loss reported here found that "The surgery was associated with significantly greater weight loss [than the control group who dieted] through 2 years (61.3 versus 11.2 pounds, $P < 0.001$)."

Example 11.12.1: what do you think

What test could have been used and how would it have been computed?

Solution

For each subject a difference score between their initial weight and final weight could be computed. A t test of whether the mean difference score differs significantly from 0 could then be computed. The mean difference score will equal the difference between the mean weight losses of the two groups ($61.3 - 11.2 = 50.1$).

This page titled 11.12: Statistical Literacy is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **12.11: Statistical Literacy** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

11.E: Tests of Means (Exercises)

General Questions

Q1

The scores of a random sample of 8 students on a physics test are as follows: $60, 62, 67, 69, 70, 72, 75, 78$.

a. Test to see if the sample mean is significantly different from 65 at the 0.05 level. Report the t and p values.

b. The researcher realizes that she accidentally recorded the score that should have been 76 as 67. Are these corrected scores significantly different from 65 at the 0.05 level? (relevant section)

Q2

A (hypothetical) experiment is conducted on the effect of alcohol on perceptual motor ability. Ten subjects are each tested twice, once after having two drinks and once after having two glasses of water. The two tests were on two different days to give the alcohol a chance to wear off. Half of the subjects were given alcohol first and half were given water first. The scores of the 10 subjects are shown below. The first number for each subject is their performance in the "water" condition. Higher scores reflect better performance. Test to see if alcohol had a significant effect. Report the t and p values. (relevant section)

water	alcohol
16	13
15	13
11	10
20	18
19	17
14	11
13	10
15	15
14	11
16	16

Q3

The scores on a (hypothetical) vocabulary test of a group of 20 year olds and a group of 60 year olds are shown below.

20 yr olds	60 yr olds
27	26
26	29
21	29
24	29
15	27
18	16
17	20
12	27
13	

a. Test the mean difference for significance using the 0.05 level. (relevant section).

b. List the assumptions made in computing your answer.(relevant section)

Q4

The sampling distribution of a statistic is normally distributed with an estimated standard error of 12, ($df = 20$).

a. What is the probability that you would have gotten a mean of 107 (or more extreme) if the population parameter were 100? Is this probability significant at the 0.05 level (two-tailed)?

b. What is the probability that you would have gotten a mean of 95 or less (one-tailed)? Is this probability significant at the 0.05 level? You may want to use the t Distribution calculator for this problem. (relevant section)

Q5

How do you decide whether to use an independent groups t test or a correlated t test (test of dependent means)? (relevant section & relevant section)

Q6

An experiment compared the ability of three groups of subjects to remember briefly-presented chess positions. The data are shown below.

Non-players	Beginners	Tournament players
22.1	32.5	40.1
22.3	37.1	45.6
26.2	39.1	51.2
29.6	40.5	56.4
31.7	45.5	58.1
33.5	51.3	71.1
38.9	52.6	74.9
39.7	55.7	75.9
43.2	55.9	80.3
43.2	57.7	85.3

a. Using the Tukey HSD procedure, determine which groups are significantly different from each other at the 0.05 level. (relevant section)
b. Now compare each pair of groups using t-tests. Make sure to control for the familywise error rate (at 0.05) by using the **Bonferroni correction**. Specify the alpha level you used.

Q7

Below are data showing the results of six subjects on a memory test. The three scores per subject are their scores on three trials (a, b, and c) of a memory task. Are the subjects getting better each trial? Test the linear effect of trial for the data.

a	b	c
4	6	7
3	7	8
2	8	5
1	4	7
4	6	9
2	4	2

a. Compute L for each subject using the contrast weights -1, 0, and 1. That is, compute $(-1)(a) + (0)(b) + (1)(c)$ for each subject.
b. Compute a one-sample t-test on this column (with the L values for each subject) you created. (relevant section)

Q8

Participants threw darts at a target. In one condition, they used their preferred hand; in the other condition, they used their other hand. All subjects performed in both conditions (the order of conditions was counterbalanced). Their scores are shown below.

Preferred	Non-preferred
12	7
7	9
11	8
13	10
10	9

a. Which kind of t-test should be used?
b. Calculate the two-tailed t and p values using this t test.
c. Calculate the one-tailed t and p values using this t test.

Q9

Assume the data in the previous problem were collected using two different groups of subjects: One group used their preferred hand and the other group used their non-preferred hand. Analyze the data and compare the results to those for the previous problem (relevant section)

Q10

You have 4 means, and you want to compare each mean to every other mean.

a. How many tests total are you going to compute?
b. What would be the chance of making at least one **Type I** error if the **Type I** error for each test was 0.05 and the tests were independent? (relevant section & relevant section)
c. Are the tests independent and how does independence/non-independence affect the probability in (b).

Q11

In an experiment, participants were divided into 4 groups. There were 20 participants in each group, so the degrees of freedom (error) for this study was $80 - 4 = 76$. Tukey's HSD test was performed on the data.

a. Calculate the p value for each pair based on the Q value given below. You will want to use the Studentized Range Calculator.
b. Which differences are significant at the 0.05 level? (relevant section)

Comparison of Groups	Q
A - B	3.4
A - C	3.8
A - D	4.3
B - C	1.7
B - D	3.9
C - D	3.7

Q12

If you have 5 groups in your study, why shouldn't you just compute a t test of each group mean with each other group mean? (relevant section)

Q13

You are conducting a study to see if students do better when they study all at once or in intervals. One group of 12 participants took a test after studying for one hour continuously. The other group of 12 participants took a test after studying for three twenty minute sessions. The first group had a mean score of 75 and a variance of 120. The second group had a mean score of 86 and a variance of 100.

a. What is the calculated t value? Are the mean test scores of these two groups significantly different at the 0.05 level?
b. What would the t value be if there were only 6 participants in each group? Would the scores be significant at the 0.05 level?

Q14

A new test was designed to have a mean of 80 and a standard deviation of 10. A random sample of 20 students at your school take the test, and the mean score turns out to be 85. Does this score differ significantly from 80? To answer this problem, you may want to use the Normal Distribution Calculator.(relevant section)

Q15

You perform a one-sample t test and calculate a t statistic of 3.0. The mean of your sample was 1.3 and the standard deviation was 2.6. How many participants were used in this study? (relevant section)

Q16

True/false: The contrasts $(-3, 111)$ and $(0, 0, -1, 1)$ are orthogonal. (relevant section)

Q17

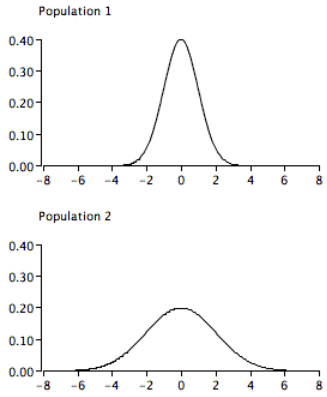
True/false: If you are making 4 comparisons between means, then based on the **Bonferroni correction**, you should use an alpha level of 0.01 for each test. (relevant section)

Q18

True/false: Correlated t tests almost always have greater power than independent t tests. (relevant section)

Q19

True/false:The graph below represents a violation of the homogeneity of variance assumption. (relevant section)



Q20

True/false: When you are conducting a one-sample t test and you know the population standard deviation, you look up the critical t value in the table based on the degrees of freedom. (relevant section)

Questions from Case Studies

The following questions use data from the Angry Moods (AM) case study.

Q21

(AM#17) Do athletes or non-athletes calm down more when angry? Conduct a t test to see if the difference between groups in Control-In scores is statistically significant.

Q22

Do people in general have a higher Anger-Out or Anger-In score? Conduct a t test on the difference between means of these two scores. Are these two means independent or dependent? (relevant section)

The following questions use data from the Smiles and Leniency (SL) case study.

Q23

Compare each mean to the neutral mean. Be sure to control for the familywise error rate. (relevant section)

Q24

Does a "felt smile" lead to more leniency than other types of smiles?

a. Calculate L (the linear combination) using the following contrast weights $false: -1,\ felt: 2,\ miserable: -1,\ neutral: 0$.
b. Perform a significance test on this value of L. (relevant section)

The following questions are from the Animal Research (AR) case study.

Q25

(AR#8) Conduct an independent samples t test comparing males to females on the belief that animal research is necessary. (relevant section)

Q26

(AR#9) Based on the t test you conducted in the previous problem, are you able to reject the null hypothesis if $alpha = 0.05$? What about if $alpha = 0.1$? (relevant section)

Q27

(AR#10) Is there any evidence that the t test assumption of homogeneity of variance is violated in the t test you computed in #25? (relevant section)

The following questions use data from the ADHD Treatment (AT) case study.

Q28

Compare each dosage with the dosage below it (compare $d0$ and $d15$, $d15$ and $d30$, and $d30$ and $d60$). Remember that the patients completed the task after every dosage.

a. If the familywise error rate is 0.05, what is the alpha level you will use for each comparison when doing the Bonferroni correction?
b. Which differences are significant at this level? (relevant section)

Q29

Does performance increase linearly with dosage?

a. Plot a line graph of this data.
b. Compute L for each patient. To do this, create a new variable where you multiply the following coefficients by their corresponding dosages and then sum up the total: $(-3)d0 + (-1)d15 + (1)d30 + (3)d60$ (see #8). What is the mean of L?
c. Perform a significance test on L. Compute the 95% confidence interval for L. (relevant section)

Select Answers

S1

a. $t(7) = 1.91$

S4

b. 0.035

S7

b. two-tailed $p = 0.0088$

S8

b. $p = 0.1662$

S11

a. $A - B : p = 0.085$

S13

a. $t(22) = 2.57$

S23

$t(76) = 3.04$

S25

a. $p = 0.0745$

S29

c. $p = 0.0006$

This page titled 11.E: Tests of Means (Exercises) is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **12.E: Tests of Means (Exercises)** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

CHAPTER OVERVIEW

12: Comparing Several Means (One-way ANOVA)

This chapter introduces one of the most widely used tools in statistics, known as "the analysis of variance", which is usually referred to as ANOVA. The basic technique was developed by Sir Ronald Fisher in the early 20th century, and it is to him that we owe the rather unfortunate terminology. The term ANOVA is a little misleading, in two respects. Firstly, although the name of the technique refers to variances, ANOVA is concerned with investigating differences in means. Secondly, there are several different things out there that are all referred to as ANOVAs, some of which have only a very tenuous connection to one another. Later on in the book we'll encounter a range of different ANOVA methods that apply in quite different situations, but for the purposes of this chapter we'll only consider the simplest form of ANOVA, in which we have several different groups of observations, and we're interested in finding out whether those groups differ in terms of some outcome variable of interest. This is the question that is addressed by a ***one-way ANOVA***.

The structure of this chapter is as follows: In Section 14.1 I'll introduce a fictitious data set that we'll use as a running example throughout the chapter. After introducing the data, I'll describe the mechanics of how a one-way ANOVA actually works (Section 14.2) and then focus on how you can run one in R (Section 14.3). These two sections are the core of the chapter. The remainder of the chapter discusses a range of important topics that inevitably arise when running an ANOVA, namely how to calculate effect sizes (Section 14.4), post hoc tests and corrections for multiple comparisons (Section 14.5) and the assumptions that ANOVA relies upon (Section 14.6). We'll also talk about how to check those assumptions and some of the things you can do if the assumptions are violated (Sections 14.7 to 14.10). At the end of the chapter we'll talk a little about the relationship between ANOVA and other statistical tools (Section 14.11).

12.1: Summary
12.2: An Illustrative Data Set
12.3: How ANOVA Works
12.4: Running an ANOVA in R
12.5: Effect Size
12.6: Multiple Comparisons and Post Hoc Tests
12.7: Assumptions of One-way ANOVA
12.8: Checking the Homogeneity of Variance Assumption
12.9: Removing the Homogeneity of Variance Assumption
12.10: Checking the Normality Assumption
12.11: Removing the Normality Assumption
12.12: On the Relationship Between ANOVA and the Student t Test

This page titled 12: Comparing Several Means (One-way ANOVA) is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

12.1: Summary

There's a fair bit covered in this chapter, but there's still a lot missing. Most obviously, I haven't yet discussed any analog of the paired samples t-test for more than two groups. There is a way of doing this, known as *repeated measures ANOVA*, which will appear in a later version of this book. I also haven't discussed how to run an ANOVA when you are interested in more than one grouping variable, but that will be discussed in a lot of detail in Chapter 16. In terms of what we have discussed, the key topics were:

- The basic logic behind how ANOVA works (Section 14.2) and how to run one in R (Section 14.3).
- How to compute an effect size for an ANOVA (Section 14.4)
- Post hoc analysis and corrections for multiple testing (Section 14.5).
- The assumptions made by ANOVA (Section 14.6).
- How to check the homogeneity of variance assumption (Section 14.7) and what to do if it is violated (Section 14.8).
- How to check the normality assumption (Section 14.9 and what to do if it is violated (Section 14.10).

As with all of the chapters in this book, there are quite a few different sources that I've relied upon, but the one stand-out text that I've been most heavily influenced by is Sahai and Ageel (2000). It's not a good book for beginners, but it's an excellent book for more advanced readers who are interested in understanding the mathematics behind ANOVA.

References

Hays, W. L. 1994. *Statistics*. 5th ed. Fort Worth, TX: Harcourt Brace.

Shaffer, J. P. 1995. "Multiple Hypothesis Testing." *Annual Review of Psychology* 46: 561–84.

Hsu, J. C. 1996. *Multiple Comparisons: Theory and Methods*. London, UK: Chapman; Hall.

Dunn, O.J. 1961. "Multiple Comparisons Among Means." *Journal of the American Statistical Association* 56: 52–64.

Holm, S. 1979. "A Simple Sequentially Rejective Multiple Test Procedure." *Scandinavian Journal of Statistics* 6: 65–70.

Levene, H. 1960. "Robust Tests for Equality of Variances." In *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling*, edited by I. Olkin et al, 278–92. Palo Alto, CA: Stanford University Press.

Brown, M. B., and A. B. Forsythe. 1974. "Robust Tests for Equality of Variances." *Journal of the American Statistical Association* 69: 364–67.

Welch, B. 1951. "On the Comparison of Several Mean Values: An Alternative Approach." *Biometrika* 38: 330–36.

Kruskal, W. H., and W. A. Wallis. 1952. "Use of Ranks in One-Criterion Variance Analysis." *Journal of the American Statistical Association* 47: 583–621.

Sahai, H., and M. I. Ageel. 2000. *The Analysis of Variance: Fixed, Random and Mixed Models*. Boston: Birkhauser.

201. When all groups have the same number of observations, the experimental design is said to be "balanced". Balance isn't such a big deal for one-way ANOVA, which is the topic of this chapter. It becomes more important when you start doing more complicated ANOVAs.
202. In a later versions I'm intending to expand on this. But because I'm writing in a rush, and am already over my deadlines, I'll just briefly note that if you read ahead to Chapter 16 and look at how the "treatment effect" at level k of a factor is defined in terms of the αk values (see Section 16.2), it turns out that Q refers to a weighted mean of the squared treatment effects, $Q = \left(\sum_{k=1}^{G} N_k \alpha_k^2\right) / (G-1)$
203. we want to be sticklers for accuracy, $1 + \frac{2}{df_2 - 2}$
204. o be precise, party like "it's 1899 and we've got no friends and nothing better to do with our time than do some calculations that wouldn't have made any sense in 1899 because ANOVA didn't exist until about the 1920s".
205. Actually, it *also* provides a function called `anova()`, but that works a bit differently, so let's just ignore it for now.
206. It's worth noting that you can get the same result by using the command `anova( my.anova )`.
207. A potentially important footnote – I wrote the `etaSquared()` function for the `lsr` package as a teaching exercise, but like all the other functions in the `lsr` package it hasn't been exhaustively tested. As of this writing – `lsr` package version

0.5 – there is at least one known bug in the code. In some cases at least, it doesn't work (and can give very silly answers) when you set the `weights` on the observations to something other than uniform. That doesn't matter at all for this book, since those kinds of analyses are well beyond the scope, but I haven't had a chance to revisit the package in a long time; it would probably be wise to be very cautious with the use of this function in any context other than very simple introductory analyses. Thanks to Emil Kirkegaard for finding the bug! (Oh, and while I'm here, there's an interesting blog post by Daniel Lakens suggesting that eta-squared itself is perhaps not the best measure of effect size in real world data analysis: http://daniellakens.blogspot.com.au/2015/06/why-you-should-use-omega-squared.html

208. I should point out that there are other functions in R for running multiple comparisons, and at least one of them works this way: the `TukeyHSD()` function takes an `aov` object as its input, and outputs Tukey's "honestly significant difference" tests. I talk about Tukey's HSD in Chapter 16.
209. If you *do* have some theoretical basis for wanting to investigate some comparisons but not others, it's a different story. In those circumstances you're not really running "post hoc" analyses at all: you're making "planned comparisons". I do talk about this situation later in the book (Section 16.9), but for now I want to keep things simple.
210. It's worth noting in passing that not all adjustment methods try to do this. What I've described here is an approach for controlling "family wise Type I error rate". However, there are other post hoc tests seek to control the "false discovery rate", which is a somewhat different thing.
211. There's also a function called `p.adjust()` in which you can input a vector of raw p-values, and it will output a vector of adjusted p-values. This can be handy sometimes. I should also note that more advanced users may wish to consider using some of the tools provided by the `multcomp` package.
212. Note that neither of these figures has been tidied up at all: if you want to create nicer looking graphs it's always a good idea to use the tools from Chapter 6 to help you draw cleaner looking images.
213. A technical term.

This page titled 12.1: Summary is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **14.1: Summary** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

12.2: An Illustrative Data Set

Suppose you've become involved in a clinical trial in which you are testing a new antidepressant drug called *Joyzepam*. In order to construct a fair test of the drug's effectiveness, the study involves three separate drugs to be administered. One is a placebo, and the other is an existing antidepressant / anti-anxiety drug called *Anxifree*. A collection of 18 participants with moderate to severe depression are recruited for your initial testing. Because the drugs are sometimes administered in conjunction with psychological therapy, your study includes 9 people undergoing cognitive behavioural therapy (CBT) and 9 who are not. Participants are randomly assigned (doubly blinded, of course) a treatment, such that there are 3 CBT people and 3 no-therapy people assigned to each of the 3 drugs. A psychologist assesses the mood of each person after a 3 month run with each drug: and the overall *improvement* in each person's mood is assessed on a scale ranging from −5 to +5.

With that as the study design, let's now look at what we've got in the data file:

```
load( "./rbook-master/data/clinicaltrial.Rdata" ) # load data
str(clin.trial)
```

```
## 'data.frame':    18 obs. of  3 variables:
##  $ drug     : Factor w/ 3 levels "placebo","anxifree",..: 1 1 1 2 2 2 3 3 3 1 ...
##  $ therapy  : Factor w/ 2 levels "no.therapy","CBT": 1 1 1 1 1 1 1 1 2 ...
##  $ mood.gain: num  0.5 0.3 0.1 0.6 0.4 0.2 1.4 1.7 1.3 0.6 ...
```

So we have a single data frame called `clin.trial`, containing three variables; `drug`, `therapy` and `mood.gain`. Next, let's print the data frame to get a sense of what the data actually look like.

```
print( clin.trial )
```

```
##        drug    therapy mood.gain
## 1   placebo no.therapy       0.5
## 2   placebo no.therapy       0.3
## 3   placebo no.therapy       0.1
## 4  anxifree no.therapy       0.6
## 5  anxifree no.therapy       0.4
## 6  anxifree no.therapy       0.2
## 7  joyzepam no.therapy       1.4
## 8  joyzepam no.therapy       1.7
## 9  joyzepam no.therapy       1.3
## 10  placebo        CBT       0.6
## 11  placebo        CBT       0.9
## 12  placebo        CBT       0.3
## 13 anxifree        CBT       1.1
## 14 anxifree        CBT       0.8
## 15 anxifree        CBT       1.2
## 16 joyzepam        CBT       1.8
## 17 joyzepam        CBT       1.3
## 18 joyzepam        CBT       1.4
```

For the purposes of this chapter, what we're really interested in is the effect of `drug` on `mood.gain`. The first thing to do is calculate some descriptive statistics and draw some graphs. In Chapter 5 we discussed a variety of different functions that can be used for this purpose. For instance, we can use the `xtabs()` function to see how many people we have in each group:

```
xtabs( ~drug, clin.trial )
```

```
## drug
##  placebo anxifree joyzepam
##        6        6        6
```

Similarly, we can use the `aggregate()` function to calculate means and standard deviations for the `mood.gain` variable broken down by which `drug` was administered:

```
aggregate( mood.gain ~ drug, clin.trial, mean )
```

```
##       drug mood.gain
## 1  placebo 0.4500000
## 2 anxifree 0.7166667
## 3 joyzepam 1.4833333
```

```
aggregate( mood.gain ~ drug, clin.trial, sd )
```

```
##       drug mood.gain
## 1  placebo 0.2810694
## 2 anxifree 0.3920034
## 3 joyzepam 0.2136976
```

Finally, we can use `plotmeans()` from the `gplots` package to produce a pretty picture.

```
library(gplots)
plotmeans(  formula = mood.gain ~ drug,  # plot mood.gain by drug
             data = clin.trial,           # the data frame
             xlab = "Drug Administered",  # x-axis label
             ylab = "Mood Gain",          # y-axis label
             n.label = FALSE              # don't display sample size
 )
```

The results are shown in Figure 14.1, which plots the average mood gain for all three conditions; error bars show 95% confidence intervals. As the plot makes clear, there is a larger improvement in mood for participants in the Joyzepam group than for either the Anxifree group or the placebo group. The Anxifree group shows a larger mood gain than the control group, but the difference isn't as large.

The question that we want to answer is: are these difference "real", or are they just due to chance?

```
## Warning: package 'gplots' was built under R version 3.5.2
```

```
##
## Attaching package: 'gplots'
```

```
## The following object is masked from 'package:stats':
##
##     lowess
```

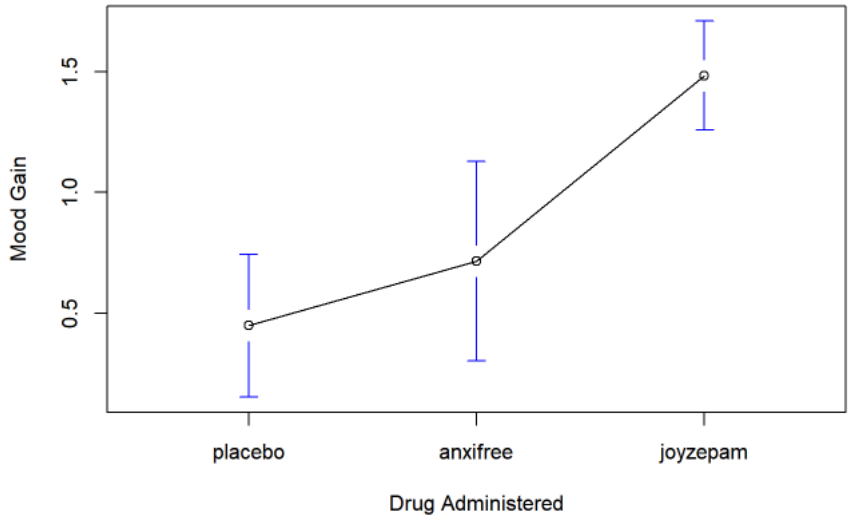


Figure 14.1: Average mood gain as a function of drug administered. Error bars depict 95% confidence intervals associated with each of the group means.

This page titled 12.2: An Illustrative Data Set is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **14.2: An Illustrative Data Set** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

12.3: How ANOVA Works

In order to answer the question posed by our clinical trial data, we're going to run a one-way ANOVA. As usual, I'm going to start by showing you how to do it the hard way, building the statistical tool from the ground up and showing you how you could do it in R if you didn't have access to any of the cool built-in ANOVA functions. And, as always, I hope you'll read it carefully, try to do it the long way once or twice to make sure you really understand how ANOVA works, and then – once you've grasped the concept – never *ever* do it this way again.

The experimental design that I described in the previous section strongly suggests that we're interested in comparing the average mood change for the three different drugs. In that sense, we're talking about an analysis similar to the t-test (Chapter 13, but involving more than two groups. If we let μ_P denote the population mean for the mood change induced by the placebo, and let μ_A and μ_J denote the corresponding means for our two drugs, Anxifree and Joyzepam, then the (somewhat pessimistic) null hypothesis that we want to test is that all three population means are identical: that is, *neither* of the two drugs is any more effective than a placebo. Mathematically, we write this null hypothesis like this:

$$H_0: \text{it is true that } \mu_P=\mu_A=\mu_J$$

As a consequence, our alternative hypothesis is that at least one of the three different treatments is different from the others. It's a little trickier to write this mathematically, because (as we'll discuss) there are quite a few different ways in which the null hypothesis can be false. So for now we'll just write the alternative hypothesis like this:

$$H_1: \text{it is *not* true that } \mu_P=\mu_A=\mu_J$$

This null hypothesis is a lot trickier to test than any of the ones we've seen previously. How shall we do it? A sensible guess would be to "do an ANOVA", since that's the title of the chapter, but it's not particularly clear why an "analysis of *variances*" will help us learn anything useful about the *means*. In fact, this is one of the biggest conceptual difficulties that people have when first encountering ANOVA. To see how this works, I find it most helpful to start by talking about variances. In fact, what I'm going to do is start by playing some mathematical games with the formula that describes the variance. That is, we'll start out by playing around with variances, and it will turn out that this gives us a useful tool for investigating means.

12.3.1 formulas for the variance of Y

Firstly, let's start by introducing some notation. We'll use G to refer to the total number of groups. For our data set, there are three drugs, so there are G=3 groups. Next, we'll use N to refer to the total sample size: there are a total of N=18 people in our data set. Similarly, let's use N_k to denote the number of people in the k-th group. In our fake clinical trial, the sample size is N_k=6 for all three groups.[201] Finally, we'll use Y to denote the outcome variable: in our case, Y refers to mood change. Specifically, we'll use Y_{ik} to refer to the mood change experienced by the i-th member of the k-th group. Similarly, we'll use $\bar{Y}$ to be the average mood change, taken across all 18 people in the experiment, and $\bar{Y}_k$ to refer to the average mood change experienced by the 6 people in group k.

Excellent. Now that we've got our notation sorted out, we can start writing down formulas. To start with, let's recall the formula for the variance that we used in Section 5.2, way back in those kinder days when we were just doing descriptive statistics. The sample variance of Y is defined as follows:

$$\text{Var}(Y) = \frac{1}{N} \sum_{k=1}^{G} \sum_{i=1}^{N_k} \left(Y_{ik} - \bar{Y}\right)^2$$

This formula looks pretty much identical to the formula for the variance in Section 5.2. The only difference is that this time around I've got two summations here: I'm summing over groups (i.e., values for k) and over the people within the groups (i.e., values for i). This is purely a cosmetic detail: if I'd instead used the notation Y_p to refer to the value of the outcome variable for person p in the sample, then I'd only have a single summation. The only reason that we have a double summation here is that I've classified people into groups, and then assigned numbers to people within groups.

A concrete example might be useful here. Let's consider this table, in which we have a total of N=5 people sorted into G=2 groups. Arbitrarily, let's say that the "cool" people are group 1, and the "uncool" people are group 2, and it turns out that we have three cool people (N_1=3) and two uncool people (N_2=2).

name	person (p)	group	group num (k)	index in group (i)	grumpiness (Y_{ik} or Y_p)
Ann	1	cool	1	1	20
Ben	2	cool	1	2	55
Cat	3	cool	1	3	21
Dan	4	uncool	2	1	91
Egg	5	uncool	2	2	22

Notice that I've constructed two different labelling schemes here. We have a "person" variable p, so it would be perfectly sensible to refer to Y_p as the grumpiness of the p-th person in the sample. For instance, the table shows that Dan is the four so we'd say p=4. So, when talking about the grumpiness Y of this "Dan" person, whoever he might be, we could refer to his grumpiness by saying that Y_p=91, for person p=4 that is. However, that's not the only way we could refer to Dan. As an alternative we could note that Dan belongs to the "uncool" group (k=2), and is in fact the first person listed in the uncool group (i=1). So it's equally valid to refer to Dan's grumpiness by saying that Y_{ik}=91, where k=2 and i=1. In other words, each person p corresponds to a unique ik combination, and so the formula that I gave above is actually identical to our original formula for the variance, which would be

$$\mathrm{Var}(Y) = \frac{1}{N}\sum_{p=1}^{N}\left(Y_p - \bar{Y}\right)^2$$

In both formulas, all we're doing is summing over all of the observations in the sample. Most of the time we would just use the simpler Y_p notation: the equation using Y_p is clearly the simpler of the two. However, when doing an ANOVA it's important to keep track of which participants belong in which groups, and we need to use the Y_{ik} notation to do this.

12.3.2 From variances to sums of squares

Okay, now that we've got a good grasp on how the variance is calculated, let's define something called the ***total sum of squares***, which is denoted SS_{tot}. This is very simple: instead of averaging the squared deviations, which is what we do when calculating the variance, we just add them up. So the formula for the total sum of squares is almost identical to the formula for the variance:

$$\mathrm{SS}_{tot} = \sum_{k=1}^{G}\sum_{i=1}^{N_k}\left(Y_{ik} - \bar{Y}\right)^2$$

When we talk about analysing variances in the context of ANOVA, what we're really doing is working with the total sums of squares rather than the actual variance. One very nice thing about the total sum of squares is that we can break it up into two different kinds of variation. Firstly, we can talk about the ***within-group sum of squares***, in which we look to see how different each individual person is from their own group mean:

$$\mathrm{SS}_{w} = \sum_{k=1}^{G}\sum_{i=1}^{N_k}\left(Y_{ik} - \bar{Y}_k\right)^2$$

where $\bar{Y}_k$ is a group mean. In our example, $\bar{Y}_k$ would be the average mood change experienced by those people given the k-th drug. So, instead of comparing individuals to the average of *all* people in the experiment, we're only comparing them to those people in the the same group. As a consequence, you'd expect the value of SS_w to be smaller than the total sum of squares, because it's completely ignoring any group differences – that is, the fact that the drugs (if they work) will have different effects on people's moods.

Next, we can define a third notion of variation which captures *only* the differences between groups. We do this by looking at the differences between the group means $\bar{Y}_k$ and grand mean $\bar{Y}$. In order to quantify the extent of this variation, what we do is calculate the ***between-group sum of squares***:

$$\begin{aligned}\mathrm{SS}_{b} &= \sum_{k=1}^{G}\sum_{i=1}^{N_k}\left(\bar{Y}_k - \bar{Y}\right)^2 \\ &= \sum_{k=1}^{G} N_k\left(\bar{Y}_k - \bar{Y}\right)^2\end{aligned}$$

It's not too difficult to show that the total variation among people in the experiment SS_{tot} is actually the sum of the differences between the groups SS_b and the variation inside the groups SS_w. That is:

$$SS_w+SS_b=SS_{tot}$$

Yay.

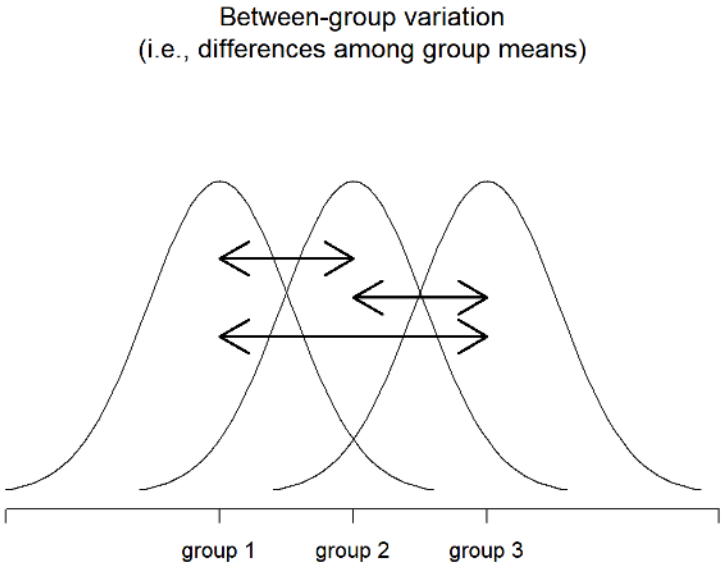


Figure 14.2: Graphical illustration of "between groups" variation

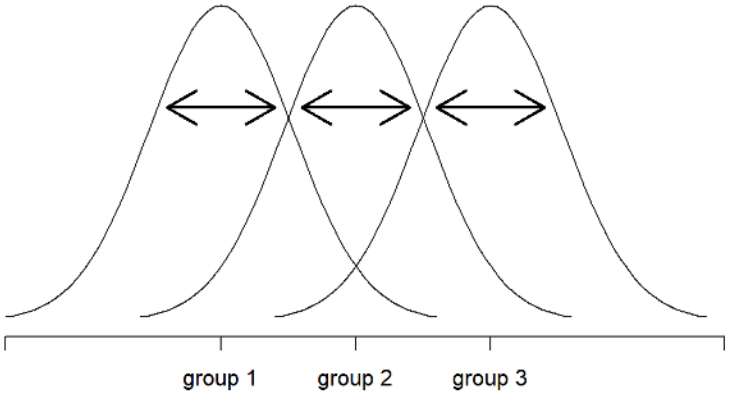


Figure 14.3: Graphical illustration of "within groups" variation

Okay, so what have we found out? We've discovered that the total variability associated with the outcome variable (SS_{tot}) can be mathematically carved up into the sum of "the variation due to the differences in the sample means for the different groups" (SS_b) plus "all the rest of the variation" (SS_w). How does that help me find out whether the groups have different population means? Um. Wait. Hold on a second… now that I think about it, this is *exactly* what we were looking for. If the null hypothesis is true, then you'd expect all the sample means to be pretty similar to each other, right? And that would imply that you'd expect SS_b to be really small, or at least you'd expect it to be a lot smaller than the "the variation associated with everything else", SS_w. Hm. I detect a hypothesis test coming on…

12.3.3 From sums of squares to the F-test

As we saw in the last section, the *qualitative* idea behind ANOVA is to compare the two sums of squares values SS_b and SS_w to each other: if the between-group variation is SS_b is large relative to the within-group variation SS_w then we have reason to suspect that the population means for the different groups aren't identical to each other. In order to convert this into a workable hypothesis test, there's a little bit of "fiddling around" needed. What I'll do is first show you *what* we do to calculate our test statistic – which is called an ***F ratio*** – and then try to give you a feel for *why* we do it this way.

In order to convert our SS values into an F-ratio, the first thing we need to calculate is the ***degrees of freedom*** associated with the SSb and SS_w values. As usual, the degrees of freedom corresponds to the number of unique "data points" that contribute to a particular calculation, minus the number of "constraints" that they need to satisfy. For the within-groups variability, what we're calculating is the variation of the individual observations (N data points) around the group means (G constraints). In contrast, for the between groups variability, we're interested in the variation of the group means (G data points) around the grand mean (1 constraint). Therefore, the degrees of freedom here are:

$$df_b = G - 1$$

$$df_w = N - G$$

Okay, that seems simple enough. What we do next is convert our summed squares value into a "mean squares" value, which we do by dividing by the degrees of freedom:

$$MS_b = \frac{SS_b}{df_b}$$

$$MS_w = \frac{SS_w}{df_w}$$

Finally, we calculate the F-ratio by dividing the between-groups MS by the within-groups MS:

$$F = \frac{MS_b}{MS_w}$$

At a very general level, the intuition behind the F statistic is straightforward: bigger values of F means that the between-groups variation is large, relative to the within-groups variation. As a consequence, the larger the value of F, the more evidence we have against the null hypothesis. But how large does F have to be in order to actually *reject* H_0? In order to understand this, you need a slightly deeper understanding of what ANOVA is and what the mean squares values actually are.

The next section discusses that in a bit of detail, but for readers that aren't interested in the details of what the test is actually measuring, I'll cut to the chase. In order to complete our hypothesis test, we need to know the sampling distribution for F if the null hypothesis is true. Not surprisingly, the sampling distribution for the F *statistic* under the null hypothesis is an F *distribution*. If you recall back to our discussion of the F distribution in Chapter @ref(probability, the F distribution has two parameters, corresponding to the two degrees of freedom involved: the first one df1 is the between groups degrees of freedom dfb, and the second one df2 is the within groups degrees of freedom df_w.

A summary of all the key quantities involved in a one-way ANOVA, including the formulas showing how they are calculated, is shown in Table 14.1.

Table 14.1: All of the key quantities involved in an ANOVA, organised into a "standard" ANOVA table. The formulas for all quantities (except the p-value, which has a very ugly formula and would be nightmarishly hard to calculate without a computer) are shown.

	df	sum of squares	mean squares	F statistic	p value
between groups	$df_b = G-1$	$SS_b = \sum_{k=1}^{G} N_k \left(\bar{Y}_k - \bar{Y}\right)^2$	$MS_b = \frac{SS_b}{df_b}$	$F = \frac{MS_b}{MS_w}$	[complicated]
within groups	$df_w = N-G$	$SS_w = \sum_{k=1}^{G} \sum_{i=1}^{N_k} \left(Y_{ik} - \bar{Y}_k\right)^2$	$MS_w = \frac{SS_w}{df_w}$	-	-

12.3.4 model for the data and the meaning of F (advanced)

At a fundamental level, ANOVA is a competition between two different statistical models, H_0 and H_1. When I described the null and alternative hypotheses at the start of the section, I was a little imprecise about what these models actually are. I'll remedy that now, though you probably won't like me for doing so. If you recall, our null hypothesis was that all of the group means are identical to one another. If so, then a natural way to think about the outcome variable Y_{ik} is to describe individual scores in terms of a single population mean μ, plus the deviation from that population mean. This deviation is usually denoted ϵ_{ik} and is traditionally called the *error* or ***residual*** associated with that observation. Be careful though: just like we saw with the word "significant", the word "error" has a technical meaning in statistics that isn't quite the same as its everyday English definition. In everyday language, "error" implies a mistake of some kind; in statistics, it doesn't (or at least, not necessarily). With that in mind, the word "residual"

is a better term than the word "error". In statistics, both words mean "leftover variability": that is, "stuff" that the model can't explain. In any case, here's what the null hypothesis looks like when we write it as a statistical model:

$$Y_{ik} = \mu + \epsilon_{ik}$$

where we make the *assumption* (discussed later) that the residual values ϵ_{ik} are normally distributed, with mean 0 and a standard deviation σ that is the same for all groups. To use the notation that we introduced in Chapter 9 we would write this assumption like this:

$$\epsilon_{ik} \sim \text{Normal}(0, \sigma^2)$$

What about the alternative hypothesis, H_1? The only difference between the null hypothesis and the alternative hypothesis is that we allow each group to have a different population mean. So, if we let μk denote the population mean for the k-th group in our experiment, then the statistical model corresponding to H_1 is:

$$Y_{ik} = \mu_k + \epsilon_{ik}$$

where, once again, we assume that the error terms are normally distributed with mean 0 and standard deviation σ. That is, the alternative hypothesis also assumes that

$$\epsilon \sim \text{Normal}(0, \sigma^2)$$

Okay, now that we've described the statistical models underpinning H_0 and H_1 in more detail, it's now pretty straightforward to say what the mean square values are measuring, and what this means for the interpretation of F. I won't bore you with the proof of this, but it turns out that the within-groups mean square, MS_w, can be viewed as an estimator (in the technical sense: Chapter 10 of the error variance σ^2. The between-groups mean square MS_b is also an estimator; but what it estimates is the error variance *plus* a quantity that depends on the true differences among the group means. If we call this quantity Q, then we can see that the F-statistic is basically[202]

$$F = \frac{\hat{Q} + \hat{\sigma}^2}{\hat{\sigma}^2}$$

where the true value Q=0 if the null hypothesis is true, and Q>0 if the alternative hypothesis is true (e.g. ch. 10 Hays 1994). Therefore, at a bare minimum *the F value must be larger than 1* to have any chance of rejecting the null hypothesis. Note that this *doesn't* mean that it's impossible to get an F-value less than 1. What it means is that, if the null hypothesis is true the sampling distribution of the F ratio has a mean of 1,[203] and so we need to see F-values larger than 1 in order to safely reject the null.

To be a bit more precise about the sampling distribution, notice that if the null hypothesis is true, both MS_b and MS_w are estimators of the variance of the residuals ϵ_{ik}. If those residuals are normally distributed, then you might suspect that the estimate of the variance of ϵ_{ik} is chi-square distributed… because (as discussed in Section 9.6 that's what a chi-square distribution *is*: it's what you get when you square a bunch of normally-distributed things and add them up. And since the F distribution is (again, by definition) what you get when you take the ratio between two things that are X^2 distributed… we have our sampling distribution. Obviously, I'm glossing over a whole lot of stuff when I say this, but in broad terms, this really is where our sampling distribution comes from.

12.3.5 worked example

The previous discussion was fairly abstract, and a little on the technical side, so I think that at this point it might be useful to see a worked example. For that, let's go back to the clinical trial data that I introduced at the start of the chapter. The descriptive statistics that we calculated at the beginning tell us our group means: an average mood gain of 0.45 for the placebo, 0.72 for Anxifree, and 1.48 for Joyzepam. With that in mind, let's party like it's 1899[204] and start doing some pencil and paper calculations. I'll only do this for the first 5 observations, because it's not bloody 1899 and I'm very lazy. Let's start by calculating SS_w, the within-group sums of squares. First, let's draw up a nice table to help us with our calculations…

group (k)	outcome (Y_{ik})
placebo	0.5
placebo	0.3
placebo	0.1
anxifree	0.6
anxifree	0.4

At this stage, the only thing I've included in the table is the raw data itself: that is, the grouping variable (i.e., `drug`) and outcome variable (i.e. `mood.gain`) for each person. Note that the outcome variable here corresponds to the Y_{ik} value in our equation previously. The next step in the calculation is to write down, for each person in the study, the corresponding group mean; that is, $\bar{Y}_k$. This is slightly repetitive, but not particularly difficult since we already calculated those group means when doing our descriptive statistics:

group (k)	outcome (Y_{ik})	**group mean ($\bar{Y}_k$**
placebo	0.5	**0.45**
placebo	0.3	**0.45**
placebo	0.1	**0.45**
anxifree	0.6	**0.72**
anxifree	0.4	**0.72**

Now that we've written those down, we need to calculate – again for every person – the deviation from the corresponding group mean. That is, we want to subtract $Y_{ik} - \bar{Y}_k$. After we've done that, we need to square everything. When we do that, here's what we get:

group (k)	outcome (Y_{ik})	group mean ($\bar{Y}_k$)	**dev. from group mean** ($Y_{ik} - \bar{Y}_k$)	**squared deviation** ($(Y_{ik} - \bar{Y}_k)^2$)
placebo	0.5	0.45	**0.05**	**0.0025**
placebo	0.3	0.45	**-0.15**	**0.0225**
placebo	0.1	0.45	**-0.35**	**0.1225**
anxifree	0.6	0.72	**-0.12**	**0.0136**
anxifree	0.4	0.72	**-0.32**	**0.1003**

The last step is equally straightforward. In order to calculate the within-group sum of squares, we just add up the squared deviations across all observations:

$$\begin{aligned} \text{SS}_w &= 0.0025 + 0.0225 + 0.1225 + 0.0136 + 0.1003 \\ &= 0.2614 \end{aligned}$$

Of course, if we actually wanted to get the *right* answer, we'd need to do this for all 18 observations in the data set, not just the first five. We could continue with the pencil and paper calculations if we wanted to, but it's pretty tedious. Alternatively, it's not too hard to get R to do it. Here's how:

```
outcome <- clin.trial$mood.gain
group <- clin.trial$drug
gp.means <- tapply(outcome,group,mean)
gp.means <- gp.means[group]
dev.from.gp.means <- outcome - gp.means
squared.devs <- dev.from.gp.means ^2
```

It might not be obvious from inspection what these commands are doing: as a general rule, the human brain seems to just shut down when faced with a big block of programming. However, I strongly suggest that – if you're like me and tend to find that the mere sight of this code makes you want to look away and see if there's any beer left in the fridge or a game of footy on the telly – you take a moment and look closely at these commands one at a time. Every single one of these commands is something you've seen before somewhere else in the book. There's nothing novel about them (though I'll have to admit that the `tapply()` function takes a while to get a handle on), so if you're not quite sure how these commands work, this might be a good time to try playing around with them yourself, to try to get a sense of what's happening. On the other hand, if this does seem to make sense, then you won't be all that surprised at what happens when I wrap these variables in a data frame, and print it out…

```
Y <- data.frame( group, outcome, gp.means,
                 dev.from.gp.means, squared.devs )
print(Y, digits = 2)
```

```
##       group outcome gp.means dev.from.gp.means squared.devs
## 1   placebo     0.5     0.45             0.050       0.0025
## 2   placebo     0.3     0.45            -0.150       0.0225
## 3   placebo     0.1     0.45            -0.350       0.1225
## 4  anxifree     0.6     0.72            -0.117       0.0136
## 5  anxifree     0.4     0.72            -0.317       0.1003
## 6  anxifree     0.2     0.72            -0.517       0.2669
## 7  joyzepam     1.4     1.48            -0.083       0.0069
## 8  joyzepam     1.7     1.48             0.217       0.0469
## 9  joyzepam     1.3     1.48            -0.183       0.0336
## 10  placebo     0.6     0.45             0.150       0.0225
## 11  placebo     0.9     0.45             0.450       0.2025
## 12  placebo     0.3     0.45            -0.150       0.0225
## 13 anxifree     1.1     0.72             0.383       0.1469
## 14 anxifree     0.8     0.72             0.083       0.0069
## 15 anxifree     1.2     0.72             0.483       0.2336
## 16 joyzepam     1.8     1.48             0.317       0.1003
## 17 joyzepam     1.3     1.48            -0.183       0.0336
## 18 joyzepam     1.4     1.48            -0.083       0.0069
```

If you compare this output to the contents of the table I've been constructing by hand, you can see that R has done exactly the same calculations that I was doing, and much faster too. So, if we want to finish the calculations of the within-group sum of squares in R, we just ask for the `sum()` of the `squared.devs` variable:

```
SSw <- sum( squared.devs )
print( SSw )
```

```
## [1] 1.391667
```

Obviously, this isn't the same as what I calculated, because R used all 18 observations. But if I'd typed `sum( squared.devs[1:5] )` instead, it would have given the same answer that I got earlier.

Okay. Now that we've calculated the within groups variation, SS_w, it's time to turn our attention to the between-group sum of squares, SS_b. The calculations for this case are very similar. The main difference is that, instead of calculating the differences between an observation Y_{ik} and a group mean $\bar{Y}_k$ for all of the observations, we calculate the differences between the group means $\bar{Y}_k$ and the grand mean $\bar{Y}$ (in this case 0.88) for all of the groups…

group (k)	group mean ($\bar{Y}_k$)	grand mean ($\bar{Y}$)	deviation ($\bar{Y}_k - \bar{Y}$)	squared deviations (($\bar{Y}_k - \bar{Y})^2$)
placebo	0.45	0.88	-0.43	0.18
anxifree	0.72	0.88	-0.16	0.03
joyzepam	1.48	0.88	0.60	0.36

However, for the between group calculations we need to multiply each of these squared deviations by N_k, the number of observations in the group. We do this because every *observation* in the group (all N_k of them) is associated with a between group difference. So if there are six people in the placebo group, and the placebo group mean differs from the grand mean by 0.19, then the *total* between group variation associated with these six people is 6×0.16=1.14. So we have to extend our little table of calculations…

group (k)	squared deviations (($\bar{Y}_k - \bar{Y})^2$)	sample size (N_k)	weighted squared dev (N_k($\bar{Y}_k - \bar{Y})^2$)
placebo	0.18	6	1.11
anxifree	0.03	6	0.16
joyzepam	0.36	6	2.18

And so now our between group sum of squares is obtained by summing these "weighted squared deviations" over all three groups in the study:

$$SS_b = 1.11 + 0.16 + 2.18 = 3.45$$

As you can see, the between group calculations are a lot shorter, so you probably wouldn't usually want to bother using R as your calculator. However, if you *did* decide to do so, here's one way you could do it:

```
gp.means <- tapply(outcome,group,mean)
grand.mean <- mean(outcome)
dev.from.grand.mean <- gp.means - grand.mean
squared.devs <- dev.from.grand.mean ^2
gp.sizes <- tapply(outcome,group,length)
wt.squared.devs <- gp.sizes * squared.devs
```

Again, I won't actually try to explain this code line by line, but – just like last time – there's nothing in there that we haven't seen in several places elsewhere in the book, so I'll leave it as an exercise for you to make sure you understand it. Once again, we can dump all our variables into a data frame so that we can print it out as a nice table:

```
Y <- data.frame( gp.means, grand.mean, dev.from.grand.mean,
                 squared.devs, gp.sizes, wt.squared.devs )
print(Y, digits = 2)
```

```
##          gp.means grand.mean dev.from.grand.mean squared.devs gp.sizes
## placebo      0.45       0.88               -0.43        0.188        6
## anxifree     0.72       0.88               -0.17        0.028        6
## joyzepam     1.48       0.88                0.60        0.360        6
##          wt.squared.devs
## placebo             1.13
## anxifree            0.17
## joyzepam            2.16
```

Clearly, these are basically the same numbers that we got before. There are a few tiny differences, but that's only because the hand-calculated versions have some small errors caused by the fact that I rounded all my numbers to 2 decimal places at each step in the calculations, whereas R only does it at the end (obviously, R s version is more accurate). Anyway, here's the R command showing the final step:

```
SSb <- sum( wt.squared.devs )
print( SSb )
```

```
## [1] 3.453333
```

which is (ignoring the slight differences due to rounding error) the same answer that I got when doing things by hand.

Now that we've calculated our sums of squares values, SS_b and SS_w, the rest of the ANOVA is pretty painless. The next step is to calculate the degrees of freedom. Since we have G=3 groups and N=18 observations in total, our degrees of freedom can be calculated by simple subtraction:

$$df_b = G - 1 = 2$$

$$df_w = N - G = 15$$

Next, since we've now calculated the values for the sums of squares and the degrees of freedom, for both the within-groups variability and the between-groups variability, we can obtain the mean square values by dividing one by the other:

$$MS_b = \frac{SS_b}{df_b} = \frac{3.45}{2} = 1.73$$
$$MS_w = \frac{SS_w}{df_w} = \frac{1.39}{15} = 0.09$$

We're almost done. The mean square values can be used to calculate the F-value, which is the test statistic that we're interested in. We do this by dividing the between-groups MS value by the and within-groups MS value.

$$F = \frac{MS_b}{MS_w} = \frac{1.73}{0.09} = 18.6$$

Woohooo! This is terribly exciting, yes? Now that we have our test statistic, the last step is to find out whether the test itself gives us a significant result. As discussed in Chapter @ref(hypothesistesting, what we really *ought* to do is choose an α level (i.e., acceptable Type I error rate) ahead of time, construct our rejection region, etc etc. But in practice it's just easier to directly calculate the p-value. Back in the "old days", what we'd do is open up a statistics textbook or something and flick to the back section which would actually have a huge lookup table… that's how we'd "compute" our p-value, because it's too much effort to do it any other way. However, since we have access to R, I'll use the `pf()` function to do it instead. Now, remember that I explained earlier that the F-test is always one sided? And that we only reject the null hypothesis for very large F-values? That means we're only interested in the *upper tail* of the F-distribution. The command that you'd use here would be this…

```
pf( 18.6, df1 = 2, df2 = 15, lower.tail = FALSE)
```

```
## [1] 8.672727e-05
```

Therefore, our p-value comes to 0.0000867, or 8.67×10^{-5} in scientific notation. So, unless we're being *extremely* conservative about our Type I error rate, we're pretty much guaranteed to reject the null hypothesis.

At this point, we're basically done. Having completed our calculations, it's traditional to organise all these numbers into an ANOVA table like the one in Table@reftab:anovatable. For our clinical trial data, the ANOVA table would look like this:

	df	sum of squares	mean squares	F-statistic	p-value
between groups	2	3.45	1.73	18.6	8.67×10^{-5}
within groups	15	1.39	0.09	-	-

These days, you'll probably never have much reason to want to construct one of these tables yourself, but you *will* find that almost all statistical software (R included) tends to organise the output of an ANOVA into a table like this, so it's a good idea to get used to reading them. However, although the software will output a full ANOVA table, there's almost never a good reason to include the whole table in your write up. A pretty standard way of reporting this result would be to write something like this:

One-way ANOVA showed a significant effect of drug on mood gain ($F(2,15)=18.6, p<.001$).

Sigh. So much work for one short sentence.

This page titled 12.3: How ANOVA Works is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **14.3: How ANOVA Works** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

12.4: Running an ANOVA in R

I'm pretty sure I know what you're thinking after reading the last section, *especially* if you followed my advice and tried typing all the commands in yourself…. doing the ANOVA calculations yourself *sucks*. There's quite a lot of calculations that we needed to do along the way, and it would be tedious to have to do this over and over again every time you wanted to do an ANOVA. One possible solution to the problem would be to take all these calculations and turn them into some R functions yourself. You'd still have to do a lot of typing, but at least you'd only have to do it the one time: once you've created the functions, you can reuse them over and over again. However, writing your own functions is a lot of work, so this is kind of a last resort. Besides, it's much better if someone else does all the work for you…

12.4.1 Using the `aov()` function to specify your ANOVA

To make life easier for you, R provides a function called `aov()`, which – obviously – is an acronym of "Analysis Of Variance."[205] If you type `?aov` and have a look at the help documentation, you'll see that there are several arguments to the `aov()` function, but the only two that we're interested in are `formula` and `data`. As we've seen in a few places previously, the `formula` argument is what you use to specify the outcome variable and the grouping variable, and the `data` argument is what you use to specify the data frame that stores these variables. In other words, to do the same ANOVA that I laboriously calculated in the previous section, I'd use a command like this:

```
aov( formula = mood.gain ~ drug, data = clin.trial )
```

Actually, that's not *quite* the whole story, as you'll see as soon as you look at the output from this command, which I've hidden for the moment in order to avoid confusing you. Before we go into specifics, I should point out that either of these commands will do the same thing:

```
aov( clin.trial$mood.gain ~ clin.trial$drug )
aov( mood.gain ~ drug, clin.trial )
```

In the first command, I didn't specify a `data` set, and instead relied on the `$` operator to tell R how to find the variables. In the second command, I dropped the argument names, which is okay in this case because `formula` is the first argument to the `aov()` function, and `data` is the second one. Regardless of how I specify the ANOVA, I can assign the output of the `aov()` function to a variable, like this for example:

```
my.anova <- aov( mood.gain ~ drug, clin.trial )
```

This is almost always a good thing to do, because there's *lots* of useful things that we can do with the `my.anova` variable. So let's assume that it's this last command that I used to specify the ANOVA that I'm trying to run, and as a consequence I have this `my.anova` variable sitting in my workspace, waiting for me to do something with it…

12.4.2 Understanding what the `aov()` function produces

Now that we've seen how to use the `aov()` function to create `my.anova` we'd better have a look at what this variable actually is. The first thing to do is to check to see what class of variable we've created, since it's kind of interesting in this case. When we do that…

```
class( my.anova )
```

```
## [1] "aov" "lm"
```

… we discover that `my.anova` actually has *two* classes! The first class tells us that it's an `aov` (analysis of variance) object, but the second tells us that it's *also* an `lm` (linear model) object. Later on, we'll see that this reflects a pretty deep statistical

relationship between ANOVA and regression (Chapter 15 and it means that any function that exists in R for dealing with regressions can also be applied to `aov` objects, which is neat; but I'm getting ahead of myself. For now, I want to note that what we've created is an `aov` object, and to also make the point that `aov` objects are actually rather complicated beasts. I *won't* be trying to explain everything about them, since it's way beyond the scope of an introductory statistics subject, but to give you a tiny hint of some of the stuff that R stores inside an `aov` object, let's ask it to print out the `names()` of all the stored quantities…

```
names( my.anova )
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "contrasts"     "xlevels"       "call"          "terms"
## [13] "model"
```

As we go through the rest of the book, I hope that a few of these will become a little more obvious to you, but right now that's going to look pretty damned opaque. That's okay. You don't need to know any of the details about it right now, and most of it you don't need at all… what you *do* need to understand is that the `aov()` function does a lot of calculations for you, not just the basic ones that I outlined in the previous sections. What this means is that it's generally a good idea to create a variable like `my.anova` that stores the output of the `aov()` function… because later on, you can use `my.anova` as an input to lots of other functions: those other functions can pull out bits and pieces from the `aov` object, and calculate various other things that you might need.

Right then. The simplest thing you can do with an `aov` object is to `print()` it out. When we do that, it shows us a few of the key quantities of interest:

```
print( my.anova )
```

```
## Call:
##    aov(formula = mood.gain ~ drug, data = clin.trial)
##
## Terms:
##                     drug Residuals
## Sum of Squares  3.453333  1.391667
## Deg. of Freedom        2        15
##
## Residual standard error: 0.3045944
## Estimated effects may be unbalanced
```

Specificially, it prints out a reminder of the command that you used when you called `aov()` in the first place, shows you the sums of squares values, the degrees of freedom, and a couple of other quantities that we're not really interested in right now. Notice, however, that R doesn't use the names "between-group" and "within-group". Instead, it tries to assign more meaningful names: in our particular example, the *between groups* variance corresponds to the effect that the `drug` has on the outcome variable; and the *within groups* variance is corresponds to the "leftover" variability, so it calls that the *residuals*. If we compare these numbers to the numbers that I calculated by hand in Section 14.2.5, you can see that they're identical… the between groups sums of squares is SS_b=3.45, the within groups sums of squares is SS_w=1.39, and the degrees of freedom are 2 and 15 repectively.

12.4.3 Running the hypothesis tests for the ANOVA

Okay, so we've verified that `my.anova` seems to be storing a bunch of the numbers that we're looking for, but the `print()` function didn't quite give us the output that we really wanted. Where's the F-value? The p-value? These are the most important numbers in our hypothesis test, but the `print()` function doesn't provide them. To get those numbers, we need to use a

different function. Instead of asking R to `print()` out the `aov` object, we should have asked for a `summary()` of it.[206] When we do that…

```
summary( my.anova )
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## drug         2  3.453  1.7267   18.61 8.65e-05 ***
## Residuals   15  1.392  0.0928
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

… we get all of the key numbers that we calculated earlier. We get the sums of squares, the degrees of freedom, the mean squares, the F-statistic, and the p-value itself. These are all identical to the numbers that we calculated ourselves when doing it the long and tedious way, and it's even organised into the same kind of ANOVA table that I showed in Table 14.1, and then filled out by hand in Section 14.2.5. The only things that are even slightly different is that some of the row and column names are a bit different.

This page titled 12.4: Running an ANOVA in R is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **14.4: Running an ANOVA in R** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

12.5: Effect Size

There's a few different ways you could measure the effect size in an ANOVA, but the most commonly used measures are η2 (***eta squared***) and partial η^2. For a one way analysis of variance they're identical to each other, so for the moment I'll just explain η^2. The definition of η^2 is actually really simple:

$$\eta^2 = \frac{\text{SS}_b}{\text{SS}_{tot}}$$

That's all it is. So when I look at the ANOVA table above, I see that SS_b=3.45 and SS_{tot}=3.45+1.39=4.84. Thus we get an η2 value of

$$\eta^2 = \frac{3.45}{4.84} = 0.71$$

The interpretation of η2 is equally straightforward: it refers to the proportion of the variability in the outcome variable (`mood.gain`) that can be explained in terms of the predictor (`drug`). A value of η2=0 means that there is no relationship at all between the two, whereas a value of η^2=1 means that the relationship is perfect. Better yet, the η^2 value is very closely related to a squared correlation (i.e., r^2). So, if you're trying to figure out whether a particular value of η^2 is big or small, it's sometimes useful to remember that

$$\eta = \sqrt{\frac{SS_b}{SS_{tot}}}$$

can be interpreted as if it referred to the *magnitude* of a Pearson correlation. So in our drugs example, the η^2 value of .71 corresponds to an η value of $\sqrt{.71}$ =.84. If we think about this as being equivalent to a correlation of about .84, we'd conclude that the relationship between `drug` and `mood.gain` is strong.

The core packages in R don't include any functions for calculating η^2. However, it's pretty straightforward to calculate it directly from the numbers in the ANOVA table. In fact, since I've already got the `SSw` and `SSb` variables lying around from my earlier calculations, I can do this:

```
SStot <- SSb + SSw          # total sums of squares
eta.squared <- SSb / SStot  # eta-squared value
print( eta.squared )
```

```
## [1] 0.7127623
```

However, since it can be tedious to do this the long way (especially when we start running more complicated ANOVAs, such as those in Chapter 16 I've included an `etaSquared()` function in the `lsr` package which will do it for you. For now, the only argument you need to care about is `x` , which should be the `aov` object corresponding to your ANOVA. When we do this, what we get as output is this:

```
etaSquared( x = my.anova )
```

```
##         eta.sq eta.sq.part
## drug 0.7127623   0.7127623
```

The output here shows two different numbers. The first one corresponds to the η^2 statistic, precisely as described above. The second one refers to "partial η^2", which is a somewhat different measure of effect size that I'll describe later. For the simple ANOVA that we've just run, they're the same number. But this won't always be true once we start running more complicated ANOVAs.[207]

This page titled 12.5: Effect Size is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **14.5: Effect Size** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

12.6: Multiple Comparisons and Post Hoc Tests

Any time you run an ANOVA with more than two groups, and you end up with a significant effect, the first thing you'll probably want to ask is which groups are actually different from one another. In our drugs example, our null hypothesis was that all three drugs (placebo, Anxifree and Joyzepam) have the exact same effect on mood. But if you think about it, the null hypothesis is actually claiming *three* different things all at once here. Specifically, it claims that:

- Your competitor's drug (Anxifree) is no better than a placebo (i.e., $\mu_A=\mu_P$)
- Your drug (Joyzepam) is no better than a placebo (i.e., $\mu_J=\mu_P$)
- Anxifree and Joyzepam are equally effective (i.e., $\mu_J=\mu_A$)

If any one of those three claims is false, then the null hypothesis is also false. So, now that we've rejected our null hypothesis, we're thinking that *at least* one of those things isn't true. But which ones? All three of these propositions are of interest: you certainly want to know if your new drug Joyzepam is better than a placebo, and it would be nice to know how well it stacks up against an existing commercial alternative (i.e., Anxifree). It would even be useful to check the performance of Anxifree against the placebo: even if Anxifree has already been extensively tested against placebos by other researchers, it can still be very useful to check that your study is producing similar results to earlier work.

When we characterise the null hypothesis in terms of these three distinct propositions, it becomes clear that there are eight possible "states of the world" that we need to distinguish between:

possibility:	is $\mu_P=\mu_A$?	is $\mu_P=\mu_J$?	is $\mu_A=\mu_J$?	which hypothesis?
1	✓	✓	✓	null
2	✓	✓		alternative
3	✓		✓	alternative
4	✓			alternative
5		✓	✓	alternative
6		✓		alternative
7			✓	alternative
8				alternative

By rejecting the null hypothesis, we've decided that we *don't* believe that #1 is the true state of the world. The next question to ask is, which of the other seven possibilities *do* we think is right? When faced with this situation, its usually helps to look at the data. For instance, if we look at the plots in Figure 14.1, it's tempting to conclude that Joyzepam is better than the placebo and better than Anxifree, but there's no real difference between Anxifree and the placebo. However, if we want to get a clearer answer about this, it might help to run some tests.

12.6.1 Running "pairwise" t-tests

How might we go about solving our problem? Given that we've got three separate pairs of means (placebo versus Anxifree, placebo versus Joyzepam, and Anxifree versus Joyzepam) to compare, what we could do is run three separate t-tests and see what happens. There's a couple of ways that we could do this. One method would be to construct new variables corresponding the groups you want to compare (e.g., `anxifree` , `placebo` and `joyzepam`), and then run a t-test on these new variables:

```
t.test( anxifree, placebo, var.equal = TRUE )   # Student t-test

anxifree <- with(clin.trial, mood.gain[drug == "anxifree"])  # mood change due to anx
placebo <- with(clin.trial, mood.gain[drug == "placebo"])    # mood change due to pla
```

or, you could use the `subset` argument in the `t.test()` function to select only those observations corresponding to one of the two groups we're interested in:

```
t.test( formula = mood.gain ~ drug,
        data = clin.trial,
        subset = drug %in% c("placebo","anxifree"),
        var.equal = TRUE
)
```

See Chapter 7 if you've forgotten how the `%in%` operator works. Regardless of which version we do, R will print out the results of the t-test, though I haven't included that output here. If we go on to do this for all possible pairs of variables, we can look to see which (if any) pairs of groups are significantly different to each other. This "lots of t-tests idea" isn't a bad strategy, though as we'll see later on there are some problems with it. However, for the moment our bigger problem is that it's a *pain* to have to type in such a long command over and over again: for instance, if your experiment has 10 groups, then you have to run 45 t-tests. That's way too much typing.

To help keep the typing to a minimum, R provides a function called `pairwise.t.test()` that automatically runs all of the t-tests for you. There are three arguments that you need to specify, the outcome variable `x`, the group variable `g`, and the `p.adjust.method` argument, which "adjusts" the p-value in one way or another. I'll explain p-value adjustment in a moment, but for now we can just set `p.adjust.method = "none"` since we're not doing any adjustments. For our example, here's what we do:

```
pairwise.t.test( x = clin.trial$mood.gain,    # outcome variable
                 g = clin.trial$drug,         # grouping variable
                 p.adjust.method = "none"     # which correction to use?
 )
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  clin.trial$mood.gain and clin.trial$drug
##
##          placebo anxifree
## anxifree 0.15021 -
## joyzepam 3e-05   0.00056
##
## P value adjustment method: none
```

One thing that bugs me slightly about the `pairwise.t.test()` function is that you can't just give it an `aov` object, and have it produce this output. After all, I went to all that trouble earlier of getting R to create the `my.anova` variable and – as we saw in Section 14.3.2 – R has actually stored enough information inside it that I should just be able to get it to run all the pairwise tests using `my.anova` as an input. To that end, I've included a `posthocPairwiseT()` function in the `lsr` package that lets you do this. The idea behind this function is that you can just input the `aov` object itself,[208] and then get the pairwise tests as an output. As of the current writing, `posthocPairwiseT()` is actually just a simple way of calling `pairwise.t.test()` function, but you should be aware that I intend to make some changes to it later on. Here's an example:

```
posthocPairwiseT( x = my.anova, p.adjust.method = "none" )
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  mood.gain and drug
##
##          placebo anxifree
## anxifree 0.15021 -
## joyzepam 3e-05   0.00056
##
## P value adjustment method: none
```

In later versions, I plan to add more functionality (e.g., adjusted confidence intervals), but for now I think it's at least kind of useful. To see why, let's suppose you've run your ANOVA and stored the results in `my.anova`, and you're happy using the Holm correction (the default method in `pairwise.t.test()`, which I'll explain this in a moment). In that case, all you have to do is type this:

```
posthocPairwiseT( my.anova )
```

and R will output the test results. Much more convenient, I think.

12.6.2 Corrections for multiple testing

In the previous section I hinted that there's a problem with just running lots and lots of t-tests. The concern is that when running these analyses, what we're doing is going on a "fishing expedition": we're running lots and lots of tests without much theoretical guidance, in the hope that some of them come up significant. This kind of theory-free search for group differences is referred to as ***post hoc analysis*** ("post hoc" being Latin for "after this").[209]

It's okay to run post hoc analyses, but a lot of care is required. For instance, the analysis that I ran in the previous section is actually pretty dangerous: each *individual* t-test is designed to have a 5% Type I error rate (i.e., α=.05), and I ran three of these tests. Imagine what would have happened if my ANOVA involved 10 different groups, and I had decided to run 45 "post hoc" t-tests to try to find out which ones were significantly different from each other, you'd expect 2 or 3 of them to come up significant *by chance alone*. As we saw in Chapter 11, the central organising principle behind null hypothesis testing is that we seek to control our Type I error rate, but now that I'm running lots of t-tests at once, in order to determine the source of my ANOVA results, my actual Type I error rate across this whole *family* of tests has gotten completely out of control.

The usual solution to this problem is to introduce an adjustment to the p-value, which aims to control the total error rate across the family of tests (see Shaffer 1995). An adjustment of this form, which is usually (but not always) applied because one is doing post hoc analysis, is often referred to as a ***correction for multiple comparisons***, though it is sometimes referred to as "simultaneous inference". In any case, there are quite a few different ways of doing this adjustment. I'll discuss a few of them in this section and in Section 16.8, but you should be aware that there are many other methods out there (see, e.g., Hsu 1996).

12.6.3 Bonferroni corrections

The simplest of these adjustments is called the ***Bonferroni correction*** (Dunn 1961), and it's very very simple indeed. Suppose that my post hoc analysis consists of m separate tests, and I want to ensure that the total probability of making *any* Type I errors at all is at most α.[210] If so, then the Bonferroni correction just says "multiply all your raw p-values by m". If we let p denote the original p-value, and let p'_j be the corrected value, then the Bonferroni correction tells that:

$$p' = m \times p$$

And therefore, if you're using the Bonferroni correction, you would reject the null hypothesis if $p' < \alpha$. The logic behind this correction is very straightforward. We're doing m different tests; so if we arrange it so that each test has a Type I error rate of at most α/m, then the *total* Type I error rate across these tests cannot be larger than α. That's pretty simple, so much so that in the original paper, the author writes:

The method given here is so simple and so general that I am sure it must have been used before this. I do not find it, however, so can only conclude that perhaps its very simplicity has kept statisticians from realizing that it is a very good method in some situations (pp 52-53 Dunn 1961)

To use the Bonferroni correction in R, you can use the `pairwise.t.test()` function,[211] making sure that you set `p.adjust.method = "bonferroni"`. Alternatively, since the whole reason why we're doing these pairwise tests in the first place is because we have an ANOVA that we're trying to understand, it's probably more convenient to use the `posthocPairwiseT()` function in the `lsr` package, since we can use `my.anova` as the input:

```
posthocPairwiseT( my.anova, p.adjust.method = "bonferroni")
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  mood.gain and drug
##
##          placebo anxifree
## anxifree 0.4506  -
## joyzepam 9.1e-05 0.0017
##
## P value adjustment method: bonferroni
```

If we compare these three p-values to those that we saw in the previous section when we made no adjustment at all, it is clear that the only thing that R has done is multiply them by 3.

12.6.4 Holm corrections

Although the Bonferroni correction is the simplest adjustment out there, it's not usually the best one to use. One method that is often used instead is the ***Holm correction*** (Holm 1979). The idea behind the Holm correction is to pretend that you're doing the tests sequentially; starting with the smallest (raw) p-value and moving onto the largest one. For the j-th largest of the p-values, the adjustment is *either*

$$p'_j = j \times p_j$$

(i.e., the biggest p-value remains unchanged, the second biggest p-value is doubled, the third biggest p-value is tripled, and so on), *or*

$$p'_j = p'_{j+1}$$

whichever one is *larger*. This might sound a little confusing, so let's go through it a little more slowly. Here's what the Holm correction does. First, you sort all of your p-values in order, from smallest to largest. For the smallest p-value all you do is multiply it by m, and you're done. However, for all the other ones it's a two-stage process. For instance, when you move to the second smallest p value, you first multiply it by m−1. If this produces a number that is bigger than the adjusted p-value that you got last time, then you keep it. But if it's smaller than the last one, then you copy the last p-value. To illustrate how this works, consider the table below, which shows the calculations of a Holm correction for a collection of five p-values:

raw p	rank j	p×j	Holm p
.001	5	.005	.005
.005	4	.020	.020
.019	3	.057	.057
.022	2	.044	.057
.103	1	.103	.103

Hopefully that makes things clear.

Although it's a little harder to calculate, the Holm correction has some very nice properties: it's more powerful than Bonferroni (i.e., it has a lower Type II error rate), but – counterintuitive as it might seem – it has the *same* Type I error rate. As a consequence, in practice there's never any reason to use the simpler Bonferroni correction, since it is always outperformed by the slightly more elaborate Holm correction. Because of this, the Holm correction is the default one used by `pairwise.t.test()` and `posthocPairwiseT()`. To run the Holm correction in R, you could specify `p.adjust.method = "Holm"` if you wanted to, but since it's the default you can just to do this:

```
posthocPairwiseT( my.anova )
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  mood.gain and drug
##
##          placebo anxifree
## anxifree 0.1502  -
## joyzepam 9.1e-05 0.0011
##
## P value adjustment method: holm
```

As you can see, the biggest p-value (corresponding to the comparison between Anxifree and the placebo) is unaltered: at a value of .15, it is exactly the same as the value we got originally when we applied no correction at all. In contrast, the smallest p-value (Joyzepam versus placebo) has been multiplied by three.

12.6.5 Writing up the post hoc test

Finally, having run the post hoc analysis to determine which groups are significantly different to one another, you might write up the result like this:

Post hoc tests (using the Holm correction to adjust p) indicated that Joyzepam produced a significantly larger mood change than both Anxifree ($p=.001$) and the placebo ($p=9.1\times10^{-5}$). We found no evidence that Anxifree performed better than the placebo ($p=.15$).

Or, if you don't like the idea of reporting exact p-values, then you'd change those numbers to p<.01, p<.001 and p>.05 respectively. Either way, the key thing is that you indicate that you used Holm's correction to adjust the p-values. And of course, I'm assuming that elsewhere in the write up you've included the relevant descriptive statistics (i.e., the group means and standard deviations), since these p-values on their own aren't terribly informative.

This page titled 12.6: Multiple Comparisons and Post Hoc Tests is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **14.6: Multiple Comparisons and Post Hoc Tests** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

12.7: Assumptions of One-way ANOVA

Like any statistical test, analysis of variance relies on some assumptions about the data. There are three key assumptions that you need to be aware of: *normality*, *homogeneity of variance* and *independence*. If you remember back to Section 14.2.4 – which I hope you at least skimmed even if you didn't read the whole thing – I described the statistical models underpinning ANOVA, which I wrote down like this:

$$H_0: Y_{ik} = \mu + \epsilon_{ik}$$

$$H_1: Y_{ik} = \mu_k + \epsilon_{ik}$$

In these equations μ refers to a single, grand population mean which is the same for all groups, and μ_k is the population mean for the k-th group. Up to this point we've been mostly interested in whether our data are best described in terms of a single grand mean (the null hypothesis) or in terms of different group-specific means (the alternative hypothesis). This makes sense, of course: that's actually the important research question! However, all of our testing procedures have – implicitly – relied on a specific assumption about the residuals, ϵ_{ik}, namely that

$$\epsilon_{ik} \sim \text{Normal}(0, \sigma^2)$$

None of the maths works properly without this bit. Or, to be precise, you can still do all the calculations, and you'll end up with an F-statistic, but you have no guarantee that this F-statistic actually measures what you think it's measuring, and so any conclusions that you might draw on the basis of the F test might be wrong.

So, how do we check whether this assumption about the residuals is accurate? Well, as I indicated above, there are three distinct claims buried in this one statement, and we'll consider them separately.

- ***Normality***. The residuals are assumed to be normally distributed. As we saw in Section 13.9, we can assess this by looking at QQ plots or running a Shapiro-Wilk test. I'll talk about this in an ANOVA context in Section 14.9.
- ***Homogeneity of variance***. Notice that we've only got the one value for the population standard deviation (i.e., σ), rather than allowing each group to have it's own value (i.e., σ_k). This is referred to as the homogeneity of variance (sometimes called homoscedasticity) assumption. ANOVA assumes that the population standard deviation is the same for all groups. We'll talk about this extensively in Section 14.7.
- ***Independence***. The independence assumption is a little trickier. What it basically means is that, knowing one residual tells you nothing about any other residual. All of the ϵ_{ik} values are assumed to have been generated without any "regard for" or "relationship to" any of the other ones. There's not an obvious or simple way to test for this, but there are some situations that are clear violations of this: for instance, if you have a repeated-measures design, where each participant in your study appears in more than one condition, then independence doesn't hold; there's a special relationship between some observations… namely those that correspond to the same person! When that happens, you need to use something like repeated measures ANOVA. I don't currently talk about repeated measures ANOVA in this book, but it will be included in later versions.

12.7.1 robust is ANOVA?

One question that people often want to know the answer to is the extent to which you can trust the results of an ANOVA if the assumptions are violated. Or, to use the technical language, how ***robust*** is ANOVA to violations of the assumptions. Due to deadline constraints I don't have the time to discuss this topic. This is a topic I'll cover in some detail in a later version of the book.

This page titled 12.7: Assumptions of One-way ANOVA is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **14.7: Assumptions of One-way ANOVA** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

12.8: Checking the Homogeneity of Variance Assumption

There's more than one way to skin a cat, as the saying goes, and more than one way to test the homogeneity of variance assumption, too (though for some reason no-one made a saying out of that). The most commonly used test for this that I've seen in the literature is the ***Levene test*** (Levene 1960), and the closely related ***Brown-Forsythe test*** (Brown and Forsythe 1974), both of which I'll describe here. Alternatively, you could use the Bartlett test, which is implemented in R via the `bartlett.test()` function, but I'll leave it as an exercise for the reader to go check that one out if you're interested.

Levene's test is shockingly simple. Suppose we have our outcome variable Y_{ik}. All we do is define a new variable, which I'll call Z_{ik}, corresponding to the absolute deviation from the group mean:

$$Z_{ik} = |Y_{ik} - \bar{Y}_k|$$

Okay, what good does this do us? Well, let's take a moment to think about what Z_{ik} actually is, and what we're trying to test. The value of Z_{ik} is a measure of how the i-th observation in the k-th group deviates from its group mean. And our null hypothesis is that all groups have the same variance; that is, the same overall deviations from the group means! So, the null hypothesis in a Levene's test is that the population means of Z are identical for all groups. Hm. So what we need now is a statistical test of the null hypothesis that all group means are identical. Where have we seen that before? Oh right, that's what ANOVA is… and so all that the Levene's test does is run an ANOVA on the new variable Z_{ik}.

What about the Brown-Forsythe test? Does that do anything particularly different? Nope. The only change from the Levene's test is that it constructs the transformed variable Z in a slightly different way, using deviations from the group *medians* rather than deviations from the group *means*. That is, for the Brown-Forsythe test,

$$Z_{ik} = |Y_{ik} - \text{median}_k(Y)|$$

where $\text{median}_k(Y)$ is the median for group k. Regardless of whether you're doing the standard Levene test or the Brown-Forsythe test, the test statistic – which is sometimes denoted F, but sometimes written as W – is calculated in exactly the same way that the F-statistic for the regular ANOVA is calculated, just using a Z_{ik} rather than Y_{ik}. With that in mind, let's just move on and look at how to run the test in R.

12.8.1 Running the Levene's test in R

Okay, so how do we run the Levene test? Obviously, since the Levene test is just an ANOVA, it would be easy enough to manually create the transformed variable Z_{ik} and then use the `aov()` function to run an ANOVA on that. However, that's the tedious way to do it. A better way to do run your Levene's test is to use the `leveneTest()` function, which is in the `car` package. As usual, we first load the package

```
library( car )
```

```
## Loading required package: carData
```

and now that we have, we can run our Levene test. The main argument that you need to specify is `y`, but you can do this in lots of different ways. Probably the simplest way to do it is actually input the original `aov` object. Since I've got the `my.anova` variable stored from my original ANOVA, I can just do this:

```
leveneTest( my.anova )
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  2  1.4672 0.2618
##       15
```

If we look at the output, we see that the test is non-significant ($F_{2,15}$=1.47,p=.26), so it looks like the homogeneity of variance assumption is fine. Remember, although R reports the test statistic as an F-value, it could equally be called W, in which case you'd just write $W_{2,15}$=1.47. Also, note the part of the output that says `center = median`. That's telling you that, by default, the `leveneTest()` function actually does the Brown-Forsythe test. If you want to use the mean instead, then you need to explicitly set the `center` argument, like this:

```
leveneTest( y = my.anova, center = mean )
```

```
## Levene's Test for Homogeneity of Variance (center = mean)
##       Df F value Pr(>F)
## group  2  1.4497 0.2657
##       15
```

That being said, in most cases it's probably best to stick to the default value, since the Brown-Forsythe test is a bit more robust than the original Levene test.

12.8.2 Additional comments

Two more quick comments before I move onto a different topic. Firstly, as mentioned above, there are other ways of calling the `leveneTest()` function. Although the vast majority of situations that call for a Levene test involve checking the assumptions of an ANOVA (in which case you probably have a variable like `my.anova` lying around), sometimes you might find yourself wanting to specify the variables directly. Two different ways that you can do this are shown below:

```
leveneTest(y = mood.gain ~ drug, data = clin.trial)   # y is a formula in this case
leveneTest(y = clin.trial$mood.gain, group = clin.trial$drug)   # y is the outcome
```

Secondly, I did mention that it's possible to run a Levene test just using the `aov()` function. I don't want to waste a lot of space on this, but just in case some readers are interested in seeing how this is done, here's the code that creates the new variables and runs an ANOVA. If you are interested, feel free to run this to verify that it produces the same answers as the Levene test (i.e., with `center = mean`):

```
Y <- clin.trial $ mood.gain    # the original outcome variable, Y
G <- clin.trial $ drug         # the grouping variable, G
gp.mean <- tapply(Y, G, mean)  # calculate group means
Ybar <- gp.mean[G]             # group mean associated with each obs
Z <- abs(Y - Ybar)             # the transformed variable, Z
summary( aov(Z ~ G) )          # run the ANOVA
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## G            2 0.0616 0.03080    1.45  0.266
## Residuals   15 0.3187 0.02125
```

That said, I don't imagine that many people will care about this. Nevertheless, it's nice to know that you could do it this way if you wanted to. And for those of you who do try it, I think it helps to demystify the test a little bit when you can see – with your own eyes – the way in which Levene's test relates to ANOVA.

This page titled 12.8: Checking the Homogeneity of Variance Assumption is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **14.8: Checking the Homogeneity of Variance Assumption** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

12.9: Removing the Homogeneity of Variance Assumption

In our example, the homogeneity of variance assumption turned out to be a pretty safe one: the Levene test came back non-significant, so we probably don't need to worry. However, in real life we aren't always that lucky. How do we save our ANOVA when the homogeneity of variance assumption is violated? If you recall from our discussion of t-tests, we've seen this problem before. The Student t-test assumes equal variances, so the solution was to use the Welch t-test, which does not. In fact, Welch (1951) also showed how we can solve this problem for ANOVA too (the ***Welch one-way test***). It's implemented in R using the `oneway.test()` function. The arguments that we'll need for our example are:

- `formula` . This is the model formula, which (as usual) needs to specify the outcome variable on the left hand side and the grouping variable on the right hand side: i.e., something like `outcome ~ group` .
- `data` . Specifies the data frame containing the variables.
- `var.equal` . If this is `FALSE` (the default) a Welch one-way test is run. If it is `TRUE` then it just runs a regular ANOVA.

The function also has a `subset` argument that lets you analyse only some of the observations and a `na.action` argument that tells it how to handle missing data, but these aren't necessary for our purposes. So, to run the Welch one-way ANOVA for our example, we would do this:

```
oneway.test(mood.gain ~ drug, data = clin.trial)
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  mood.gain and drug
## F = 26.322, num df = 2.0000, denom df = 9.4932, p-value = 0.000134
```

To understand what's happening here, let's compare these numbers to what we got earlier in Section 14.3 when we ran our original ANOVA. To save you the trouble of flicking back, here are those numbers again, this time calculated by setting `var.equal = TRUE` for the `oneway.test()` function:

```
oneway.test(mood.gain ~ drug, data = clin.trial, var.equal = TRUE)
```

```
##
##  One-way analysis of means
##
## data:  mood.gain and drug
## F = 18.611, num df = 2, denom df = 15, p-value = 8.646e-05
```

Okay, so originally our ANOVA gave us the result F(2,15)=18.6, whereas the Welch one-way test gave us F(2,9.49)=26.32. In other words, the Welch test has reduced the within-groups degrees of freedom from 15 to 9.49, and the F-value has increased from 18.6 to 26.32.

This page titled 12.9: Removing the Homogeneity of Variance Assumption is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **14.9: Removing the Homogeneity of Variance Assumption** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

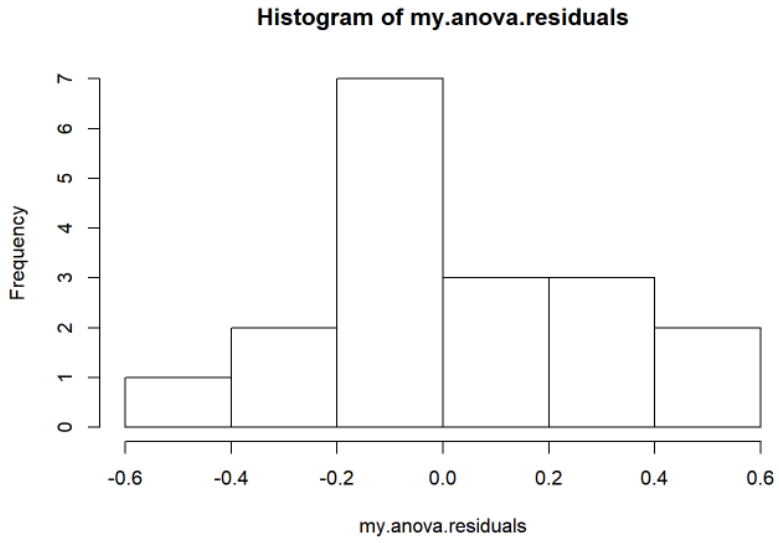
12.10: Checking the Normality Assumption

Testing the normality assumption is relatively straightforward. We covered most of what you need to know in Section 13.9. The only thing we really need to know how to do is pull out the residuals (i.e., the ϵ_{ik} values) so that we can draw our QQ plot and run our Shapiro-Wilk test. First, let's extract the residuals. R provides a function called `residuals()` that will do this for us. If we pass our `my.anova` to this function, it will return the residuals. So let's do that:

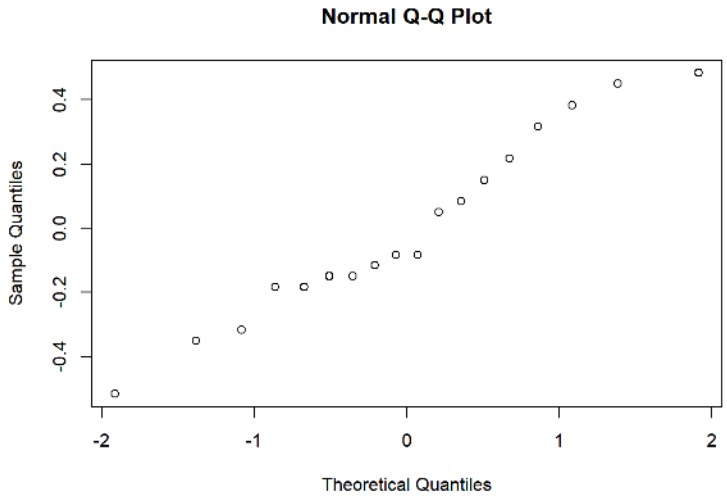
```
my.anova.residuals <- residuals( object = my.anova )   # extract the residuals
```

We can print them out too, though it's not exactly an edifying experience. In fact, given that I'm on the verge of putting *myself* to sleep just typing this, it might be a good idea to skip that step. Instead, let's draw some pictures and run ourselves a hypothesis test:

```
hist( x = my.anova.residuals )          # plot a histogram (similar to Figure @ref{f
```




```
qqnorm( y = my.anova.residuals )        # draw a QQ plot (similar to Figure @ref{fig
```




```
shapiro.test( x = my.anova.residuals )   # run Shapiro-Wilk test
```

```
##
##  Shapiro-Wilk normality test
##
## data:  my.anova.residuals
## W = 0.96019, p-value = 0.6053
```

The histogram and QQ plot are both look pretty normal to me.[212] This is supported by the results of our Shapiro-Wilk test (W=.96, p=.61) which finds no indication that normality is violated.

This page titled 12.10: Checking the Normality Assumption is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **14.10: Checking the Normality Assumption** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

12.11: Removing the Normality Assumption

Now that we've seen how to check for normality, we are led naturally to ask what we can do to address violations of normality. In the context of a one-way ANOVA, the easiest solution is probably to switch to a non-parametric test (i.e., one that doesn't rely on any particular assumption about the kind of distribution involved). We've seen non-parametric tests before, in Chapter 13: when you only have two groups, the Wilcoxon test provides the non-parametric alternative that you need. When you've got three or more groups, you can use the ***Kruskal-Wallis rank sum test*** (Kruskal and Wallis 1952). So that's the test we'll talk about next.

12.11.1 logic behind the Kruskal-Wallis test

The Kruskal-Wallis test is surprisingly similar to ANOVA, in some ways. In ANOVA, we started with Y_{ik}, the value of the outcome variable for the ith person in the kth group. For the Kruskal-Wallis test, what we'll do is rank order all of these Y_{ik} values, and conduct our analysis on the ranked data. So let's let R_{ik} refer to the ranking given to the ith member of the kth group. Now, let's calculate $\bar{R}_k$, the average rank given to observations in the kth group:

$$\bar{R}_k = \frac{1}{N_K} \sum_i R_{ik}$$

and let's also calculate $\bar{R}$, the grand mean rank:

$$\bar{R} = \frac{1}{N} \sum_i \sum_k R_{ik}$$

Now that we've done this, we can calculate the squared deviations from the grand mean rank $\bar{R}$. When we do this for the individual scores – i.e., if we calculate $(R_{ik} - \bar{R})^2$ – what we have is a "nonparametric" measure of how far the ik-th observation deviates from the grand mean rank. When we calculate the squared deviation of the group means from the grand means – i.e., if we calculate $(\bar{R}_k - \bar{R})^2$ – then what we have is a nonparametric measure of how much the *group* deviates from the grand mean rank. With this in mind, let's follow the same logic that we did with ANOVA, and define our *ranked* sums of squares measures in much the same way that we did earlier. First, we have our "total ranked sums of squares":

$$\text{RSS}_{tot} = \sum_k \sum_i (R_{ik} - R)^2$$

and we can define the "between groups ranked sums of squares" like this:

$$\begin{aligned} \text{RSS}_b &= \sum_k \sum_i \left(\bar{R}_k - \bar{R}\right)^2 \\ &= \sum_k N_k \left(\bar{R}_k - \bar{R}\right)^2 \end{aligned}$$

So, if the null hypothesis is true and there are no true group differences at all, you'd expect the between group rank sums RSS_b to be very small, much smaller than the total rank sums RSS_{tot}. Qualitatively this is very much the same as what we found when we went about constructing the ANOVA F-statistic; but for technical reasons the Kruskal-Wallis test statistic, usually denoted K, is constructed in a slightly different way:

$$K = (N-1) \; x \; \frac{RSS_b}{RSS_{tot}}$$

and, if the null hypothesis is true, then the sampling distribution of K is *approximately* chi-square with G−1 degrees of freedom (where G is the number of groups). The larger the value of K, the less consistent the data are with null hypothesis, so this is a one-sided test: we reject H_0 when K is sufficiently large.

12.11.2 Additional details

The description in the previous section illustrates the logic behind the Kruskal-Wallis test. At a conceptual level, this is the right way to think about how the test works. However, from a purely mathematical perspective it's needlessly complicated. I won't show you the derivation, but you can use a bit of algebraic jiggery-pokery[213] to show that the equation for K can be rewritten as

$$K = \frac{12}{N(N-1)} \sum_k N_k \bar{R}_k^2 - 3(N+1)$$

It's this last equation that you sometimes see given for K. This is way easier to calculate than the version I described in the previous section, it's just that it's totally meaningless to actual humans. It's probably best to think of K the way I described it earlier… as an

analogue of ANOVA based on ranks. But keep in mind that the test statistic that gets calculated ends up with a rather different look to it than the one we used for our original ANOVA.

But wait, there's more! Dear lord, why is there always *more*? The story I've told so far is only actually true when there are no ties in the raw data. That is, if there are no two observations that have exactly the same value. If there *are* ties, then we have to introduce a correction factor to these calculations. At this point I'm assuming that even the most diligent reader has stopped caring (or at least formed the opinion that the tie-correction factor is something that doesn't require their immediate attention). So I'll very quickly tell you how it's calculated, and omit the tedious details about *why* it's done this way. Suppose we construct a frequency table for the raw data, and let f_j be the number of observations that have the j-th unique value. This might sound a bit abstract, so here's the R code showing a concrete example:

```
f <- table( clin.trial$mood.gain )   # frequency table for mood gain
print(f)   # we have some ties
```

```
##
## 0.1 0.2 0.3 0.4 0.5 0.6 0.8 0.9 1.1 1.2 1.3 1.4 1.7 1.8
##   1   1   2   1   1   2   1   1   1   1   2   2   1   1
```

Looking at this table, notice that the third entry in the frequency table has a value of 2. Since this corresponds to a `mood.gain` of 0.3, this table is telling us that two people's mood increased by 0.3. More to the point, note that we can say that `f[3]` has a value of `2` . Or, in the mathematical notation I introduced above, this is telling us that f_3=2. Yay. So, now that we know this, the tie correction factor (TCF) is:

$$\text{TCF} = 1 - \frac{\sum_j f_j^3 - f_j}{N^3 - N}$$

The tie-corrected value of the Kruskal-Wallis statistic obtained by dividing the value of K by this quantity: it is this tie-corrected version that R calculates. And at long last, we're actually finished with the theory of the Kruskal-Wallis test. I'm sure you're all terribly relieved that I've cured you of the existential anxiety that naturally arises when you realise that you *don't* know how to calculate the tie-correction factor for the Kruskal-Wallis test. Right?

12.11.3 run the Kruskal-Wallis test in R

Despite the horror that we've gone through in trying to understand what the Kruskal-Wallis test actually does, it turns out that running the test is pretty painless, since R has a function called `kruskal.test()` . The function is pretty flexible, and allows you to input your data in a few different ways. Most of the time you'll have data like the `clin.trial` data set, in which you have your outcome variable `mood.gain` , and a grouping variable `drug` . If so, you can call the `kruskal.test()` function by specifying a formula, and a data frame:

```
kruskal.test(mood.gain ~ drug, data = clin.trial)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  mood.gain by drug
## Kruskal-Wallis chi-squared = 12.076, df = 2, p-value = 0.002386
```

A second way of using the `kruskal.test()` function, which you probably won't have much reason to use, is to directly specify the outcome variable and the grouping variable as separate input arguments, `x` and `g` :

```
kruskal.test(x = clin.trial$mood.gain, g = clin.trial$drug)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  clin.trial$mood.gain and clin.trial$drug
## Kruskal-Wallis chi-squared = 12.076, df = 2, p-value = 0.002386
```

This isn't very interesting, since it's just plain easier to specify a formula. However, sometimes it can be useful to specify `x` as a list. What I mean is this. Suppose you actually had data as three separate variables, `placebo`, `anxifree` and `joyzepam`. If that's the format that your data are in, then it's convenient to know that you can bundle all three together as a list:

```
mood.gain <- list( placebo, joyzepam, anxifree )
kruskal.test( x = mood.gain )
```

And again, this would give you exactly the same results as the command we tried originally.

This page titled 12.11: Removing the Normality Assumption is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **14.11: Removing the Normality Assumption** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

12.12: On the Relationship Between ANOVA and the Student t Test

There's one last thing I want to point out before finishing. It's something that a lot of people find kind of surprising, but it's worth knowing about: an ANOVA with two groups is identical to the Student t-test. No, really. It's not just that they are similar, but they are actually equivalent in every meaningful way. I won't try to prove that this is always true, but I will show you a single concrete demonstration. Suppose that, instead of running an ANOVA on our `mood.gain ~ drug` model, let's instead do it using `therapy` as the predictor. If we run this ANOVA, here's what we get:

```
summary( aov( mood.gain ~ therapy, data = clin.trial ))
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## therapy      1  0.467  0.4672   1.708   0.21
## Residuals   16  4.378  0.2736
```

Overall, it looks like there's no significant effect here at all but, as we'll see in Chapter @ref(anova2 this is actually a misleading answer! In any case, it's irrelevant to our current goals: our interest here is in the F-statistic, which is F(1,16)=1.71, and the p-value, which is .21. Since we only have two groups, I didn't actually need to resort to an ANOVA, I could have just decided to run a Student t-test. So let's see what happens when I do that:

```
t.test( mood.gain ~ therapy, data = clin.trial, var.equal = TRUE )
```

```
##
##  Two Sample t-test
##
## data:  mood.gain by therapy
## t = -1.3068, df = 16, p-value = 0.2098
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.8449518  0.2005073
## sample estimates:
## mean in group no.therapy        mean in group CBT
##                0.7222222                1.0444444
```

Curiously, the p-values are identical: once again we obtain a value of p=.21. But what about the test statistic? Having run a t-test instead of an ANOVA, we get a somewhat different answer, namely t(16)=−1.3068. However, there is a fairly straightforward relationship here. If we square the t-statistic

```
1.3068 ^ 2
```

```
## [1] 1.707726
```

we get the F-statistic from before.

This page titled 12.12: On the Relationship Between ANOVA and the Student t Test is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **14.12: On the Relationship Between ANOVA and the Student t Test** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

CHAPTER OVERVIEW

13: Introduction to Linear Regression

Linear regression is a very powerful statistical technique. Many people have some familiarity with regression just from reading the news, where graphs with straight lines are overlaid on scatterplots. Linear models can be used for prediction or to evaluate whether there is a linear relationship between two numerical variables.

13.1: Prelude to Linear Regression

13.2: Line Fitting, Residuals, and Correlation

13.3: Fitting a Line by Least Squares Regression

13.4: Types of Outliers in Linear Regression

13.5: Inference for Linear Regression

13.6: Exercises

This page titled 13: Introduction to Linear Regression is shared under a CC BY-SA 3.0 license and was authored, remixed, and/or curated by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel via source content that was edited to the style and standards of the LibreTexts platform.

13.1: Prelude to Linear Regression

Linear regression is a very powerful statistical technique. Many people have some familiarity with regression just from reading the news, where graphs with straight lines are overlaid on scatterplots. Linear models can be used for prediction or to evaluate whether there is a linear relationship between two numerical variables.

Figure 13.1.1 shows two variables whose relationship can be modeled perfectly with a straight line. The equation for the line is

$$y = 5 + 57.49x \tag{13.1.1}$$

Imagine what a perfect linear relationship would mean: you would know the exact value of y just by knowing the value of x. This is unrealistic in almost any natural process. For example, if we took family income x, this value would provide some useful information about how much financial support y a college may offer a prospective student. However, there would still be variability in financial support, even when comparing students whose families have similar financial backgrounds.

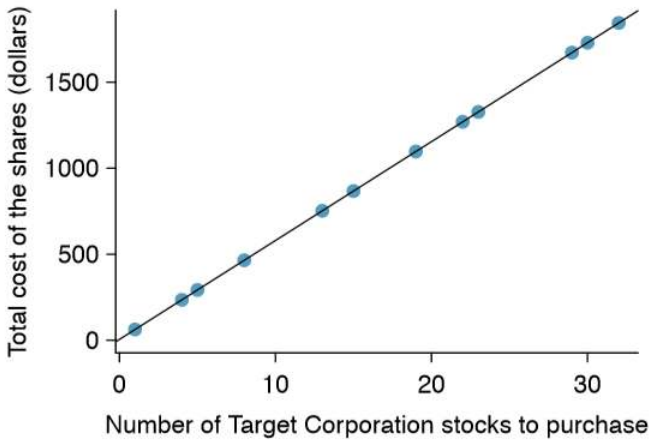


Figure 13.1.1: Requests from twelve separate buyers were simultaneously placed with a trading company to purchase Target Corporation stock (ticker TGT, April 26th, 2012), and the total cost of the shares were reported. Because the cost is computed using a linear formula, the linear t is perfect.

Linear regression assumes that the relationship between two variables, x and y, can be modeled by a straight line:

$$y = \beta_0 + \beta_1 x \tag{13.1.2}$$

where β_0 and β_1 represent two model parameters (β is the Greek letter beta). These parameters are estimated using data, and we write their point estimates as β_0 and β_1. When we use x to predict y, we usually call x the *explanatory or predictor variable*, and we call y the *response*.

It is rare for all of the data to fall on a straight line, as seen in the three scatterplots in Figure 13.1.2. In each case, the data fall around a straight line, even if none of the observations fall exactly on the line. The first plot shows a relatively strong downward linear trend, where the remaining variability in the data around the line is minor relative to the strength of the relationship between x and y. The second plot shows an upward trend that, while evident, is not as strong as the first. The last plot shows a very weak downward trend in the data, so slight we can hardly notice it. In each of these examples, we will have some uncertainty regarding our estimates of the model parameters, β_0 and β_1. For instance, we might wonder, should we move the line up or down a little, or should we tilt it more or less?

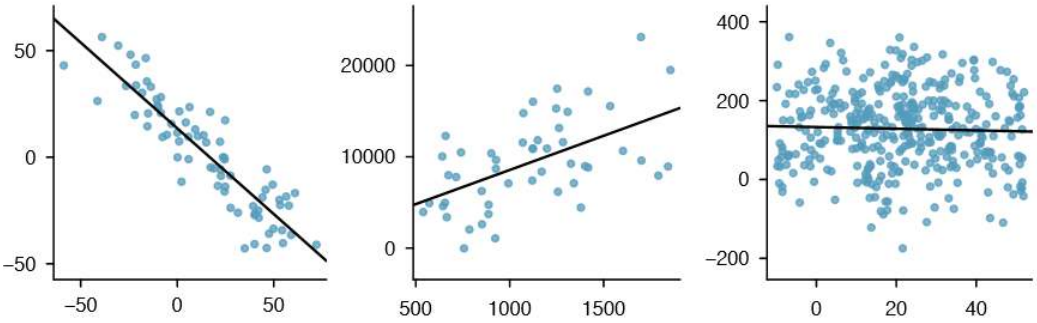


Figure **13.1.1**: Three data sets where a linear model may be useful even though the data do not all fall exactly on the line.

As we move forward in this chapter, we will learn different criteria for line-fitting, and we will also learn about the uncertainty associated with estimates of model parameters.

This page titled 13.1: Prelude to Linear Regression is shared under a CC BY-SA 3.0 license and was authored, remixed, and/or curated by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel via source content that was edited to the style and standards of the LibreTexts platform.

- **7.1: Prelude to Linear Regression** by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel is licensed CC BY-SA 3.0. Original source: https://www.openintro.org/book/os.

13.2: Line Fitting, Residuals, and Correlation

We will also see examples in this chapter where fitting a straight line to the data, even if there is a clear relationship between the variables, is not helpful. One such case is shown in Figure **13.2.1** where there is a very strong relationship between the variables even though the trend is not linear. We will discuss nonlinear trends in this chapter and the next, but the details of fitting nonlinear models discussed elsewhere. In this section, we examine criteria for identifying a linear model and introduce a new statistic, correlation.

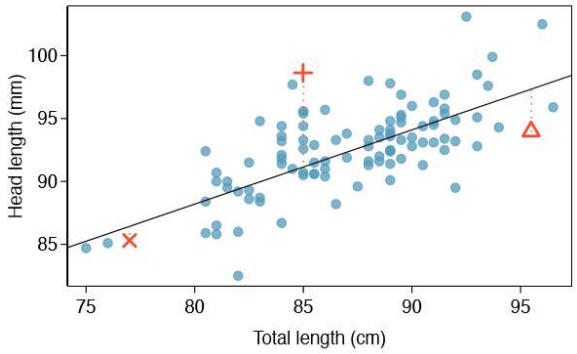


Figure **13.2.1**: A linear model is not useful in this nonlinear case. These data are from an introductory physics experiment.

Beginning with Straight Lines

Scatterplots were introduced in Chapter 1 as a graphical technique to present two numerical variables simultaneously. Such plots permit the relationship between the variables to be examined with ease. Figure **13.2.2** shows a scatterplot for the head length and total length of 104 brushtail possums from Australia. Each point represents a single possum from the data.

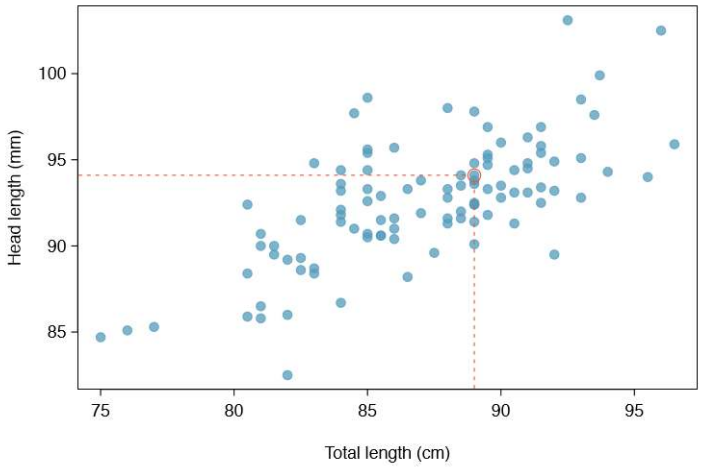


Figure **13.2.2**: A scatterplot showing head length against total length for 104 brushtail possums. A point representing a possum with head length 94.1mm and total length 89 cm is highlighted.

The head and total length variables are associated. Possums with an above average total length also tend to have above average head lengths. While the relationship is not perfectly linear, it could be helpful to partially explain the connection between these variables with a straight line.



Figure 13.2.3: The common brushtail possum of Australia. *Photo by wollombi on Flickr: www.ickr.com/photos/wollombi/58499575*

Straight lines should only be used when the data appear to have a linear relationship, such as the case shown in the left panel of Figure 13.2.4. The right panel of Figure 13.2.4 shows a case where a curved line would be more useful in understanding the relationship between the two variables.

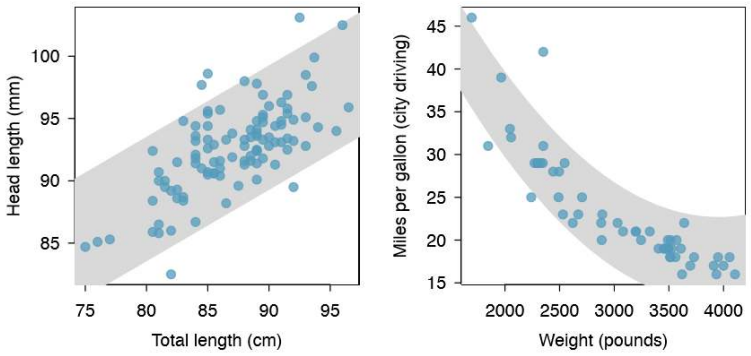


Figure 13.2.4: The figure on the left shows head length versus total length, and reveals that many of the points could be captured by a straight band. On the right, we see that a curved band is more appropriate in the scatterplot for weight and mpgCity from the cars data set.

Caution: Watch out for curved trends

We only consider models based on straight lines in this chapter. If data show a nonlinear trend, like that in the right panel of Figure 13.2.4, more advanced techniques should be used.

Fitting a line "By Eye"

We want to describe the relationship between the head length and total length variables in the possum data set using a line. In this example, we will use the total length as the predictor variable, x, to predict a possum's head length, y. We could fit the linear relationship by eye, as in Figure 13.2.5. The equation for this line is

$$\hat{y} = 41 + 0.59x \tag{7.2}$$

We can use this line to discuss properties of possums. For instance, the equation predicts a possum with a total length of 80 cm will have a head length of

$$\hat{y} = 41 + 0.59 \times 80 \tag{13.2.1}$$

$$= 88.2 \tag{13.2.2}$$

A "hat" on y is used to signify that this is an estimate. This estimate may be viewed as an average: the equation predicts that possums with a total length of 80 cm will have an average head length of 88.2 mm. Absent further information about an 80 cm possum, the prediction for head length that uses the average is a reasonable estimate.

Residuals

Residuals are the leftover variation in the data after accounting for the model fit:

$$\text{Data} = \text{Fit} + \text{Residual} \tag{13.2.3}$$

Each observation will have a residual. If an observation is above the regression line, then its residual, the vertical distance from the observation to the line, is positive. Observations below the line have negative residuals. One goal in picking the right linear model is for these residuals to be as small as possible.

Three observations are noted specially in Figure 13.2.5. The observation marked by an "X" has a small, negative residual of about -1; the observation marked by "+" has a large residual of about +7; and the observation marked by Δ has a moderate residual of about -4. The size of a residual is usually discussed in terms of its absolute value. For example, the residual for Δ is larger than that of "X" because | - 4| is larger than | - 1|.

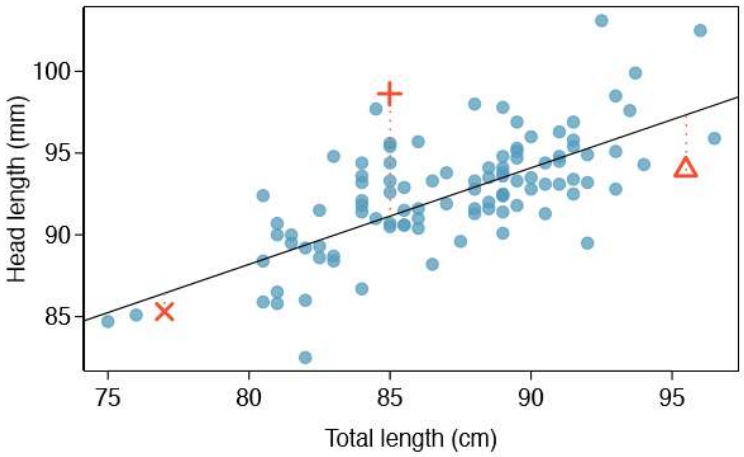


Figure 13.2.5: A reasonable linear model was to represent the relationship between head length and total length.

Residual: difference between observed and expected

The residual of the fifth observation (x_i, y_i) is the difference of the observed response (y_i) and the response we would predict based on the model fit $(\hat{y}_i)$:

$$e_i = y_i - \hat{y}_i \tag{13.2.4}$$

We typically identify $\hat{y}_i$ by plugging xi into the model.

Example 13.2.1

The linear fit shown in Figure 13.2.5 is given as $\hat{y} = 41 + 0.59x$. Based on this line, formally compute the residual of the observation (77.0, 85.3). This observation is denoted by "X" on the plot. Check it against the earlier visual estimate, -1.

Solution

We first compute the predicted value of point "X" based on the model:

$$\hat{y} = 41 + 0.59x_x = 41 + 0.59 \times 77.0 = 86.4 \tag{13.2.5}$$

Next we compute the difference of the actual head length and the predicted head length:

$$e_x = y_x - \hat{y}_x = 85.3 - 86.4 = -1.1 \tag{13.2.6}$$

This is very close to the visual estimate of -1.

Exercise 13.2.1A

If a model underestimates an observation, will the residual be positive or negative? What about if it overestimates the observation?

Answer

If a model underestimates an observation, then the model estimate is below the actual. The residual, which is the actual observation value minus the model estimate, must then be positive. The opposite is true when the model overestimates the observation: the residual is negative.

Exercise 13.2.1B

Compute the residuals for the observations (85.0, 98.6) ("+" in Figure 13.2.5) and (95.5, 94.0) ("Δ") using the linear relationship

$$\hat{y} = 41 + 0.59x. \tag{13.2.7}$$

Answer

(+) First compute the predicted value based on the model:

$$\hat{y}_+ = 41 + 0.59x_+ = 41 + 0.59 \times 85.0 = 91.15 \tag{13.2.8}$$

Then the residual is given by

$$e_+ = y_+ - \hat{y}_+ = 98.6 - 91.15 = 7.45 \tag{13.2.9}$$

This was close to the earlier estimate of 7.

$(\Delta)\hat{y}_\Delta = 41 + 0.59x_\Delta = 97.3 . e_\Delta = y_\Delta - \hat{y}_\Delta = -3.3$, close to the estimate of -4.

Residuals are helpful in evaluating how well a linear model fits a data set. We often display them in a **residual plot** such as the one shown in Figure 13.2.6 for the regression line in Figure 13.2.5. The residuals are plotted at their original horizontal locations but with the vertical coordinate as the residual. For instance, the point $(85.0, 98.6)_+$ had a residual of 7.45, so in the residual plot it is placed at (85.0, 7.45). Creating a residual plot is sort of like tipping the scatterplot over so the regression line is horizontal.

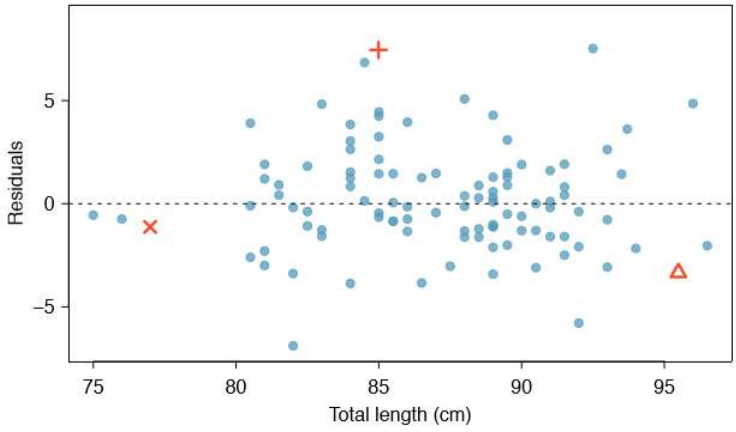


Figure 13.2.5.

Example 13.2.1

One purpose of residual plots is to identify characteristics or patterns still apparent in data after fitting a model. Figure 13.2.7 shows three scatterplots with linear models in the first row and residual plots in the second row. Can you identify any patterns remaining in the residuals?

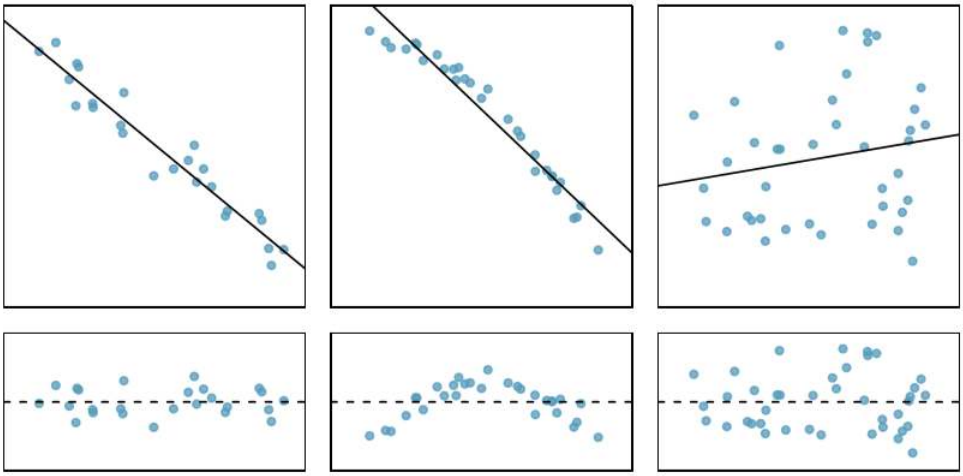


Figure **13.2.7**: Sample data with their best fitting lines (top row) and their corresponding residual plots (bottom row).

Solution

In the first data set (first column), the residuals show no obvious patterns. The residuals appear to be scattered randomly around the dashed line that represents 0.

The second data set shows a pattern in the residuals. There is some curvature in the scatterplot, which is more obvious in the residual plot. We should not use a straight line to model these data. Instead, a more advanced technique should be used.

The last plot shows very little upwards trend, and the residuals also show no obvious patterns. It is reasonable to try to fit a linear model to the data. However, it is unclear whether there is statistically significant evidence that the slope parameter is different from zero. The point estimate of the slope parameter, labeled b_1, is not zero, but we might wonder if this could just be due to chance. We will address this sort of scenario in Section 7.4.

Describing Linear Relationships with Correlation

We can compute the correlation using a formula, just as we did with the sample mean and standard deviation. However, this formula is rather complex, so we generally perform the calculations on a computer or calculator. Figure **13.2.8** shows eight plots and their corresponding correlations. Only when the relationship is perfectly linear is the correlation either -1 or 1. If the relationship is strong and positive, the correlation will be near +1. If it is strong and negative, it will be near -1. If there is no apparent linear relationship between the variables, then the correlation will be near zero.

Formally, we can compute the correlation for observations $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ using the formula

$$R = \frac{1}{n-1} \sum_{i=1}^{n} \frac{x_i - \bar{x}}{s_x} \frac{y_i - \bar{y}}{s_y} \tag{13.2.10}$$

where $\bar{x}, \bar{y}, s_x$, and s_y are the sample means and standard deviations for each variable.

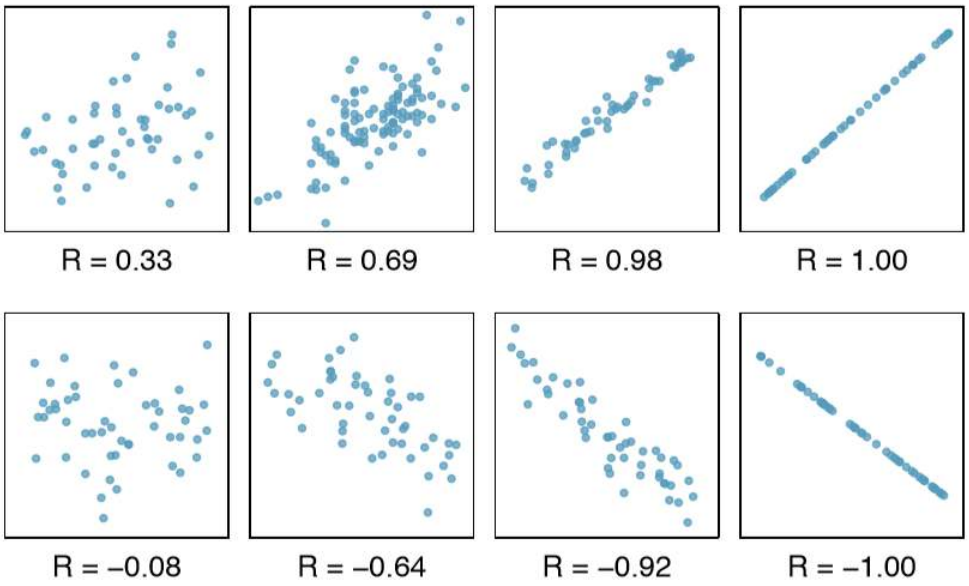


Figure **13.2.8**: Sample scatterplots and their correlations. The first row shows variables with a positive relationship, represented by the trend up and to the right. The second row shows variables with a negative trend, where a large value in one variable is associated with a low value in the other.

Correlation: strength of a linear relationship

Correlation, which always takes values between -1 and 1, describes the strength of the linear relationship between two variables. We denote the correlation by R.

The correlation is intended to quantify the strength of a linear trend. Nonlinear trends, even when strong, sometimes produce correlations that do not reflect the strength of the relationship; see three such examples in Figure **13.2.9**.

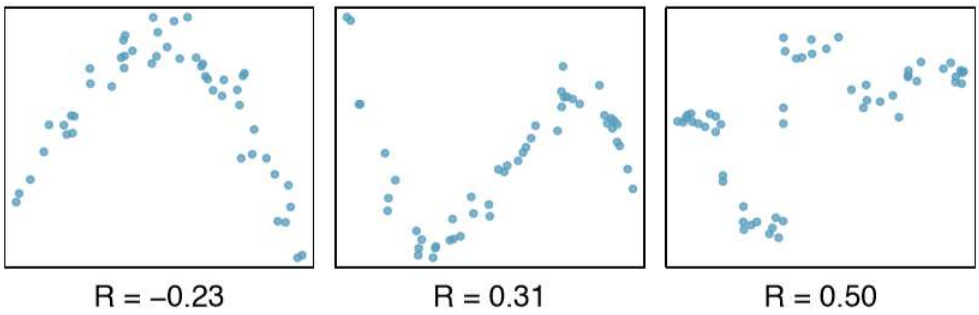


Figure **13.2.9**: Sample scatterplots and their correlations. In each case, there is a strong relationship between the variables. However, the correlation is not very strong, and the relationship is not linear.

Exercise 13.2.1

It appears no straight line would fit any of the datasets represented in Figure **13.2.9**. Try drawing nonlinear curves on each plot. Once you create a curve for each, describe what is important in your fit.[4]

Answer

We'll leave it to you to draw the lines. In general, the lines you draw should be close to most points and reflect overall trends in the data.

This page titled 13.2: Line Fitting, Residuals, and Correlation is shared under a CC BY-SA 3.0 license and was authored, remixed, and/or curated by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel via source content that was edited to the style and standards of the LibreTexts

platform.

- **7.2: Line Fitting, Residuals, and Correlation** by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel is licensed CC BY-SA 3.0. Original source: https://www.openintro.org/book/os.

13.3: Fitting a Line by Least Squares Regression

Fitting linear models by eye is open to criticism since it is based on an individual preference. In this section, we use least squares regression as a more rigorous approach.

This section considers family income and gift aid data from a random sample of fifty students in the 2011 freshman class of Elmhurst College in Illinois. Gift aid is financial aid that is a gift, as opposed to a loan. A scatterplot of the data is shown in Figure **13.3.1** along with two linear fits. The lines follow a negative trend in the data; students who have higher family incomes tended to have lower gift aid from the university.

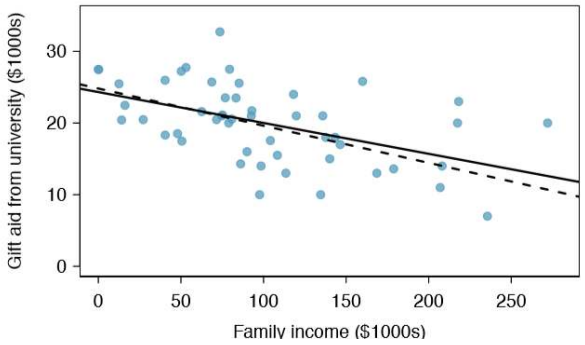


Figure **13.3.1**: Gift aid and family income for a random sample of 50 freshman students from Elmhufirst College. Two lines are fit to the data, the solid line being the least squares line. These data were sampled from a table of data for all freshman from the 2011 class at Elmhurst College that accompanied an article titled What Students Really Pay to Go to College published online by The Chronicle of Higher Education: chronicle.com/article/What-Students-Really-Pay-to-Go/131435

Exercise 13.3.1

Is the correlation positive or negative in Figure **13.3.1**?[6]

Solution

[6]*Larger family incomes are associated with lower amounts of aid, so the correlation will be negative. Using a computer, the correlation can be computed: -0.499.*

An Objective Measure for Finding the Best Line

We begin by thinking about what we mean by "best". Mathematically, we want a line that has small residuals. Perhaps our criterion could minimize the sum of the residual magnitudes:

$$|e_1| + |e_2| + \cdots + |e_n| \tag{13.3.1}$$

which we could accomplish with a computer program. The resulting dashed line shown in Figure **13.3.1** demonstrates this fit can be quite reasonable. However, a more common practice is to choose the line that minimizes the sum of the *squared* residuals:

$$e_1^2 + e_2^2 + \cdots + e_n^2 \tag{13.3.2}$$

The line that minimizes this *least squares criterion* is represented as the solid line in Figure **13.3.1**. This is commonly called the *least squares line*. The following are three possible reasons to choose Criterion **13.3.2** over Criterion **13.3.1**:

1. It is the most commonly used method.
2. Computing the line based on Criterion **13.3.2** is much easier by hand and in most statistical software.
3. In many applications, a residual twice as large as another residual is more than twice as bad. For example, being off by 4 is usually more than twice as bad as being off by squaring the residuals accounts for this discrepancy.

The first two reasons are largely for tradition and convenience; the last reason explains why Criterion **13.3.2** is typically most helpful.

There are applications where Criterion 13.3.1 may be more useful, and there are plenty of other criteria we might consider. However, this book only applies the least squares criterion.

Conditions for the Least Squares Line

When fitting a least squares line, we generally require

- **Linearity**. The data should show a linear trend. If there is a nonlinear trend (e.g. left panel of Figure 13.3.2), an advanced regression method from another book or later course should be applied.
- **Nearly normal residuals**. Generally the residuals must be nearly normal. When this condition is found to be unreasonable, it is usually because of outliers or concerns about influential points, which we will discuss in greater depth in Section 7.3. An example of non-normal residuals is shown in the second panel of Figure 13.3.2.
- **Constant variability**. The variability of points around the least squares line remains roughly constant. An example of non-constant variability is shown in the third panel of Figure 13.3.2.

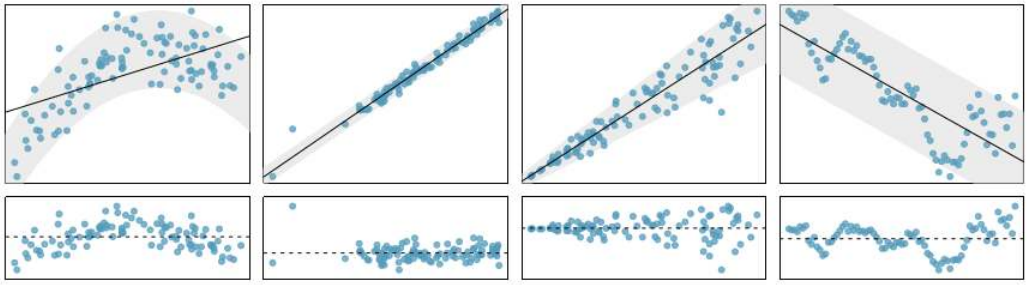


Figure 13.3.2: Four examples showing when the methods in this chapter are insufficient to apply to the data. In the left panel, a straight line does not t the data. In the second panel, there are outliers; two points on the left are relatively distant from the rest of the data, and one of these points is very far away from the line. In the third panel, the variability of the data around the line increases with larger values of x. In the last panel, a time series data set is shown, where successive observations are highly correlated.

Be cautious about applying regression to data collected sequentially in what is called a time series. Such data may have an underlying structure that should be considered in a model and analysis. There are other instances where correlations within the data are important. This topic will be further discussed in Chapter 8.

Exercise 13.3.2

Should we have concerns about applying least squares regression to the Elmhurst data in Figure 13.3.1?

Solution

The trend appears to be linear, the data fall around the line with no obvious outliers, the variance is roughly constant. These are also not time series observations. Least squares regression can be applied to these data.

Finding the Least Squares Line

For the Elmhurst data, we could write the equation of the least squares regression line as

$$\hat{aid} = \beta_0 + \beta_1 \times \text{family income} \tag{13.3.3}$$

Here the equation is set up to predict gift aid based on a student's family income, which would be useful to students considering Elmhurst. These two values, β_0 and β_1, are the parameters of the regression line.

As in Chapters 4-6, the parameters are estimated using observed data. In practice, this estimation is done using a computer in the same way that other estimates, like a sample mean, can be estimated using a computer or calculator. However, we can also find the parameter estimates by applying two properties of the least squares line:

• The slope of the least squares line can be estimated by

$$b_1 = \frac{s_y}{s_x} R \tag{13.3.4}$$

where R is the correlation between the two variables, and s_x and s_y are the sample standard deviations of the explanatory variable and response, respectively.

• If $\bar{x}$ is the mean of the horizontal variable (from the data) and $\bar{y}$ is the mean of the vertical variable, then the point $(\bar{x}, \bar{y})$ is on the least squares line.

We use b_0 and b_1 to represent the point estimates of the parameters β_0 and β_1.

Exercise 13.3.3

Table 7.14 shows the sample means for the family income and gift aid as \$101,800 and \$19,940, respectively. Plot the point (101.8, 19.94) on Figure **13.3.1** on page 324 to verify it falls on the least squares line (the solid line).[9]

Table 7.14: Summary statistics for family income and gift aid.

	family income, in \$1000s ("x")	gift aid, in \$1000s ("y")
mean sd	$\bar{x}$ = 101.8 s_x = 63.2	$\bar{y}$ = 19.94 s_y = 5.46
		R = -0.499

[9] *If you need help finding this location, draw a straight line up from the x-value of 100 (or thereabout). Then draw a horizontal line at 20 (or thereabout). These lines should intersect on the least squares line.*

Exercise 13.3.4

sing the summary statistics in Table 7.14, compute the slope for the regression line of gift aid against family income.

Hint:

Apply Equation 13.3.4 with the summary statistics from Table 7.14 to compute the slope:

$$b_1 = \frac{s_y}{s_x} R = \frac{5.46}{63.2}(-0.499) = -0.0431 \tag{13.3.5}$$

You might recall the point-slope form of a line from math class (another common form is slope-intercept). Given the slope of a line and a point on the line, (x_0, y_0), the equation for the line can be written as

$$y - y_0 = \text{slope} \times (x - x_0) \tag{13.3.6}$$

A common exercise to become more familiar with foundations of least squares regression is to use basic summary statistics and point-slope form to produce the least squares line.

TIP: Identifying the least squares line from summary statistics

To identify the least squares line from summary statistics:

- Estimate the slope parameter, b_1, using Equation 13.3.4.
- Noting that the point $(\bar{x}, \bar{y})$ is on the least squares line, use $x_0 = \bar{x}$ and $y_0 = \bar{y}$ along with the slope b_1 in the point-slope equation:

$$y - \bar{y} = b_1(x - \bar{x}) \tag{13.3.7}$$

- Simplify the equation.

Example 13.3.1

Using the point (101.8, 19.94) from the sample means and the slope estimate $b_1 = -0.0431$ from Exercise 7.14, and the least-squares line for predicting aid based on family income.

Solution

Apply the point-slope equation using (101.8, 19.94) and the slope $b_1 = -0.0431$:

$$y - y_0 = b_1(x - x_0) \tag{13.3.8}$$

$$y - 19.94 = -0.0431(x - 101.8) \tag{13.3.9}$$

Expanding the right side and then adding 19.94 to each side, the equation simplifies:

$$\hat{aid} = 24.3 - 0.0431 \times \text{family income} \tag{13.3.10}$$

Here we have replaced y with $\hat{aid}$ and x with $family_{income}$ to put the equation in context.

We mentioned earlier that a computer is usually used to compute the least squares line. A summary table based on computer output is shown in Table 7.15 for the Elmhurst data. The first column of numbers provides estimates for b0 and b1, respectively. Compare these to the result from Example 7.16.

Table 7.15: Summary of least squares t for the Elmhurst data. Compare the parameter estimates in the rst column to the results of Example 7.16.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	24.3193	1.2915	18.83	0.0000
family_income	-0.0431	0.0108	-3.98	0.0002

Example 13.3.2

Examine the second, third, and fourth columns in Table 7.15. Can you guess what they represent?

Solution

We'll describe the meaning of the columns using the second row, which corresponds to β_1. The first column provides the point estimate for β_1, as we calculated in an earlier example: -0.0431. The second column is a standard error for this point estimate: 0.0108. The third column is a t test statistic for the null hypothesis that $\beta_1 = \beta_0 : T = -3.98$. The last column is the p-value for the t test statistic for the null hypothesis $\beta_1 = 0$ and a two-sided alternative hypothesis: 0.0002. We will get into more of these details in Section 7.4.

Example 13.3.3

Suppose a high school senior is considering Elmhurst College. Can she simply use the linear equation that we have estimated to calculate her nancial aid from the university?

Solution

She may use it as an estimate, though some qualifiers on this approach are important. First, the data all come from one freshman class, and the way aid is determined by the university may change from year to year. Second, the equation will provide an imperfect estimate. While the linear equation is good at capturing the trend in the data, no individual student's aid will be perfectly predicted.

Interpreting Regression Line Parameter Estimates

Interpreting parameters in a regression model is often one of the most important steps in the analysis.

Example 13.3.2

The slope and intercept estimates for the Elmhurst data are -0.0431 and 24.3. What do these numbers really mean?

Solution

Interpreting the slope parameter is helpful in almost any application. For each additional \$1,000 of family income, we would expect a student to receive a net difference of $\$1,000 \times (-0.0431) = -\$43.10$ in aid on average, i.e. \$43.10 less. Note that a higher family income corresponds to less aid because the coefficient of family income is negative in the model. We must be

cautious in this interpretation: while there is a real association, we cannot interpret a causal connection between the variables because these data are observational. That is, increasing a student's family income may not cause the student's aid to drop. (It would be reasonable to contact the college and ask if the relationship is causal, i.e. if Elmhurst College's aid decisions are partially based on students' family income.)

The estimated intercept $b_0 = 24.3$ (in $1000s) describes the average aid if a student's family had no income. The meaning of the intercept is relevant to this application since the family income for some students at Elmhurst is $0. In other applications, the intercept may have little or no practical value if there are no observations where x is near zero.

Interpreting parameters estimated by least squares

The slope describes the estimated difference in the y variable if the explanatory variable x for a case happened to be one unit larger. The intercept describes the average outcome of y if x = 0 and the linear model is valid all the way to x = 0, which in many applications is not the case.

Extrapolation is Treacherous

When those blizzards hit the East Coast this winter, it proved to my satisfaction that global warming was a fraud. That snow was freezing cold. But in an alarming trend, temperatures this spring have risen. Consider this: On February 6th it was 10 degrees. Today it hit almost 80. At this rate, by August it will be 220 degrees. So clearly folks the climate debate rages on.

(13.3.11)

[11] *http://www.colbertnation.com/the-col...videos/269929/*

Linear models can be used to approximate the relationship between two variables. However, these models have real limitations. Linear regression is simply a modeling framework. The truth is almost always much more complex than our simple line. For example, we do not know how the data outside of our limited window will behave.

Example 13.3.3

Use the model $\hat{aid} = 24.3 - 0.0431$ family income to estimate the aid of another freshman student whose family had income of $1 million.

Recall that the units of family income are in $1000s, so we want to calculate the aid for family income = 1000:

$$24.3 - 0.0431 \times \text{family income} = 24.3 - 0.0431 \times 1000 = -18.8 \quad (13.3.12)$$

The model predicts this student will have -$18,800 in aid (!). Elmhurst College cannot (or at least does not) require any students to pay extra on top of tuition to attend.

Applying a model estimate to values outside of the realm of the original data is called **extrapolation**. Generally, a linear model is only an approximation of the real relationship between two variables. If we extrapolate, we are making an unreliable bet that the approximate linear relationship will be valid in places where it has not been analyzed.

Using R^2 to describe the strength of a fit

We evaluated the strength of the linear relationship between two variables earlier using the correlation, R. However, it is more common to explain the strength of a linear t using R^2, called **R-squared**. If provided with a linear model, we might like to describe how closely the data cluster around the linear fit.

The R^2 of a linear model describes the amount of variation in the response that is explained by the least squares line. For example, consider the Elmhurst data, shown in Figure 7.16. The variance of the response variable, aid received, is $s^2_{aid} = 29.8$. However, if we apply our least squares line, then this model reduces our uncertainty in predicting

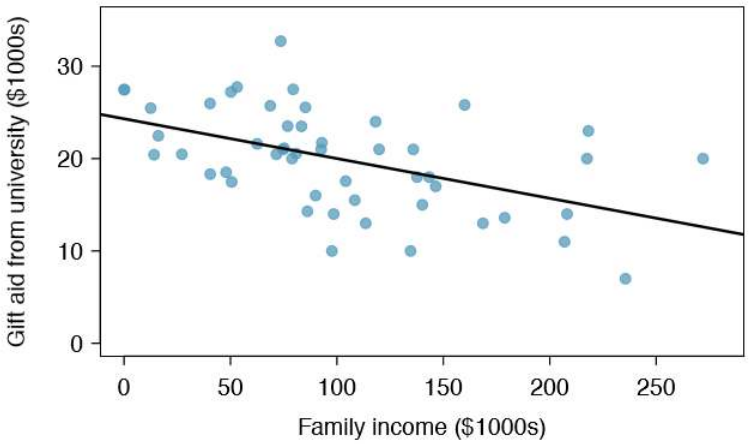


Figure 7.16: Gift aid and family income for a random sample of 50 freshman students from Elmhurst College, shown with the least squares regression line.

aid using a student's family income. The variability in the residuals describes how much variation remains after using the model: $s^2_{RES} = 22.4$. In short, there was a reduction of

$$\frac{s^2_{aid} - s^2_{RES}}{s^2_{GPA}} = \frac{29.9 - 22.4}{29.9} = \frac{7.5}{29.9} = 0.25 \tag{13.3.13}$$

or about 25% in the data's variation by using information about family income for predicting aid using a linear model. This corresponds exactly to the R-squared value:

$$R = -0.499 R^2 = 0.25 \tag{13.3.14}$$

Exercise 13.3.5

If a linear model has a very strong negative relationship with a correlation of -0.97, how much of the variation in the response is explained by the explanatory variable?[12]

Categorical Predictors with two Levels

Categorical variables are also useful in predicting outcomes. Here we consider a categorical predictor with two levels (recall that a level is the same as a category). We'll consider Ebay auctions for a video game, Mario Kart for the Nintendo Wii, where both the total price of the auction and the condition of the game were recorded.[13] Here we want to predict total price based on game condition, which takes values used and new. A plot of the auction data is shown in Figure 7.17.

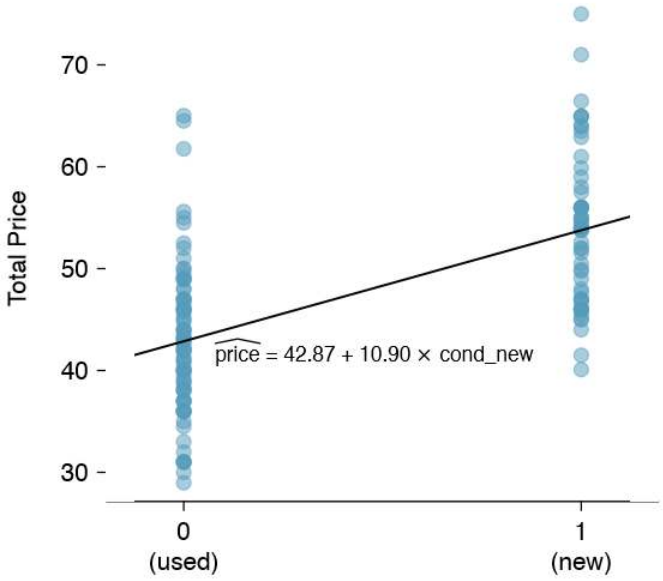


Figure 7.17: Total auction prices for the video game Mario Kart, divided into used (x = 0) and new (x = 1) condition games. The least squares regression line is also shown.

To incorporate the game condition variable into a regression equation, we must convert the categories into a numerical form. We will do so using an indicator variable called cond new, which takes value 1 when the game is new and 0 when the game is used. Using this indicator variable, the linear model may be written as

$$\hat{price} = \beta_0 + \beta_1 \times \text{cond new} \tag{13.3.15}$$

[12]*About* $R^2 = (-0.97)^2 = 0.94$ *or 94% of the variation is explained by the linear model.*

[13]*These data were collected in Fall 2009 and may be found at openintro.org.*

Table 7.18: Least squares regression summary for the nal auction price against the condition of the game.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	42.87	0.81	52.67	0.0000
cond_new	10.90	1.26	8.66	0.0000

The fitted model is summarized in Table 7.18, and the model with its parameter estimates is given as

$$\hat{price} = 42.87 + 10.90 \times \text{cond new} \tag{13.3.16}$$

For categorical predictors with just two levels, the linearity assumption will always be satis ed. However, we must evaluate whether the residuals in each group are approximately normal and have approximately equal variance. As can be seen in Figure 7.17, both of these conditions are reasonably satis ed by the auction data.

Example 7.22 Interpret the two parameters estimated in the model for the price of Mario Kart in eBay auctions.

The intercept is the estimated price when cond new takes value 0, i.e. when the game is in used condition. That is, the average selling price of a used version of the game is $42.87.

The slope indicates that, on average, new games sell for about $10.90 more than used games.

TIP: Interpreting model estimates for categorical predictors.

The estimated intercept is the value of the response variable for the first category (i.e. the category corresponding to an indicator value of 0). The estimated slope is the average change in the response variable between the two categories.

We'll elaborate further on this Ebay auction data in Chapter 8, where we examine the influence of many predictor variables simultaneously using multiple regression. In multiple regression, we will consider the association of auction price with regard to each variable while controlling for the influence of other variables. This is especially important since some of the predictors are associated. For example, auctions with games in new condition also often came with more accessories.

This page titled 13.3: Fitting a Line by Least Squares Regression is shared under a CC BY-SA 3.0 license and was authored, remixed, and/or curated by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel via source content that was edited to the style and standards of the LibreTexts platform.

- **7.3: Fitting a Line by Least Squares Regression** by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel is licensed CC BY-SA 3.0. Original source: https://www.openintro.org/book/os.

13.4: Types of Outliers in Linear Regression

In this section, we identify criteria for determining which outliers are important and influential. Outliers in regression are observations that fall far from the "cloud" of points. These points are especially important because they can have a strong influence on the least squares line.

Example 13.4.1

There are six plots shown in Figure **13.4.1** along with the least squares line and residual plots. For each scatter plot and residual plot pair, identify any obvious outliers and note how they influence the least squares line. Recall that an outlier is any point that doesn't appear to belong with the vast majority of the other points.

1. There is one outlier far from the other points, though it only appears to slightly influence the line.
2. There is one outlier on the right, though it is quite close to the least squares line, which suggests it wasn't very influential.
3. There is one point far away from the cloud, and this outlier appears to pull the least squares line up on the right; examine how the line around the primary cloud doesn't appear to t very well.
4. There is a primary cloud and then a small secondary cloud of four outliers. The secondary cloud appears to be influencing the line somewhat strongly, making the least square line t poorly almost everywhere. There might be an interesting explanation for the dual clouds, which is something that could be investigated.
5. There is no obvious trend in the main cloud of points and the outlier on the right appears to largely control the slope of the least squares line.
6. There is one outlier far from the cloud, however, it falls quite close to the least squares line and does not appear to be very influential.

Examine the residual plots in Figure **13.4.1**. You will probably nd that there is some trend in the main clouds of (3) and (4). In these cases, the outliers influenced the slope of the least squares lines. In (5), data with no clear trend were assigned a line with a large trend simply due to one outlier (!).

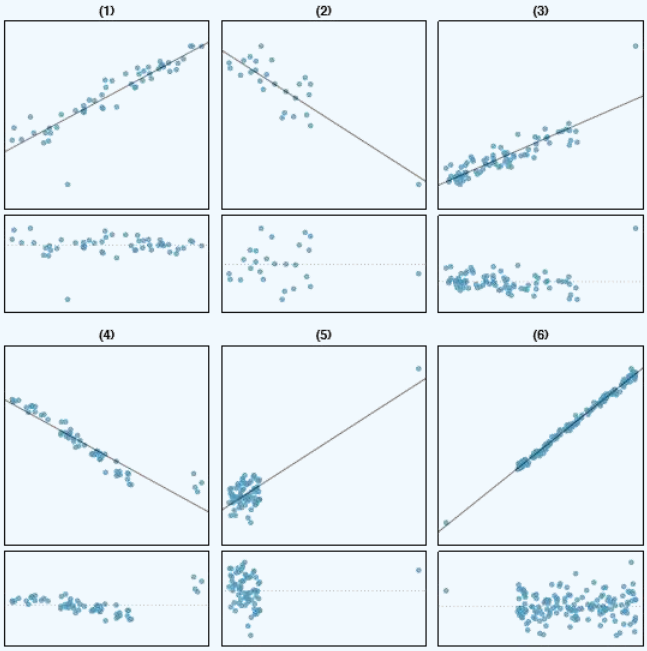


Figure **13.4.1**: *Six plots, each with a least squares line and residual plot. All data sets have at least one outlier.*

Definition: Leverage

Points that fall horizontally away from the center of the cloud tend to pull harder on the line, so we call them points with high leverage.

Points that fall horizontally far from the line are points of high leverage; these points can strongly influence the slope of the least squares line. If one of these high leverage points does appear to actually invoke its influence on the slope of the line (as in cases (3), (4), and (5) of Example **13.4.1**) then we call it an **influential point**. Usually we can say a point is influential if, had we plotted the line without it, the influential point would have been unusually far from the least squares line.

It is tempting to remove outliers. Do not do this without a very good reason. Models that ignore exceptional (and interesting) cases often perform poorly. For instance, if a financial firm ignored the largest market swings - the "outliers" - they would soon go bankrupt by making poorly thought-out investments.

Caution: Don't ignore outliers when fitting a final model

If there are outliers in the data, they should not be removed or ignored without a good reason. Whatever final model is fit to the data would not be very helpful if it ignores the most exceptional cases.

Caution: Outliers for a categorical predictor with two levels

Be cautious about using a categorical predictor when one of the levels has very few observations. When this happens, those few observations become inuential points.

This page titled 13.4: Types of Outliers in Linear Regression is shared under a CC BY-SA 3.0 license and was authored, remixed, and/or curated by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel via source content that was edited to the style and standards of the LibreTexts platform.

- **7.4: Types of Outliers in Linear Regression** by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel is licensed CC BY-SA 3.0. Original source: https://www.openintro.org/book/os.

13.5: Inference for Linear Regression

In this section we discuss uncertainty in the estimates of the slope and y-intercept for a regression line. Just as we identified standard errors for point estimates in previous chapters, we first discuss standard errors for these new estimates. However, in the case of regression, we will identify standard errors using statistical software.

Midterm elections and unemployment

Elections for members of the United States House of Representatives occur every two years, coinciding every four years with the U.S. Presidential election. The set of House elections occurring during the middle of a Presidential term are called midterm elections. In America's two-party system, one political theory suggests the higher the unemployment rate, the worse the President's party will do in the midterm elections.

To assess the validity of this claim, we can compile historical data and look for a connection. We consider every midterm election from 1898 to 2010, with the exception of those elections during the Great Depression. Figure **13.5.1** shows these data and the least-squares regression line:

$$\tag{13.5.1}$$

$$= -6.71 - 1.00 \times (\text{unemployment rate}) \tag{13.5.2}$$

We consider the percent change in the number of seats of the President's party (e.g. percent change in the number of seats for Democrats in 2010) against the unemployment rate.

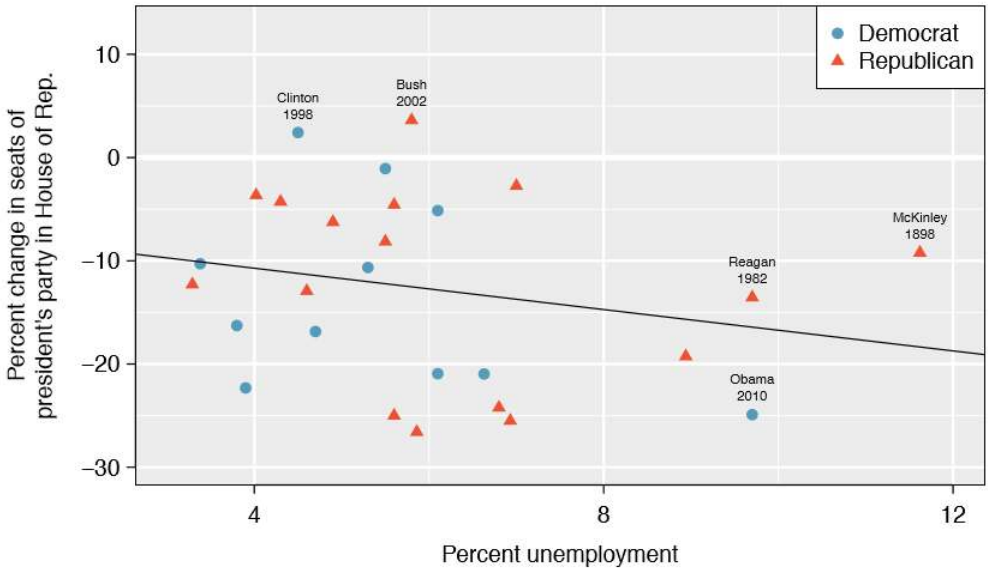


Figure **13.5.1**: The percent change in House seats for the President's party in each election from 1898 to 2010 plotted against the unemployment rate. The two points for the Great Depression have been removed, and a least squares regression line has been t to the data.

Examining the data, there are no clear deviations from linearity, the constant variance condition, or in the normality of residuals (though we don't examine a normal probability plot here). While the data are collected sequentially, a separate analysis was used to check for any apparent correlation between successive observations; no such correlation was found.

Exercise 13.5.1

The data for the Great Depression (1934 and 1938) were removed because the unemployment rate was 21% and 18%, respectively. Do you agree that they should be removed for this investigation? Why or why not?

Answer

We will provide two considerations. Each of these points would have very high leverage on any least-squares regression line, and years with such high unemployment may not help us understand what would happen in other years where the unemployment is only modestly high. On the other hand, these are exceptional cases, and we would be discarding important information if we exclude them from a final analysis.

There is a negative slope in the line shown in Figure **13.5.1**. However, this slope (and the y-intercept) are only estimates of the parameter values. We might wonder, is this convincing evidence that the "true" linear model has a negative slope? That is, do the data provide strong evidence that the political theory is accurate? We can frame this investigation into a one-sided statistical hypothesis test:

- H_0: $\beta_1 = 0$. The true linear model has slope zero.
- HA: $\beta_1 < 0$. The true linear model has a slope less than zero. The higher the unemployment, the greater the losses for the President's party in the House of Representatives.

We would reject H_0 in favor of H_A if the data provide strong evidence that the true slope parameter is less than zero. To assess the hypotheses, we identify a standard error for the estimate, compute an appropriate test statistic, and identify the p-value.

Understanding regression output from software

Just like other point estimates we have seen before, we can compute a standard error and test statistic for β_1. We will generally label the test statistic using a T, since it follows the t distribution.

We will rely on statistical software to compute the standard error and leave the explanation of how this standard error is determined to a second or third statistics course. Table **13.5.1** shows software output for the least squares regression line in Figure **13.5.1**. The row labeled unemp represents the information for the slope, which is the coefficient of the unemployment variable.

Table **13.5.1**: Output from statistical software for the regression line modeling the midterm election losses for the President's party as a response to unemployment.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-6.7142	5.4567	-1.23	0.2300
unemp	-1.0010	0.8717	-1.15	0.2617

Example 13.5.2

What do the first and second columns of Table **13.5.1** represent?

Solution

The entries in the first column represent the least squares estimates, β_0 and β_1, and the values in the second column correspond to the standard errors of each estimate.

We previously used a t test statistic for hypothesis testing in the context of numerical data. Regression is very similar. In the hypotheses we consider, the null value for the slope is 0, so we can compute the test statistic using the T (or Z) score formula:

$$T = \frac{\text{estimate - null value}}{SE} = \frac{-1.0010 - 0}{0.8717} = -1.15$$

We can look for the one-sided p-value - shown in Figure **13.5.2** - using the probability table for the t distribution in Appendix B.2

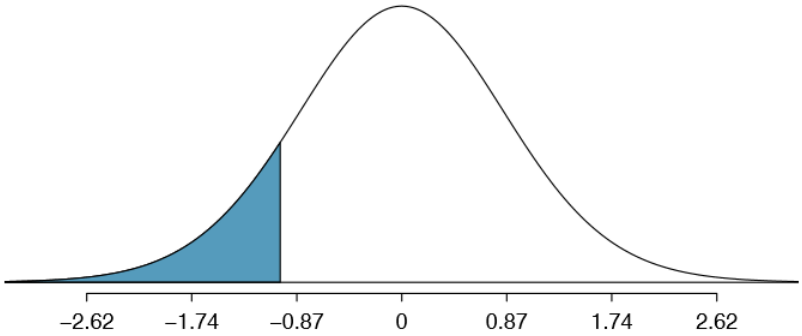


Figure **13.5.2**: The distribution shown here is the sampling distribution for b1, if the null hypothesis was true. The shaded tail represents the p-value for the hypothesis test evaluating whether there is convincing evidence that higher unemployment corresponds to a greater loss of House seats for the President's party during a midterm election.

Exercise 13.5.2

Table **13.5.1** offers the degrees of freedom for the test statistic T: df = 25. Identify the p-value for the hypothesis test.

Answer

Add answer text here and it will automatically be hidden if you have a "AutoNum" template active on the page.

Looking in the 25 degrees of freedom row in Appendix B.2, we see that the absolute value of the test statistic is smaller than any value listed, which means the tail area and therefore also the p-value is larger than 0.100 (one tail!). Because the p-value is so large, we fail to reject the null hypothesis. That is, the data do not provide convincing evidence that a higher unemployment rate has any correspondence with smaller or larger losses for the President's party in the House of Representatives in midterm elections.

We could have identified the t test statistic from the software output in Table **13.5.1**, shown in the second row (unemp) and third column (t value). The entry in the second row and last column in Table **13.5.1** represents the p-value for the two-sided hypothesis test where the null value is zero. The corresponding one-sided test would have a p-value half of the listed value.

Inference for regression

We usually rely on statistical software to identify point estimates and standard errors for parameters of a regression line. After verifying conditions hold for fitting a line, we can use the methods learned in Section 5.3 for the t distribution to create con dence intervals for regression parameters or to evaluate hypothesis tests.

Caution: Don't carelessly use the p-value from regression output

The last column in regression output often lists p-values for one particular hypothesis: a two-sided test where the null value is zero. If your test is one-sided and the point estimate is in the direction of HA, then you can halve the software's p-value to get the one-tail area. If neither of these scenarios match your hypothesis test, be cautious about using the software output to obtain the p-value.

Example 13.5.3

Examine Figure 7.16, which relates the Elmhurst College aid and student family income. How sure are you that the slope is statistically significantly different from zero? That is, do you think a formal hypothesis test would reject the claim that the true slope of the line should be zero?

Solution

While the relationship between the variables is not perfect, there is an evident decreasing trend in the data. This suggests the hypothesis test will reject the null claim that the slope is zero.

Exercise 13.5.3

Table **13.5.2** shows statistical software output from tting the least squares regression line shown in Figure 7.16. Use this output to formally evaluate the following hypotheses.

- H_0: The true coefficient for family income is zero.
- H_A: The true coefficient for family income is not zero.

Table **13.5.2**: Summary of least squares t for the Elmhurst College data.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	24.3193	1.2915	18.83	0.0000
family_income	-0.0431	0.0108	-3.98	0.0002

Answer

We look in the second row corresponding to the family income variable. We see the point estimate of the slope of the line is -0.0431, the standard error of this estimate is 0.0108, and the t test statistic is -3.98. The p-value corresponds exactly to the two-sided test we are interested in: 0.0002. The p-value is so small that we reject the null hypothesis and conclude that family income and nancial aid at Elmhurst College for freshman entering in the year 2011 are negatively correlated and the true slope parameter is indeed less than 0, just as we believed in Example 7.27.

TIP: Always check assumptions

If conditions for tting the regression line do not hold, then the methods presented here should not be applied. The standard error or distribution assumption of the point estimate - assumed to be normal when applying the t test statistic - may not be valid.

An alternative Test Statistic

We considered the t test statistic as a way to evaluate the strength of evidence for a hypothesis test in Section 7.4.2. However, we could focus on R^2. Recall that R^2 described the proportion of variability in the response variable (y) explained by the explanatory variable (x). If this proportion is large, then this suggests a linear relationship exists between the variables. If this proportion is small, then the evidence provided by the data may not be convincing.

This concept - considering the amount of variability in the response variable explained by the explanatory variable - is a key component in some statistical techniques. The analysis of variance (ANOVA) technique introduced in Section 5.5 uses this general principle. The method states that if enough variability is explained away by the categories, then we would conclude the mean varied between the categories. On the other hand, we might not be convinced if only a little variability is explained. ANOVA can be further employed in advanced regression modeling to evaluate the inclusion of explanatory variables, though we leave these details to a later course.

This page titled 13.5: Inference for Linear Regression is shared under a CC BY-SA 3.0 license and was authored, remixed, and/or curated by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel via source content that was edited to the style and standards of the LibreTexts platform.

- **7.5: Inference for Linear Regression** by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel is licensed CC BY-SA 3.0. Original source: https://www.openintro.org/book/os.

13.6: Exercises

Line tting, residuals, and correlation

7.1 Visualize the residuals. The scatterplots shown below each have a superimposed regression line. If we were to construct a residual plot (residuals versus x) for each, describe what those plots would look like.

7.2 Trends in the residuals. Shown below are two plots of residuals remaining after fitting a linear model to two different sets of data. Describe important features and determine if a linear model would be appropriate for these data. Explain your reasoning.

7.3 Identify relationships, Part I. For each of the six plots, identify the strength of the relationship (e.g. weak, moderate, or strong) in the data and whether tting a linear model would be reasonable.

7.4 Identify relationships, Part I. For each of the six plots, identify the strength of the relationship (e.g. weak, moderate, or strong) in the data and whether tting a linear model would be reasonable.

7.5 The two scatterplots below show the relationship between nal and mid-semester exam grades recorded during several years for a Statistics course at a university.

(a) Based on these graphs, which of the two exams has the strongest correlation with the final exam grade? Explain.

(b) Can you think of a reason why the correlation between the exam you chose in part (a) and the final exam is higher?

7.6 Husbands and wives, Part I. The Great Britain Office of Population Census and Surveys once collected data on a random sample of 170 married couples in Britain, recording the age (in years) and heights (converted here to inches) of the husbands and wives.16 The scatterplot on the left shows the wife's age plotted against her husband's age, and the plot on the right shows wife's height plotted against husband's height.

(a) Describe the relationship between husbands' and wives' ages.

(b) Describe the relationship between husbands' and wives' heights.

(c) Which plot shows a stronger correlation? Explain your reasoning.

(d) Data on heights were originally collected in centimeters, and then converted to inches. Does this conversion affect the correlation between husbands' and wives' heights?

7.7 Match the correlation, Part I.

Match the calculated correlations to the corresponding scatterplot.

(a) R = -0.7

(b) R = 0.45

(c) R = 0.06

(d) R = 0.92

7.8 Match the correlation, Part II.

Match the calculated correlations to the corresponding scatterplot.

(a) R = 0.49

(b) R = -0.48

(c) R = -0.03

(d) R = -0.85

[16]*D.J. Hand. A handbook of small data sets. Chapman & Hall/CRC, 1994.*

7.9 Speed and height. 1,302 UCLA students were asked to ll out a survey where they were asked about their height, fastest speed they have ever driven, and gender. The scatterplot on the left displays the relationship between height and fastest speed, and the scatterplot on the right displays the breakdown by gender in this relationship.

(a) Describe the relationship between height and fastest speed.

(b) Why do you think these variables are positively associated?

(c) What role does gender play in the relationship between height and fastest driving speed?

7.10 Trees. The scatterplots below show the relationship between height, diameter, and volume of timber in 31 felled black cherry trees. The diameter of the tree is measured 4.5 feet above the ground.17

(a) Describe the relationship between volume and height of these trees.

(b) Describe the relationship between volume and diameter of these trees.

(c) Suppose you have height and diameter measurements for another black cherry tree. Which of these variables would be preferable to use to predict the volume of timber in this tree using a simple linear regression model? Explain your reasoning.

[17]*Source: R Dataset, http://stat.ethz.ch/R-manual/R-patch...tml/trees.html*.

7.11 The Coast Starlight, Part I. The Coast Starlight Amtrak train runs from Seattle to Los Angeles. The scatterplot below displays the distance between each stop (in miles) and the amount of time it takes to travel from one stop to another (in minutes).

(a) Describe the relationship between distance and travel time.

(b) How would the relationship change if travel time was instead measured in hours, and distance was instead measured in kilometers?

(c) Correlation between travel time (in miles) and distance (in minutes) is R = 0.636. What is the correlation between travel time (in kilometers) and distance (in hours)?

7.12 Crawling babies, Part I. A study conducted at the University of Denver investigated whether babies take longer to learn to crawl in cold months, when they are often bundled in clothes that restrict their movement, than in warmer months.18 Infants born during the study year were split into twelve groups, one for each birth month. We consider the average crawling age of babies in each group against the average temperature when the babies are six months old (that's when babies often begin trying to crawl). Temperature is measured in degrees Fahrenheit (^{0}F) and age is measured in weeks.

(a) Describe the relationship between temperature and crawling age.

(b) How would the relationship change if temperature was measured in degrees Celsius (^{0}C) and age was measured in months?

(c) The correlation between temperature in ^{0}F and age in weeks was R = -0.70. If we converted the temperature to ^{0}C and age to months, what would the correlation be?

[18]J.B. Benson. "Season of birth and onset of locomotion: Theoretical and methodological implications". In: Infant behavior and development 16.1 (1993), pp. 69-81. issn: 0163-6383.

7.13 Body measurements, Part I. Researchers studying anthropometry collected body girth measurements and skeletal diameter measurements, as well as age, weight, height and gender for 507 physically active individuals.19 The scatterplot below shows the relationship between height and shoulder girth (over deltoid muscles), both measured in centimeters.

(a) Describe the relationship between shoulder girth and height.

(b) How would the relationship change if shoulder girth was measured in inches while the units of height remained in centimeters?

7.14 Body measurements, Part II. The scatterplot below shows the relationship between weight measured in kilograms and hip girth measured in centimeters from the data described in Exercise 7.13.

(a) Describe the relationship between hip girth and weight.

(b) How would the relationship change if weight was measured in pounds while the units for hip girth remained in centimeters?

7.15 Correlation, Part I. What would be the correlation between the ages of husbands and wives if men always married woman who were

(a) 3 years younger than themselves?

(b) 2 years older than themselves?

(c) half as old as themselves?

7.16 Correlation, Part II. What would be the correlation between the annual salaries of males and females at a company if for a certain type of position men always made

(a) $5,000 more than women?

(b) 25% more than women?

(c) 15% less than women?

[19]*G. Heinz et al. "Exploring relationships in body dimensions". In: Journal of Statistics Education 11.2 (2003).*

Fitting a line by least squares regression

7.17 Tourism spending. The Association of Turkish Travel Agencies reports the number of foreign tourists visiting Turkey and tourist spending by year.20 The scatterplot below shows the relationship between these two variables along with the least squares fit.

(a) Describe the relationship between number of tourists and spending.

(b) What are the explanatory and response variables?

(c) Why might we want to t a regression line to these data?

(d) Do the data meet the conditions required for tting a least squares line? In addition to the scatterplot, use the residual plot and histogram to answer this question.

7.18 Nutrition at Starbucks, Part I. The scatterplot below shows the relationship between the number of calories and amount of carbohydrates (in grams) Starbucks food menu items contain.21 Since Starbucks only lists the number of calories on the display items, we are interested in predicting the amount of carbs a menu item has based on its calorie content.

(a) Describe the relationship between number of calories and amount of carbohydrates (in grams) that Starbucks food menu items contain.

(b) In this scenario, what are the explanatory and response variables?

(c) Why might we want to t a regression line to these data?

(d) Do these data meet the conditions required for tting a least squares line?

7.19 The Coast Starlight, Part II. Exercise 7.11 introduces data on the Coast Starlight Amtrak train that runs from Seattle to Los Angeles. The mean travel time from one stop to the next on the Coast Starlight is 129 mins, with a standard deviation of 113 minutes. The mean distance traveled from one stop to the next is 107 miles with a standard deviation of 99 miles. The correlation between travel time and distance is 0.636.

(a) Write the equation of the regression line for predicting travel time.

(b) Interpret the slope and the intercept in this context.

(c) Calculate R^2 of the regression line for predicting travel time from distance traveled for the Coast Starlight, and interpret R^2 in the context of the application.

(d) The distance between Santa Barbara and Los Angeles is 103 miles. Use the model to estimate the time it takes for the Starlight to travel between these two cities.

(e) It actually takes the the Coast Starlight about 168 mins to travel from Santa Barbara to Los Angeles. Calculate the residual and explain the meaning of this residual value.

(f) Suppose Amtrak is considering adding a stop to the Coast Starlight 500 miles away from Los Angeles. Would it be appropriate to use this linear model to predict the travel time from Los Angeles to this point?

[21]*Source: Starbucks.com, collected on March 10, 2011, www.starbucks.com/menu/nutrition.*

7.20 Body measurements, Part III. Exercise 7.13 introduces data on shoulder girth and height of a group of individuals. The mean shoulder girth is 108.20 cm with a standard deviation of 10.37 cm. The mean height is 171.14 cm with a standard deviation of 9.41 cm. The correlation between height and shoulder girth is 0.67.

(a) Write the equation of the regression line for predicting height.

(b) Interpret the slope and the intercept in this context.

(c) Calculate R2 of the regression line for predicting height from shoulder girth, and interpret it in the context of the application.

(d) A randomly selected student from your class has a shoulder girth of 100 cm. Predict the height of this student using the model.

(e) The student from part (d) is 160 cm tall. Calculate the residual, and explain what this residual means.

(f) A one year old has a shoulder girth of 56 cm. Would it be appropriate to use this linear model to predict the height of this child?

7.21 Grades and TV. Data were collected on the number of hours per week students watch TV and the grade they earned in a biology class on a 100 point scale. Based on the scatterplot and the residual plot provided, describe the relationship between the two variables, and determine if a simple linear model is appropriate to predict a student's grade from the number of hours per week the student watches TV.

7.22 Nutrition at Starbucks, Part II. Exercise 7.18 introduced a data set on nutrition information on Starbucks food menu items. Based on the scatterplot and the residual plot provided, describe the relationship between the protein content and calories of these menu items, and determine if a simple linear model is appropriate to predict amount of protein from the number of calories.

7.23 Helmets and lunches. The scatterplot shows the relationship between socioeconomic status measured as the percentage of children in a neighborhood receiving reduced-fee lunches at school (lunch) and the percentage of bike riders in the neighborhood wearing helmets (helmet). The average percentage of children receiving reduced-fee lunches is 30.8% with a standard deviation of 26.7% and the average percentage of bike riders wearing helmets is 38.8% with a standard deviation of 16.9%.

(a) If the R2 for the least-squares regression line for these data is 72%, what is the correlation between lunch and helmet?

(b) Calculate the slope and intercept for the leastsquares regression line for these data.

(c) Interpret the intercept of the least-squares regression line in the context of the application.

(d) Interpret the slope of the least-squares regression line in the context of the application.

(e) What would the value of the residual be for a neighborhood where 40% of the children receive reduced-fee lunches and 40% of the bike riders wear helmets? Interpret the meaning of this residual in the context of the application.

Types of outliers in linear regression

7.24 Outliers, Part I. Identify the outliers in the scatterplots shown below, and determine what type of outliers they are. Explain your reasoning.

7.25 Outliers, Part II. Identify the outliers in the scatterplots shown below and determine what type of outliers they are. Explain your reasoning.

7.26 Crawling babies, Part II. Exercise 7.12 introduces data on the average monthly temperature during the month babies first try to crawl (about 6 months after birth) and the average rst crawling age for babies born in a given month. A scatterplot of these two variables reveals a potential outlying month when the average temperature is about 53^0F and average crawling age is about 28.5 weeks. Does this point have high leverage? Is it an inuential point?

7.27 Urban homeowners, Part I. The scatterplot below shows the percent of families who own their home vs. the percent of the population living in urban areas in 2010.22 There are 52 observations, each corresponding to a state in the US. Puerto Rico and District of Columbia are also included.

(a) Describe the relationship between the percent of families who own their home and the percent of the population living in urban areas in 2010.

(b) The outlier at the bottom right corner is District of Columbia, where 100% of the population is considered urban. What type of

outlier is this observation?

Inference for linear regression

In the following exercises, visually check the conditions for tting a least squares regression line, but you do not need to report these conditions in your solutions.

7.28 Beer and blood alcohol content. Many people believe that gender, weight, drinking habits, and many other factors are much more important in predicting blood alcohol content (BAC) than simply considering the number of drinks a person consumed. Here we examine data from sixteen student volunteers at Ohio State University who each drank a randomly assigned number of cans of beer. These students were evenly divided between men and women, and they differed in weight and drinking habits. Thirty minutes later, a police officer measured their blood alcohol content (BAC) in grams of alcohol per deciliter of blood.23 The scatterplot and regression table summarize the ndings.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.0127	0.0126	-1.00	0.3320
beers	0.0180	0.0024	7.48	0.0000

(a) Describe the relationship between the number of cans of beer and BAC.

(b) Write the equation of the regression line. Interpret the slope and intercept in context.

(c) Do the data provide strong evidence that drinking more cans of beer is associated with an increase in blood alcohol? State the null and alternative hypotheses, report the p-value, and state your conclusion.

(d) The correlation coefficient for number of cans of beer and BAC is 0.89. Calculate R^2 and interpret it in context.

(e) Suppose we visit a bar, ask people how many drinks they have had, and also take their BAC. Do you think the relationship between number of drinks and BAC would be as strong as the relationship found in the Ohio State study?

[22]*United States Census Bureau, 2010 Census Urban and Rural Classi cation and Urban Area Criteria and Housing Characteristics: 2010.*

[23]*J. Malkevitch and L.M. Lesser. For All Practical Purposes: Mathematical Literacy in Today's World. WH Freeman & Co, 2008.*

7.29 Body measurements, Part IV. The scatterplot and least squares summary below show the relationship between weight measured in kilograms and height measured in centimeters of 507 physically active individuals.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-105.0113	7.5394	-13.93	0.0000
height	1.0176	0.0440	23.13	0.0000

(a) Describe the relationship between height and weight.

(b) Write the equation of the regression line. Interpret the slope and intercept in context.

(c) Do the data provide strong evidence that an increase in height is associated with an increase in weight? State the null and alternative hypotheses, report the p-value, and state your conclusion.

(d) The correlation coefficient for height and weight is 0.72. Calculate R2 and interpret it in context.

7.30 Husbands and wives, Part II. Exercise 7.6 presents a scatterplot displaying the relationship between husbands' and wives' ages in a random sample of 170 married couples in Britain, where both partners' ages are below 65 years. Given below is summary output of the least squares fit for predicting wife's age from husband's age.

	Estimate	Std. Error	t value	Pr(>|t|)

(Intercept)	1.5740	1.1501	1.37	0.1730
age_husband	0.9112	0.0259	35.25	0.0000

(a) We might wonder, is the age difference between husbands and wives constant over time? If this were the case, then the slope parameter would be 1 = 1. Use the information above to evaluate if there is strong evidence that the difference in husband and wife ages actually has changed.

(b) Write the equation of the regression line for predicting wife's age from husband's age.

(c) Interpret the slope and intercept in context.

(d) Given that R2 = 0:88, what is the correlation of ages in this data set?

(e) You meet a married man from Britain who is 55 years old. What would you predict his wife's age to be? How reliable is this prediction?

(f) You meet another married man from Britain who is 85 years old. Would it be wise to use the same linear model to predict his wife's age? Explain.

7.31 Husbands and wives, Part III. The scatterplot below summarizes husbands' and wives' heights in a random sample of 170 married couples in Britain, where both partners' ages are below 65 years. Summary output of the least squares t for predicting wife's height from husband's height is also provided in the table.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	43.5755	4.6842	9.30	0.0000
height_husband	0.2863	0.0686	4.17	0.0000

(a) Is there strong evidence that taller men marry taller women? State the hypotheses and include any information used to conduct the test.

(b) Write the equation of the regression line for predicting wife's height from husband's height.

(c) Interpret the slope and intercept in the context of the application.

(d) Given that R2 = 0:09, what is the correlation of heights in this data set?

(e) You meet a married man from Britain who is 5'9" (69 inches). What would you predict his wife's height to be? How reliable is this prediction?

(f) You meet another married man from Britain who is 6'7" (79 inches). Would it be wise to use the same linear model to predict his wife's height? Why or why not?

7.32 Urban homeowners, Part II. Exercise 7.27 gives a scatterplot displaying the relationship between the percent of families that own their home and the percent of the population living in urban areas. Below is a similar scatterplot, excluding District of Columbia, as well as the residuals plot. There were 51 cases.

(a) For these data, R2 = 0:28. What is the correlation? How can you tell if it is positive or negative?

(b) Examine the residual plot. What do you observe? Is a simple least squares fit appropriate for these data?

7.33 Babies. Is the gestational age (time between conception and birth) of a low birth-weight baby useful in predicting head circumference at birth? Twenty- ve low birth-weight babies were studied at a Harvard teaching hospital; the investigators calculated the regression of head circumference (measured in centimeters) against gestational age (measured in weeks). The estimated regression line is

head circdumference = 3:91 + 0:78 gestational age

(a) What is the predicted head circumference for a baby whose gestational age is 28 weeks?

(b) The standard error for the coefficient of gestational age is 0.35, which is associated with df = 23. Does the model provide strong evidence that gestational age is signi cantly associated with head circumference?

7.34 Rate my professor. Some college students critique professors' teaching at RateMyProfessors.com, a web page where students anonymously rate their professors on quality, easiness, and attractiveness. Using the self-selected data from this public forum, researchers examine the relations between quality, easiness, and attractiveness for professors at various universities. In this exercise we will work with a portion of these data that the researchers made publicly available.[24]

The scatterplot on the right shows the relationship between teaching evaluation score (higher score means better) and standardized beauty score (a score of 0 means average, negative score means below average, and a positive score means above average) for a sample of 463 professors. Given below are associated diagnostic plots. Also given is a regression output for predicting teaching evaluation score from beauty score.

[24] *J. Felton et al. "Web-based student evaluations of professors: the relations between perceived quality, easiness and sexiness". In: Assessment & Evaluation in Higher Education 29.1 (2004), pp. 91-108.*

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	4.010	0.0255	157.21	0.0000
beauty	----------	0.0322	4.13	0.0000

(a) Given that the average standardized beauty score is -0.0883 and average teaching evaluation score is 3.9983, calculate the slope. Alternatively, the slope may be computed using just the information provided in the model summary table.

(b) Do these data provide convincing evidence that the slope of the relationship between teaching evaluation and beauty is positive? Explain your reasoning.

(c) List the conditions required for linear regression and check if each one is satis ed for this model.

Contributors

David M Diez (Google/YouTube), Christopher D Barr (Harvard School of Public Health), Mine Çetinkaya-Rundel (Duke University)

This page titled 13.6: Exercises is shared under a CC BY-SA 3.0 license and was authored, remixed, and/or curated by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel via source content that was edited to the style and standards of the LibreTexts platform.

- **7.E: Introduction to Linear Regression (Exercises)** by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel has no license indicated. Original source: https://www.openintro.org/book/os.

CHAPTER OVERVIEW

14: Multiple and Logistic Regression

The principles of simple linear regression lay the foundation for more sophisticated regression methods used in a wide range of challenging settings. In Chapter 8, we explore multiple regression, which introduces the possibility of more than one predictor, and logistic regression, a technique for predicting categorical outcomes with two possible categories.

14.1: Introduction to Multiple Regression
14.2: Model Selection
14.3: Checking Model Assumptions using Graphs
14.4: Introduction to Logistic Regression
14.5: Exercises
14.6: Statistical Literacy
14.E: Regression (Exercises)

This page titled 14: Multiple and Logistic Regression is shared under a CC BY-SA 3.0 license and was authored, remixed, and/or curated by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel via source content that was edited to the style and standards of the LibreTexts platform.

14.1: Introduction to Multiple Regression

Multiple regression extends simple two-variable regression to the case that still has one response but many predictors (denoted $x_1, x_2, x_3, ...$). The method is motivated by scenarios where many variables may be simultaneously connected to an output.

We will consider Ebay auctions of a video game called Mario Kart for the Nintendo Wii. The outcome variable of interest is the total price of an auction, which is the highest bid plus the shipping cost. We will try to determine how total price is related to each characteristic in an auction while simultaneously controlling for other variables. For instance, all other characteristics held constant, are longer auctions associated with higher or lower prices? And, on average, how much more do buyers tend to pay for additional Wii wheels(plastic steering wheels that attach to the Wii controller) in auctions? Multiple regression will help us answer these and other questions.

The data set mario kart includes results from 141 auctions.[1] Four observations from this data set are shown in Table **14.1.1**, and descriptions for each variable are shown in Table **14.1.2**. Notice that the condition and stock photo variables are indicator variables. For instance, the cond new variable takes value 1 if the game up for auction is new and 0 if it is used. Using indicator variables in place of category names allows for these variables to be directly used in regression. See Section 7.2.7 for additional details. Multiple regression also allows for categorical variables with many levels, though we do not have any such variables in this analysis, and we save these details for a second or third course.

[1]*Diez DM, Barr CD, and Cetinkaya-Rundel M. 2012. openintro: OpenIntro data sets and supplemental functions. cran.r-project.org/web/packages/openintro.*

Table **14.1.1**: Four observations from the mario kart data set.

	price	cond new	stock photo	duration	wheels
1	51.55	1	1	3	1
2	37.04	0	1	3	1
⋮	⋮	⋮	⋮	⋮	⋮
140	38.76	0	0	7	0
141	54.51	1	1	1	2

Table **14.1.2**: *Variables and their descriptions for the mario kart data set.*

variable	description
price	final auction price plus shipping costs, in US dollars
cond_new	a coded two-level categorical variable, which takes value 1 when the game is new and 0 if the game is used
stock_photo	a coded two-level categorical variable, which takes value 1 if the primary photo used in the auction was a stock photo and 0 if the photo was unique to that auction
duration	the length of the auction, in days, taking values from 1 to 10
wheels	the number of Wii wheels included with the auction (a Wii wheel is a plastic racing wheel that holds the Wii controller and is an optional but helpful accessory for playing Mario Kart)

A Single-Variable Model for the Mario Kart Data

Let's fit a linear regression model with the game's condition as a predictor of auction price. The model may be written as

'_' allowed only in math mode

Results of this model are shown in Table **14.1.3** and a scatterplot for price versus game condition is shown in Figure **14.1.4**.

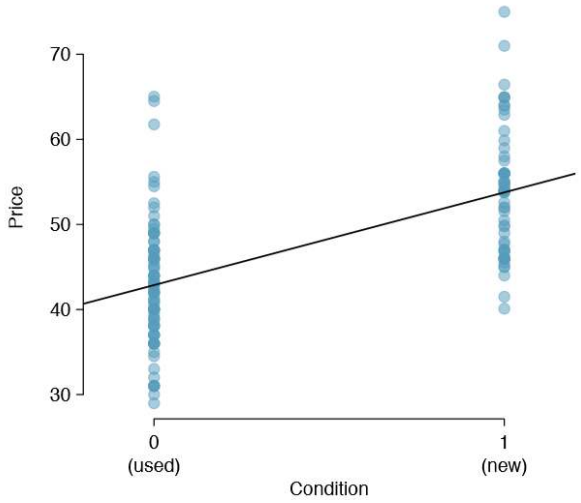


Figure **14.1.4**: Scatterplot of the total auction price against the game's condition. The least squares line is also shown.

Table **14.1.3**: Summary of a linear model for predicting auction price based on game condition.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	42.8711	0.8140	52.67	0.0000
cond_new	10.8996	1.2583	8.66	0.0000

Exercise 14.1.1

Figure **14.1.4**. Does the linear model seem reasonable?

Answer

Yes. Constant variability, nearly normal residuals, and linearity all appear reasonable.

Exercise 14.1.2

Interpret the coefficient for the game's condition in the model. Is this coefficient significantly different from 0?

Note that cond new is a two-level categorical variable that takes value 1 when the game is new and value 0 when the game is used. So 10.90 means that the model predicts an extra $10.90 for those games that are new versus those that are used. (See Section 7.2.7 for a review of the interpretation for two-level categorical predictor variables.) Examining the regression output in Table **14.1.3**, we can see that the p-value for cond new is very close to zero, indicating there is strong evidence that the coefficient is different from zero when using this simple one-variable model.

Including and Assessing Many Variables in a Model

Sometimes there are underlying structures or relationships between predictor variables. For instance, new games sold on Ebay tend to come with more Wii wheels, which may have led to higher prices for those auctions. We would like to fit a model that includes all potentially important variables simultaneously. This would help us evaluate the relationship between a predictor variable and the outcome while controlling for the potential influence of other variables. This is the strategy used in **multiple regression**. While we remain cautious about making any causal interpretations using multiple regression, such models are a common first step in providing evidence of a causal connection.

We want to construct a model that accounts for not only the game condition, as in Section 8.1.1, but simultaneously accounts for three other variables: stock photo, duration, and wheels.

'_' allowed only in math mode

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 \quad (14.1.1)$$

In this equation, y represents the total price, x_1 indicates whether the game is new, x_2 indicates whether a stock photo was used, x_3 is the duration of the auction, and x_4 is the number of Wii wheels included with the game. Just as with the single predictor case, a multiple regression model may be missing important components or it might not precisely represent the relationship between the outcome and the available explanatory variables.

While no model is perfect, we wish to explore the possibility that this one may fit the data reasonably well.

We estimate the parameters $\beta_0, \beta_1, \dots, \beta_4$ in the same way as we did in the case of a single predictor. We select $b_0, b_1, \dots, b_4$ that minimize the sum of the squared residuals:

$$SSE = e_1^2 + e_2^2 + \cdots + e_{141}^2 = \sum_{i=1}^{141} e_i^2 = \sum_{i=1}^{141} (y_i - \hat{y}_i)^2 \quad (14.1.2)$$

Here there are 141 residuals, one for each observation. We typically use a computer to minimize the sum in Equation (8.4) and compute point estimates, as shown in the sample output in Table **14.1.5**. Using this output, we identify the point estimates b_i of each β_i, just as we did in the one-predictor case.

Table **14.1.5**: Output for the regression model where price is the outcome and cond_new, stock_photo, duration, and wheels are the predictors.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	36.2110	1.5140	23.92	0.0000
cond_new	5.1306	1.0511	4.88	0.0000
stock_photo	1.0803	1.0568	1.02	0.3085
duration	-0.0268	0.1904	-0.14	0.8882
wheels	7.2852	0.5547	13.13	0.0000

Multiple regression model

A multiple regression model is a linear model with many predictors. In general, we write the model as

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k \quad (14.1.3)$$

when there are k predictors. We often estimate the β_i parameters using a computer.

Exercise 14.1.3

Write out the model in Equation (8.3) using the point estimates from Table **14.1.5**. How many predictors are there in this model?[3]

Answer

$\hat{y} = 36.21 + 5.13x_1 + 1.08x_2 - 0.03x_3 + 7.29x_4$, and there are k = 4 predictor variables.

Exercise 14.1.4

What does β_4, the coefficient of variable x_4 (Wii wheels), represent? What is the point estimate of β_4?

Answer

It is the average difference in auction price for each additional Wii wheel included when holding the other variables constant. The point estimate is $b_4 = 7.29$.

Exercise 14.1.5

Compute the residual of the first observation in Table **14.1.1** on page 355 using the equation identified in Exercise 8.5.

Answer

$e_i = y_i - \hat{y}_i = 51.55 - 49.62 = 1.93$, where 49.62 was computed using the variables values from the observation and the equation identified in Exercise 14.1.3.

Example 14.1.1

We estimated a coefficient for cond new in Section 8.1.1 of $b_1 = 10.90$ with a standard error of $SE_{b1} = 1.26$ when using simple linear regression. Why might there be a difference between that estimate and the one in the multiple regression setting?

If we examined the data carefully, we would see that some predictors are correlated. For instance, when we estimated the connection of the outcome price and predictor cond new using simple linear regression, we were unable to control for other variables like the number of Wii wheels included in the auction. That model was biased by the confounding variable wheels. When we use both variables, this particular underlying and unintentional bias is reduced or eliminated (though bias from other confounding variables may still remain).

Example 14.1.1 describes a common issue in multiple regression: correlation among predictor variables. We say the two predictor variables are **collinear** (pronounced as co-linear) when they are correlated, and this collinearity complicates model estimation. While it is impossible to prevent collinearity from arising in observational data, experiments are usually designed to prevent predictors from being collinear.

Exercise 14.1.6

The estimated value of the intercept is 36.21, and one might be tempted to make some interpretation of this coefficient, such as, it is the model's predicted price when each of the variables take value zero: the game is used, the primary image is not a stock photo, the auction duration is zero days, and there are no wheels included. Is there any value gained by making this interpretation?

Solution

Three of the variables (cond new, stock photo, and wheels) do take value 0, but the auction duration is always one or more days. If the auction is not up for any days, then no one can bid on it! That means the total auction price would always be zero for such an auction; the interpretation of the intercept in this setting is not insightful.

Adjusted R^2 as a better estimate of explained variance

We first used R^2 in Section 7.2 to determine the amount of variability in the response that was explained by the model:

$$R^2 = 1 - \frac{\text{variability in residuals}}{\text{variability in the outcome}} = 1 - \frac{Var(e_i)}{Var(y_i)} \tag{14.1.4}$$

where e_i represents the residuals of the model and yi the outcomes. This equation remains valid in the multiple regression framework, but a small enhancement can often be even more informative.

Exercise 14.1.7

The variance of the residuals for the model given in Exercise 8.7 is 23.34, and the variance of the total price in all the auctions is 83.06. Calculate R^2 for this model.

Solution

$R^2 = 1 - \frac{23.34}{83.06} = 0.719$.

This strategy for estimating R^2 is acceptable when there is just a single variable. However, it becomes less helpful when there are many variables. The regular R^2 is actually a biased estimate of the amount of variability explained by the model. To get a better estimate, we use the adjusted R^2.

Adjusted R^2 as a tool for model assessment

The **adjusted R^2** is computed as

$$R^2_{adj} = 1 - \frac{\frac{Var(e_i)}{(n-k-1)}}{\frac{Var(y_i)}{(n-1)}} = 1 - \frac{Var(e_i)}{Var(y_i)} \times \frac{n-1}{n-k-1} \quad (14.1.5)$$

where n is the number of cases used to fit the model and k is the number of predictor variables in the model.

Because k is never negative, the adjusted R^2 will be smaller - often times just a little smaller - than the unadjusted R^2. The reasoning behind the adjusted R^2 lies in the degrees of freedom associated with each variance.

In multiple regression, the degrees of freedom associated with the variance of the estimate of the residuals is n - k - 1, not n - 1. For instance, if we were to make predictions for new data using our current model, we would nd that the unadjusted R^2 is an overly optimistic estimate of the reduction in variance in the response, and using the degrees of freedom in the adjusted R^2 formula helps correct this bias.

Exercise 14.1.8

There were n = 141 auctions in the mario_kart data set and k = 4 predictor variables in the model. Use n, k, and the variances from Exercise 8.10 to calculate R^2_{adj} for the Mario Kart model.[9]

Solution

$R^2_{adj} = 1 - \frac{23.34}{83.06} \times \frac{141-1}{141-4-1} = 0.711$.

Exercise 14.1.9

Suppose you added another predictor to the model, but the variance of the errors $Var(e_i)$ didn't go down. What would happen to the R^2? What would happen to the adjusted R^2?

Solution

The unadjusted R^2 would stay the same and the adjusted R^2 would go down.

This page titled 14.1: Introduction to Multiple Regression is shared under a CC BY-SA 3.0 license and was authored, remixed, and/or curated by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel via source content that was edited to the style and standards of the LibreTexts platform.

- **8.1: Introduction to Multiple Regression** by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel is licensed CC BY-SA 3.0. Original source: https://www.openintro.org/book/os.

14.2: Model Selection

The best model is not always the most complicated. Sometimes including variables that are not evidently important can actually reduce the accuracy of predictions. In this section we discuss model selection strategies, which will help us eliminate from the model variables that are less important. In this section, and in practice, the model that includes all available explanatory variables is often referred to as the **full model**. Our goal is to assess whether the full model is the best model. If it isn't, we want to identify a smaller model that is preferable.

Identifying Variables in the Model that may not be Helpful

Table 8.6 provides a summary of the regression output for the full model for the auction data. The last column of the table lists p-values that can be used to assess hypotheses of the following form:

- H_0: $\beta_i = 0$ when the other explanatory variables are included in the model.
- H_A: $\beta_i \neq 0$ when the other explanatory variables are included in the model.

Table 8.6: The fit for the full regression model, including the adjusted R^2.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	36.2110	1.5140	23.92	0.0000
cond_new	5.1306	1.0511	4.88	0.0000
stock_photo	1.0803	1.0568	1.02	0.3085
duration	-0.0268	0.1904	-0.14	0.8882
wheels	7.2852	0.5547	13.13	0.0000

Example 14.2.1

The coefficient of cond new has a t test statistic of T = 4.88 and a p-value for its corresponding hypotheses ($H_0 : \beta_1 = 0, H_A : \beta_1 \neq 0$) of about zero. How can this be interpreted?

Solution

If we keep all the other variables in the model and add no others, then there is strong evidence that a game's condition (new or used) has a real relationship with the total auction price.

Example 14.2.2

Is there strong evidence that using a stock photo is related to the total auction price?

Solution

The t test statistic for stock photo is T = 1.02 and the p-value is about 0.31. After accounting for the other predictors, there is not strong evidence that using a stock photo in an auction is related to the total price of the auction. We might consider removing the stock photo variable from the model.

Exercise 14.2.1

Identify the p-values for both the duration and wheels variables in the model. Is there strong evidence supporting the connection of these variables with the total price in the model?

Answer

The p-value for the auction duration is 0.8882, which indicates that there is not statistically significant evidence that the duration is related to the total auction price when accounting for the other variables. The p-value for the Wii wheels variable is about zero, indicating that this variable is associated with the total auction price.

There is not statistically significant evidence that either the stock photo or duration variables contribute meaningfully to the model. Next we consider common strategies for pruning such variables from a model.

TIP: Using adjusted R^2 instead of p-values for model selection

The adjusted R^2 may be used as an alternative to p-values for model selection, where a higher adjusted R^2 represents a better model t. For instance, we could compare two models using their adjusted R^2, and the model with the higher adjusted R^2 would be preferred. This approach tends to include more variables in the final model when compared to the p-value approach.

Two model selection strategies

Two common strategies for adding or removing variables in a multiple regression model are called backward-selection and forward-selection. These techniques are often referred to as **stepwise** model selection strategies, because they add or delete one variable at a time as they "step" through the candidate predictors. We will discuss these strategies in the context of the p-value approach. Alternatively, we could have employed an R^2_{adj} approach.

The **backward-elimination** strategy starts with the model that includes all potential predictor variables. Variables are eliminated one-at-a-time from the model until only variables with statistically significant p-values remain. The strategy within each elimination step is to drop the variable with the largest p-value, re t the model, and reassess the inclusion of all variables.

Example 14.2.3

Results corresponding to the full model for the mario kart data are shown in Table 8.6. How should we proceed under the backward-elimination strategy?

Solution

There are two variables with coefficients that are not statistically different from zero: stock_photo and duration. We first drop the duration variable since it has a larger corresponding p-value, then we re t the model. A regression summary for the new model is shown in Table 8.7.

In the new model, there is not strong evidence that the coefficient for stock photo is different from zero, even though the p-value decreased slightly, and the other p-values remain very small. Next, we again eliminate the variable with the largest non-significant p-value, stock photo, and re t the model. The updated regression summary is shown in Table 8.8.

In the latest model, we see that the two remaining predictors have statistically significant coefficients with p-values of about zero. Since there are no variables remaining that could be eliminated from the model, we stop. The final model includes only the cond_new and wheels variables in predicting the total auction price:

$$\hat{y} = b_0 + b_1 x_1 + b_4 x_4 \quad (14.2.1)$$
$$= 36.78 + 5.58 x_1 + 7.23 x_4 \quad (14.2.2)$$

where x_1 represents cond new and x4 represents wheels.

An alternative to using p-values in model selection is to use the adjusted R^2. At each elimination step, we refit the model without each of the variables up for potential elimination. For example, in the first step, we would fit four models, where each would be missing a different predictor. If one of these smaller models has a higher adjusted R^2 than our current model, we pick the smaller model with the largest adjusted R^2. We continue in this way until removing variables does not increase R^2_{adj}. Had we used the adjusted R^2 criteria, we would have kept the stock photo variable along with the cond new and wheels variables.

Notice that the p-value for stock photo changed a little from the full model (0.309) to the model that did not include the duration variable (0.275). It is common for p-values of one variable to change, due to collinearity, after eliminating a different variable. This fluctuation emphasizes the importance of refitting a model after each variable elimination step. The p-values tend to change dramatically when the eliminated variable is highly correlated with another variable in the model.

The **forward-selection** strategy is the reverse of the backward-elimination technique. Instead of eliminating variables one-at-a-time, we add variables one-at-a-time until we cannot nd any variables that present strong evidence of their importance in the model.

Table 8.7: The output for the regression model where price is the outcome and the duration variable has been eliminated from the model.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	36.0483	0.9745	36.99	0.0000
cond_new	5.1763	0.9961	5.20	0.0000
stock_photo	1.1177	1.0192	1.10	0.2747
wheels	7.2984	0.5448	13.40	0.0000

Table 8.8: The output for the regression model where price is the outcome and the duration and stock photo variables have been eliminated from the model.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	36.7849	0.7066	52.06	0.0000
cond_new	5.5848	0.9245	6.04	0.0000
wheels	7.2328	0.5419	13.35	0.0000

Example 14.2.4: forward selection strategy

Construct a model for the mario kart data set using the forward selection strategy.

Solution

We start with the model that includes no variables. Then we t each of the possible models with just one variable. That is, we fit the model including just the cond new predictor, then the model including just the stock photo variable, then a model with just duration, and a model with just wheels. Each of the four models (yes, we fit four models!) provides a p-value for the coefficient of the predictor variable. Out of these four variables, the wheels variable had the smallest p-value. Since its p-value is less than 0.05 (the p-value was smaller than 2e-16), we add the Wii wheels variable to the model. Once a variable is added in forward-selection, it will be included in all models considered as well as the nal model.

Since we successfully found a first variable to add, we consider adding another. We fit three new models: (1) the model including just the cond_new and wheels variables (output in Table 8.8), (2) the model including just the stock photo and wheels variables, and (3) the model including only the duration and wheels variables. Of these models, the first had the lowest p-value for its new variable (the p-value corresponding to cond new was 1.4e-08). Because this p-value is below 0.05, we add the cond_new variable to the model. Now the final model is guaranteed to include both the condition and wheels variables.

We must then repeat the process a third time, fitting two new models: (1) the model including the stock photo, cond_new, and wheels variables (output in Table 8.7) and (2) the model including the duration, cond new, and wheels variables. The p-value corresponding to stock photo in the first model (0.275) was smaller than the p-value corresponding to duration in the second model (0.682). However, since this smaller p-value was not below 0.05, there was not strong evidence that it should be included in the model. Therefore, neither variable is added and we are finished.

The final model is the same as that arrived at using the backward-selection strategy.

Example 14.2.5: backward-selection strategy

As before, we could have used the R^2_{adj} criteria instead of examining p-values in selecting variables for the model. Rather than look for variables with the smallest p-value, we look for the model with the largest R^2_{adj}. What would the result of forward-selection be using the adjusted R^2 approach?

Solution

Using the forward-selection strategy, we start with the model with no predictors. Next we look at each model with a single predictor. If one of these models has a larger R^2_{adj} than the model with no variables, we use this new model. We repeat this procedure, adding one variable at a time, until we cannot nd a model with a larger R^2_{adj}. If we had done the forward-selection strategy using R^2_{adj}, we would have arrived at the model including cond new, stock photo, and wheels, which is a slightly larger

model than we arrived at using the p-value approach and the same model we arrived at using the adjusted R^2 and backwards-elimination.

Model selection strategies

The backward-elimination strategy begins with the largest model and eliminates variables one-by-one until we are satis ed that all remaining variables are important to the model. The forward-selection strategy starts with no variables included in the model, then it adds in variables according to their importance until no other important variables are found.

There is no guarantee that the backward-elimination and forward-selection strategies will arrive at the same nal model using the p-value or adjusted R^2 methods. If the backwards-elimination and forward-selection strategies are both tried and they arrive at different models, choose the model with the larger R^2_{adj} as a tie-breaker; other tie-break options exist but are beyond the scope of this book.

It is generally acceptable to use just one strategy, usually backward-elimination with either the p-value or adjusted R^2 criteria. However, before reporting the model results, we must verify the model conditions are reasonable.

This page titled 14.2: Model Selection is shared under a CC BY-SA 3.0 license and was authored, remixed, and/or curated by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel via source content that was edited to the style and standards of the LibreTexts platform.

- **8.2: Model Selection** by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel is licensed CC BY-SA 3.0. Original source: https://www.openintro.org/book/os.

14.3: Checking Model Assumptions using Graphs

Multiple regression methods using the model

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k \tag{14.3.1}$$

generally depend on the following four assumptions:

1. the residuals of the model are nearly normal,
2. the variability of the residuals is nearly constant,
3. the residuals are independent, and
4. each variable is linearly related to the outcome.

Simple and effective plots can be used to check each of these assumptions. We will consider the model for the auction data that uses the game condition and number of wheels as predictors.

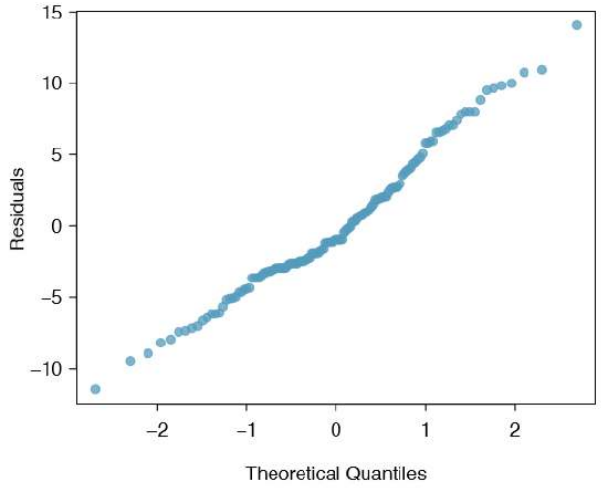


Figure **14.3.1**: A normal probability plot of the residuals is helpful in identifying observations that might be outliers.

Normal probability plot. A normal probability plot of the residuals is shown in Figure **14.3.1**. While the plot exhibits some minor irregularities, there are no outliers that might be cause for concern. In a normal probability plot for residuals, we tend to be most worried about residuals that appear to be outliers, since these indicate long tails in the distribution of residuals.

Absolute values of residuals against fitted values. A plot of the absolute value of the residuals against their corresponding fitted values ($\hat{y}_i$) is shown in Figure **14.3.2**. This plot is helpful to check the condition that the variance of the residuals is approximately constant. We do not see any obvious deviations from constant variance in this example.

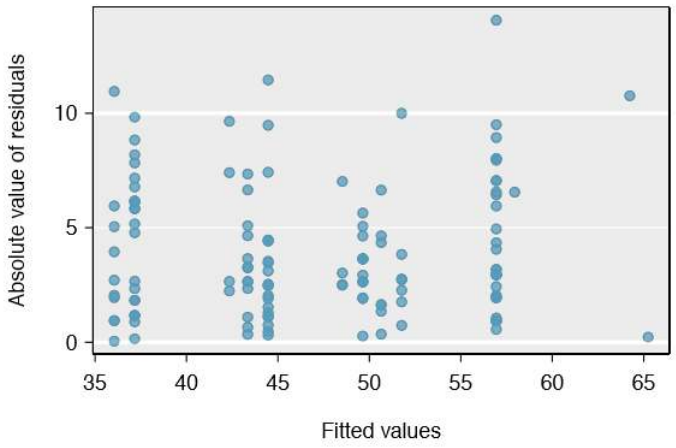


Figure **14.3.2**: Comparing the absolute value of the residuals against the fitted values ($\hat{y}_i$) is helpful in identifying deviations from the constant variance assumption.

Residuals in order of their data collection. A plot of the residuals in the order their corresponding auctions were observed is shown in Figure **14.3.3**. Such a plot is helpful in identifying any connection between cases that are close to one another, e.g. we could look for declining prices over time or if there was a time of the day when auctions tended to fetch a higher price. Here we see no structure that indicates a problem.[12]

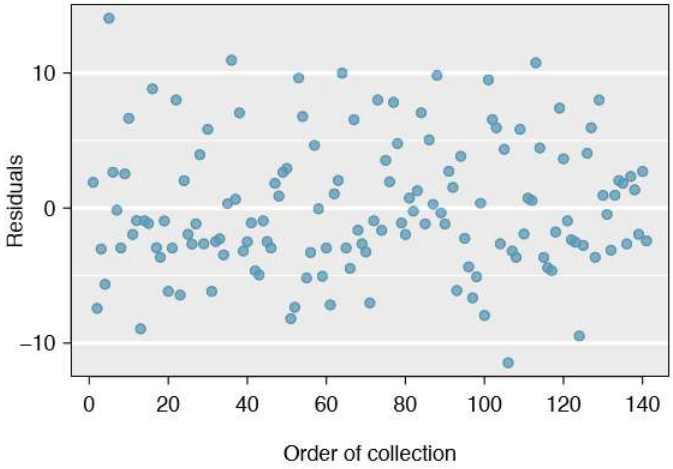


Figure **14.3.3**: Plotting residuals in the order that their corresponding observations were collected helps identify connections between successive observations. If it seems that consecutive observations tend to be close to each other, this indicates the independence assumption of the observations would fail.

Residuals against each predictor variable. We consider a plot of the residuals against the cond_new variable and the residuals against the wheels variable. These plots are shown in Figure **14.3.4**. For the two-level condition variable, we are guaranteed not to see any remaining trend, and instead we are checking that the variability does not fluctuate across groups. In this example, when we consider the residuals against the wheels variable, we see some possible structure. There appears to be curvature in the residuals, indicating the relationship is probably not linear.

[12]*An especially rigorous check would use time series methods. For instance, we could check whether consecutive residuals are correlated. Doing so with these residuals yields no statistically significant correlations.*

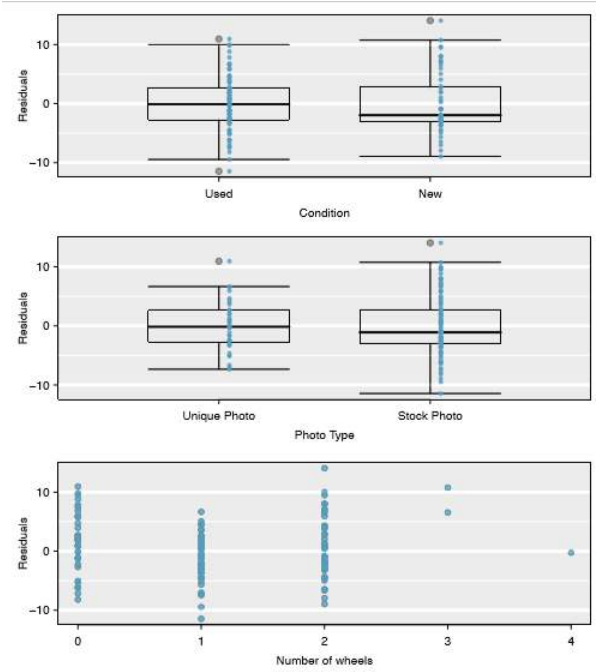


Figure **14.3.4**: In the two-level variable for the game's condition, we check for differences in distribution shape or variability. For numerical predictors, we also check for trends or other structure. We see some slight bowing in the residuals against the wheels variable.

It is necessary to summarize diagnostics for any model fit. If the diagnostics support the model assumptions, this would improve credibility in the ndings. If the diagnostic assessment shows remaining underlying structure in the residuals, we should try to adjust the model to account for that structure. If we are unable to do so, we may still report the model but must also note its shortcomings. In the case of the auction data, we report that there may be a nonlinear relationship between the total price and the number of wheels included for an auction. This information would be important to buyers and sellers; omitting this information could be a setback to the very people who the model might assist.

"All models are wrong, but some are useful" -George E.P. Box

The truth is that no model is perfect. However, even imperfect models can be useful. Reporting a awed model can be reasonable so long as we are clear and report the model's shortcomings.

Caution: do not report results when assumptions are grossly violated

While there is a little leeway in model assumptions, do not go too far. If model assumptions are very clearly violated, consider a new model, even if it means learning more statistical methods or hiring someone who can help.

TIP: Confidence intervals in multiple regression

Confidence intervals for coefficients in multiple regression can be computed using the same formula as in the single predictor model:

$$b_i \pm t^*_{df} SE_{b_i} \tag{14.3.2}$$

where t^*_{df} is the appropriate t value corresponding to the confidence level and model degrees of freedom, $df = n - k - 1$.

This page titled 14.3: Checking Model Assumptions using Graphs is shared under a CC BY-SA 3.0 license and was authored, remixed, and/or curated by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel via source content that was edited to the style and standards of the LibreTexts platform.

- **8.3: Checking Model Assumptions using Graphs** by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel is licensed CC BY-SA 3.0. Original source: https://www.openintro.org/book/os.

14.4: Introduction to Logistic Regression

In this section we introduce **logistic regression** as a tool for building models when there is a categorical response variable with two levels. Logistic regression is a type of **generalized linear model** (GLM) for response variables where regular multiple regression does not work very well. In particular, the response variable in these settings often takes a form where residuals look completely different from the normal distribution.

GLMs can be thought of as a two-stage modeling approach. We first model the response variable using a probability distribution, such as the binomial or Poisson distribution. Second, we model the parameter of the distribution using a collection of predictors and a special form of multiple regression.

In Section 8.4 we will revisit the email data set from Chapter 1. These emails were collected from a single email account, and we will work on developing a basic spam filter using these data. The response variable, spam, has been encoded to take value 0 when a message is not spam and 1 when it is spam. Our task will be to build an appropriate model that classi es messages as spam or not spam using email characteristics coded as predictor variables. While this model will not be the same as those used in large-scale spam filters, it shares many of the same features.

Table **14.4.1**: Descriptions for 11 variables in the email data set. Notice that all of the variables are indicator variables, which take the value 1 if the specified characteristic is present and 0 otherwise.

variable	*description*
spam	Specifies whether the message was spam.
to_multiple	An indicator variable for if more than one person was listed in the To field of the email.
cc	An indicator for if someone was CCed on the email.
attach	An indicator for if there was an attachment, such as a document or image.
dollar	An indicator for if the word "dollar" or dollar symbol ($) appeared in the email.
winner	An indicator for if the word "winner" appeared in the email message.
inherit	An indicator for if the word "inherit" (or a variation, like "inheritance") appeared in the email.
password	An indicator for if the word "password" was present in the email.
format	Indicates if the email contained special formatting, such as bolding, tables, or links
re_subj	Indicates whether "Re:" was included at the the start of the email subject.
exclaim_subj	Indicates whether any exclamation point was included in the email subject.

Email data

The email data set was first presented in Chapter 1 with a relatively small number of variables. In fact, there are many more variables available that might be useful for classifying spam. Descriptions of these variables are presented in Table **14.4.1**. The spam variable will be the outcome, and the other 10 variables will be the model predictors. While we have limited the predictors used in this section to be categorical variables (where many are represented as indicator variables), numerical predictors may also be used in logistic regression. See the footnote for an additional discussion on this topic.[13]

Modeling the probability of an event

TIP: Notation for a logistic regression model

The outcome variable for a GLM is denoted by Y_i, where the index i is used to represent observation i. In the email application, Y_i will be used to represent whether email i is spam ($Y_i = 1$) or not ($Y_i = 0$). The predictor variables are represented as follows: $x_{1;i}$ is the value of variable 1 for observation i, $x_{2;i}$ is the value of variable 2 for observation i, and so on.

Logistic regression is a generalized linear model where the outcome is a two-level categorical variable. The outcome, Y_i, takes the value 1 (in our application, this represents a spam message) with probability p_i and the value 0 with probability $1 - p_i$. It is the probability pi that we model in relation to the predictor variables.

[13]*Recall from Chapter 7 that if outliers are present in predictor variables, the corresponding observations may be especially influential on the resulting model. This is the motivation for omitting the numerical variables, such as the number of characters and line breaks in emails, that we saw in Chapter 1. These variables exhibited extreme skew. We could resolve this issue by transforming these variables (e.g. using a log-transformation), but we will omit this further investigation for brevity.*

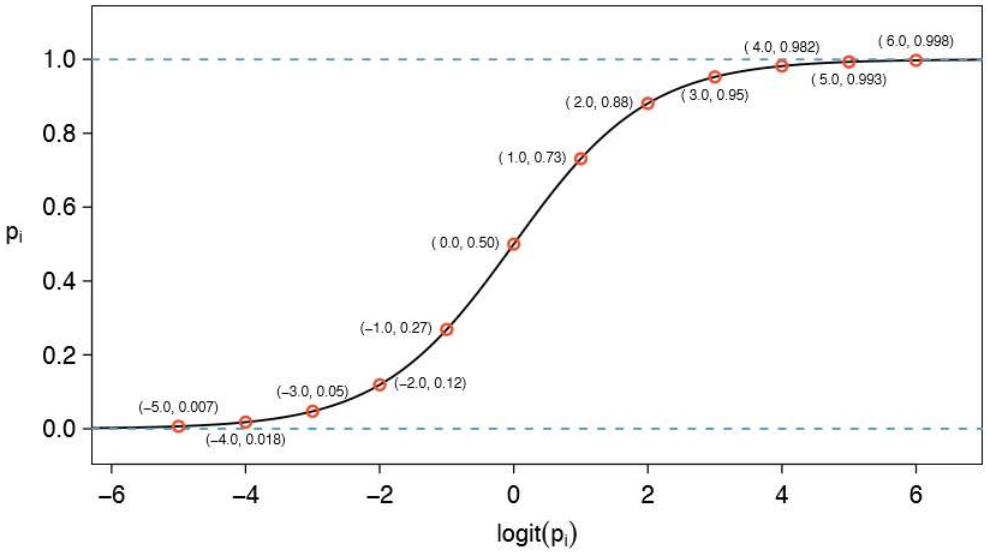


Figure **14.4.1**: Values of pi against values of logit(p_i).

The logistic regression model relates the probability an email is spam (p_i) to the predictors $x_{1;i}, x_{2;i}, \dots, x_{k;i}$ through a framework much like that of multiple regression:

$$\text{transformation}(\text{pi}) = \beta_0 + \beta_1 x_{1;i} + \beta_2 x_{2;i} + \dots \beta_k x_{k;i} \tag{14.4.1}$$

We want to choose a transformation in Equation 14.4.1 that makes practical and mathematical sense. For example, we want a transformation that makes the range of possibilities on the left hand side of Equation 14.4.1 equal to the range of possibilities for the right hand side; if there was no transformation for this equation, the left hand side could only take values between 0 and 1, but the right hand side could take values outside of this range. A common transformation for p_i is the **logit transformation**, which may be written as

$$logit(p_i) = log_e(\frac{p_i}{1 - p_i}) \tag{14.4.2}$$

The logit transformation is shown in Figure 8.14. Below, we rewrite Equation 14.4.1 using the logit transformation of p_i:

$$log_e(\frac{p_i}{1 - p_i}) = \beta_0 + \beta_1 x_{1;i} + \beta_2 x_{2;i} + \dots + \beta_k x_{k;i} \tag{14.4.3}$$

In our spam example, there are 10 predictor variables, so k = 10. This model isn't very intuitive, but it still has some resemblance to multiple regression, and we can t this model using software. In fact, once we look at results from software, it will start to feel like

we're back in multiple regression, even if the interpretation of the coefficients is more complex.

Example 14.4.1

Here we create a spam lter with a single predictor: to_multiple. This variable indicates whether more than one email address was listed in the To field of the email. The following logistic regression model was fit using statistical software:

'_' allowed only in math mode

If an email is randomly selected and it has just one address in the T_o field, what is the probability it is spam? What if more than one address is listed in the T_o field?

Solution

If there is only one email in the T_o field, then to multiple takes value 0 and the right side of the model equation equals -2.12. Solving for p_i : $\frac{e^{2.12}}{1+e^{-2.12}} = 0.11$. Just as we labeled a tted value of y_i with a "hat" in single-variable and multiple regression, we will do the same for this probability: $\hat{p}_i = 0.11$.

If there is more than one address listed in the T_o field, then the right side of the model equation is $-2.12 - 1.81 \times 1 = -3.93$, which corresponds to a probability $\hat{p}_i = 0.02$. Notice that we could examine -2.12 and -3.93 in Figure 8.14 to estimate the probability before formally calculating the value.

To convert from values on the regression-scale (e.g. -2.12 and -3.93 in Example 8.20), use the following formula, which is the result of solving for p_i in the regression model:

$$p_i = \frac{e^{\beta_0+\beta_1 x_{1,i}+\cdots+\beta_k x_{k,i}}}{1+e^{\beta_0+\beta_1 x_{1,i}+\cdots+\beta_k x_{k,i}}} \tag{14.4.4}$$

As with most applied data problems, we substitute the point estimates for the parameters (the β_i) so that we may make use of this formula. In Example **14.4.1**, the probabilities were calculated as

$$\frac{e^{-2.12}}{1+e^{-2.12}} = 0.11 \frac{e^{-2.12-1.81}}{1+e^{-2.12-1.81}} = 0.02 \tag{14.4.5}$$

While the information about whether the email is addressed to multiple people is a helpful start in classifying email as spam or not, the probabilities of 11% and 2% are not dramatically different, and neither provides very strong evidence about which particular email messages are spam. To get more precise estimates, we'll need to include many more variables in the model.

We used statistical software to fit the logistic regression model with all ten predictors described in Table 8.13. Like multiple regression, the result may be presented in a summary table, which is shown in Table **14.4.2**. The structure of this table is almost identical to that of multiple regression; the only notable difference is that the p-values are calculated using the normal distribution rather than the t distribution.

Just like multiple regression, we could trim some variables from the model using the p-value. Using backwards elimination with a p-value cutoff of 0.05 (start with the full model and trim the predictors with p-values greater than 0.05), we ultimately eliminate the exclaim_subj, dollar, inherit, and cc predictors. The remainder of this section will rely on this smaller model, which is summarized in Table **14.4.3**.

Exercise 14.4.1

Examine the summary of the reduced model in Table **14.4.3**, and in particular, examine the to_multiple row. Is the point estimate the same as we found before, -1.81, or is it different? Explain why this might be.

Solution

The new estimate is different: -2.87. This new value represents the estimated coefficient when we are also accounting for other variables in the logistic regression model.

Table **14.4.2**: Summary table for the full logistic regression model for the spam lter example.

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.8362	0.0962	-8.69	0.0000
to multiple	-2.8836	0.3121	-9.24	0.0000
winner	1.7038	0.3254	5.24	0.0000
format	-1.5902	0.1239	-12.84	0.0000
re_subj	-2.9082	0.3708	-7.84	0.0000
exclaim_subj	0.1355	0.2268	0.60	0.5503
cc	-0.4863	0.3054	-1.59	0.1113
attach	0.9790	0.2170	4.51	0.0000
dollar	-0.0582	0.1589	-0.37	0.7144
inherit	0.2093	0.3197	0.65	0.5127
password	-1.4929	0.5295	-2.82	0.0048

Table **14.4.3**: Summary table for the logistic regression model for the spam lter, where variable selection has been performed.

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.8595	0.0910	-9.44	0.0000
to multiple	-2.8836	0.3092	-9.18	0.0000
winner	1.7370	0.3218	5.40	0.0000
format	-1.5569	0.1207	-12.90	0.0000
re_subj	-3.0482	0.3630	-8.40	0.0000
attach	0.8643	0.2042	4.23	0.0000
password	-1.4871	0.5290	-2.81	0.0049

Point estimates will generally change a little - and sometimes a lot - depending on which other variables are included in the model. This is usually due to colinearity in the predictor variables. We previously saw this in the Ebay auction example when we compared the coefficient of cond new in a single-variable model and the corresponding coefficient in the multiple regression model that used three additional variables (see Sections 8.1.1 and 8.1.2).

Example 14.4.2

Spam lters are built to be automated, meaning a piece of software is written to collect information about emails as they arrive, and this information is put in the form of variables. These variables are then put into an algorithm that uses a statistical model, like the one we've t, to classify the email. Suppose we write software for a spam lter using the reduced model shown in Table **14.4.3**. If an incoming email has the word "winner" in it, will this raise or lower the model's calculated probability that the incoming email is spam?

Solution

The estimated coefficient of winner is positive (1.7370). A positive coefficient estimate in logistic regression, just like in multiple regression, corresponds to a positive association between the predictor and response variables when accounting for the other variables in the model. Since the response variable takes value 1 if an email is spam and 0 otherwise, the positive coefficient indicates that the presence of "winner" in an email raises the model probability that the message is spam.

Example 14.4.3

Suppose the same email from Example **14.4.2** was in HTML format, meaning the format variable took value 1. Does this characteristic increase or decrease the probability that the email is spam according to the model?

Solution

Since HTML corresponds to a value of 1 in the format variable and the coefficient of this variable is negative (-1.5569), this would lower the probability estimate returned from the model.

Practical decisions in the email application

Examples 8.22 and 8.23 highlight a key feature of logistic and multiple regression. In the spam lter example, some email characteristics will push an email's classification in the direction of spam while other characteristics will push it in the opposite direction. If we were to implement a spam filter using the model we have fit, then each future email we analyze would fall into one of three categories based on the email's characteristics:

1. The email characteristics generally indicate the email is not spam, and so the resulting probability that the email is spam is quite low, say, under 0.05.
2. The characteristics generally indicate the email is spam, and so the resulting probability that the email is spam is quite large, say, over 0.95.
3. The characteristics roughly balance each other out in terms of evidence for and against the message being classified as spam. Its probability falls in the remaining range, meaning the email cannot be adequately classified as spam or not spam.

If we were managing an email service, we would have to think about what should be done in each of these three instances. In an email application, there are usually just two possibilities: filter the email out from the regular inbox and put it in a "spambox", or let the email go to the regular inbox.

Exercise 14.4.2

The first and second scenarios are intuitive. If the evidence strongly suggests a message is not spam, send it to the inbox. If the evidence strongly suggests the message is spam, send it to the spambox. How should we handle emails in the third category?

Solution

In this particular application, we should err on the side of sending more mail to the inbox rather than mistakenly putting good messages in the spambox. So, in summary: emails in the first and last categories go to the regular inbox, and those in the second scenario go to the spambox.

Exercise 14.4.3

Suppose we apply the logistic model we have built as a spam filter and that 100 messages are placed in the spambox over 3 months. If we used the guidelines above for putting messages into the spambox, about how many legitimate (non-spam) messages would you expect to find among the 100 messages?

Solution

First, note that we proposed a cutoff for the predicted probability of 0.95 for spam. In a worst case scenario, all the messages in the spambox had the minimum probability equal to about 0.95. Thus, we should expect to nd about 5 or fewer legitimate messages among the 100 messages placed in the spambox.

Almost any classifier will have some error. In the spam lter guidelines above, we have decided that it is okay to allow up to 5% of the messages in the spambox to be real messages. If we wanted to make it a little harder to classify messages as spam, we could use a cutoff of 0.99. This would have two effects. Because it raises the standard for what can be classified as spam, it reduces the number of good emails that are classified as spam.

However, it will also fail to correctly classify an increased fraction of spam messages. No matter the complexity and the confidence we might have in our model, these practical considerations are absolutely crucial to making a helpful spam filter. Without them, we could actually do more harm than good by using our statistical model.

Diagnostics for the email classifier

Logistic regression conditions

There are two key conditions for fitting a logistic regression model:

1. The model relating the parameter p_i to the predictors $x_{1;i}, x_{2;i}, \dots, x_{k;i}$ closely resembles the true relationship between the parameter and the predictors.

2. Each outcome Y_i is independent of the other outcomes.

The first condition of the logistic regression model is not easily checked without a fairly sizable amount of data. Luckily, we have 3,921 emails in our data set! Let's first visualize these data by plotting the true classification of the emails against the model's fitted probabilities, as shown in Figure **14.4.2**. The vast majority of emails (spam or not) still have fitted probabilities below 0.5.

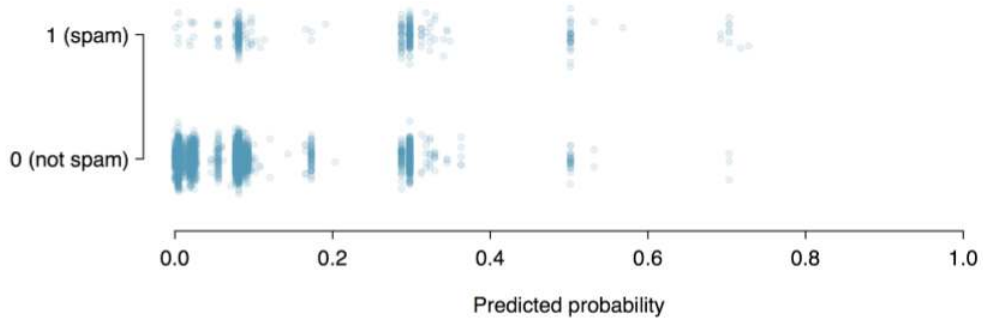


Figure **14.4.2**: The predicted probability that each of the 3,912 emails is spam is classified by their grouping, spam or not. Noise (small, random vertical shifts) have been added to each point so that points with nearly identical values aren't plotted exactly on top of one another. This makes it possible to see more observations.

This may at first seem very discouraging: we have t a logistic model to create a spam filter, but no emails have a fitted probability of being spam above 0.75. Don't despair; we will discuss ways to improve the model through the use of better variables in Section 8.4.5.

We'd like to assess the quality of our model. For example, we might ask: if we look at emails that we modeled as having a 10% chance of being spam, do we nd about 10% of them actually are spam? To help us out, we'll borrow an advanced statistical method called **natural splines** that estimates the local probability over the region 0.00 to 0.75 (the largest predicted probability was 0.73, so we avoid extrapolating). All you need to know about natural splines to understand what we are doing is that they are used to fit flexible lines rather than straight lines.

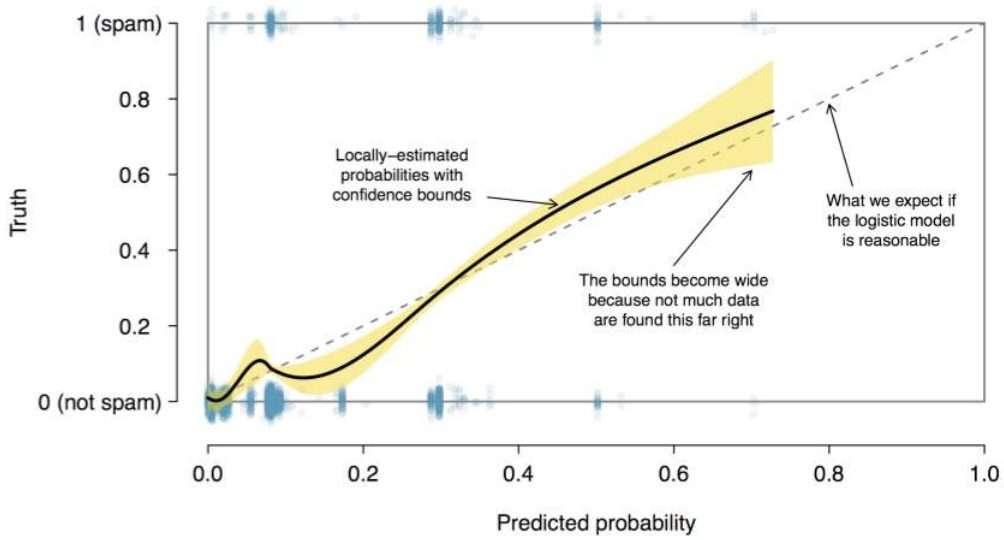


Figure **14.4.3**: The solid black line provides the empirical estimate of the probability for observations based on their predicted probabilities (confidence bounds are also shown for this line), which is t using natural splines. A small amount of noise was added to the observations in the plot to allow more observations to be seen.

The curve fit using natural splines is shown in Figure **14.4.3** as a solid black line. If the logistic model fits well, the curve should closely follow the dashed $y = x$ line. We have added shading to represent the confidence bound for the curved line to clarify what

fluctuations might plausibly be due to chance. Even with this confidence bound, there are weaknesses in the first model assumption. The solid curve and its confidence bound dips below the dashed line from about 0.1 to 0.3, and then it drifts above the dashed line from about 0.35 to 0.55. These deviations indicate the model relating the parameter to the predictors does not closely resemble the true relationship.

We could evaluate the second logistic regression model assumption - independence of the outcomes - using the model residuals. The residuals for a logistic regression model are calculated the same way as with multiple regression: the observed outcome minus the expected outcome. For logistic regression, the expected value of the outcome is the fitted probability for the observation, and the residual may be written as

$$e_i = Y_i - \hat{p}_i \tag{14.4.6}$$

We could plot these residuals against a variety of variables or in their order of collection, as we did with the residuals in multiple regression. However, since we know the model will need to be revised to effective classify spam and you have already seen similar residual plots in Section 8.3, we won't investigate the residuals here.

Improving the set of variables for a spam filter

If we were building a spam filter for an email service that managed many accounts (e.g. Gmail or Hotmail), we would spend much more time thinking about additional variables that could be useful in classifying emails as spam or not. We also would use transformations or other techniques that would help us include strongly skewed numerical variables as predictors.

Take a few minutes to think about additional variables that might be useful in identifying spam. Below is a list of variables we think might be useful:

1. An indicator variable could be used to represent whether there was prior two-way correspondence with a message's sender. For instance, if you sent a message to john@example.com and then John sent you an email, this variable would take value 1 for the email that John sent. If you had never sent John an email, then the variable would be set to 0.
2. A second indicator variable could utilize an account's past spam flagging information. The variable could take value 1 if the sender of the message has previously sent messages flagged as spam.
3. A third indicator variable could flag emails that contain links included in previous spam messages. If such a link is found, then set the variable to 1 for the email. otherwise, set it to 0.

The variables described above take one of two approaches. Variable (1) is specially designed to capitalize on the fact that spam is rarely sent between individuals that have two-way communication. Variables (2) and (3) are specially designed to flag common spammers or spam messages. While we would have to verify using the data that each of the variables is effective, these seem like promising ideas.

Table **14.4.4** shows a contingency table for spam and also for the new variable described in (1) above. If we look at the 1,090 emails where there was correspondence with the sender in the preceding 30 days, not one of these message was spam. This suggests variable (1) would be very effective at accurately classifying some messages as not spam. With this single variable, we would be able to send about 28% of messages through to the inbox with confidence that almost none are spam.

Table **14.4.4**: A contingency table for spam and a new variable that represents whether there had been correspondence with the sender in the preceding 30 days.

	prior	correspondence	
	no	yes	Total
spam	367	0	367
not spam	2464	1090	3554
Total	2831	1090	3921

The variables described in (2) and (3) would provide an excellent foundation for distinguishing messages coming from known spammers or messages that take a known form of spam. To utilize these variables, we would need to build databases: one holding email addresses of known spammers, and one holding URLs found in known spam messages. Our access to such information is

limited, so we cannot implement these two variables in this textbook. However, if we were hired by an email service to build a spam filter, these would be important next steps.

In addition to finding more and better predictors, we would need to create a customized logistic regression model for each email account. This may sound like an intimidating task, but its complexity is not as daunting as it may at first seem. We'll save the details for a statistics course where computer programming plays a more central role. For what is the extremely challenging task of classifying spam messages, we have made a lot of progress. We have seen that simple email variables, such as the format, inclusion of certain words, and other circumstantial characteristics, provide helpful information for spam classi cation. Many challenges remain, from better understanding logistic regression to carrying out the necessary computer programming, but completing such a task is very nearly within your reach.

This page titled 14.4: Introduction to Logistic Regression is shared under a CC BY-SA 3.0 license and was authored, remixed, and/or curated by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel via source content that was edited to the style and standards of the LibreTexts platform.

- **8.4: Introduction to Logistic Regression** by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel is licensed CC BY-SA 3.0. Original source: https://www.openintro.org/book/os.

14.5: Exercises

Introduction to multiple regression

8.1 Baby weights, Part I. The Child Health and Development Studies investigate a range of topics. One study considered all pregnancies between 1960 and 1967 among women in the Kaiser Foundation Health Plan in the San Francisco East Bay area. Here, we study the relationship between smoking and weight of the baby. The variable smoke is coded 1 if the mother is a smoker, and 0 if not. The summary table below shows the results of a linear regression model for predicting the average birth weight of babies, measured in ounces, based on the smoking status of the mother.[17]

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	123.05	0.65	189.60	0.0000
smoke	-8.94	1.03	-8.65	0.0000

The variability within the smokers and non-smokers are about equal and the distributions are symmetric. With these conditions satisfied, it is reasonable to apply the model. (Note that we don't need to check linearity since the predictor has only two levels.)

1. (a) Write the equation of the regression line.
2. (b) Interpret the slope in this context, and calculate the predicted birth weight of babies born to smoker and non-smoker mothers.
3. (c) Is there a statistically signi cant relationship between the average birth weight and smoking?

8.2 Baby weights, Part II. Exercise 8.1 introduces a data set on birth weight of babies. Another variable we consider is parity, which is 0 if the child is the first born, and 1 otherwise. The summary table below shows the results of a linear regression model for predicting the average birth weight of babies, measured in ounces, from parity.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	120.07	0.60	199.94	0.0000
smoke	-1.93	1.19	-1.62	0.1052

1. (a) Write the equation of the regression line.
2. (b) Interpret the slope in this context, and calculate the predicted birth weight of first borns and others.
3. (c) Is there a statistically signi cant relationship between the average birth weight and parity?

[17] *Child Health and Development Studies, Baby weights data set.*

8.3 Baby weights, Part III. We considered the variables smoke and parity, one at a time, in modeling birth weights of babies in Exercises 8.1 and 8.2. A more realistic approach to modeling infant weights is to consider all possibly related variables at once. Other variables of interest include length of pregnancy in days (gestation), mother's age in years (age), mother's height in inches (height), and mother's pregnancy weight in pounds (weight). Below are three observations from this data set.

	bwt	gestation	parity	age	height	weight	smoke
1	120	284	0	27	62	100	0
2	113	282	0	33	64	135	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1236	117	297	0	38	65	129	0

The summary table below shows the results of a regression model for predicting the average birth weight of babies based on all of the variables included in the data set.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-80.41	14.35	-5.60	0.0000
gestation	0.44	0.03	15.26	0.0000
parity	-3.33	1.13	-2.95	0.0033
age	-0.01	0.09	-0.10	0.9170
height	1.15	0.21	5.63	0.0000
weight	0.05	0.03	1.99	0.0471
smoke	-8.40	0.95	-8.81	0.0000

1. (a) Write the equation of the regression line that includes all of the variables.
2. (b) Interpret the slopes of gestation and age in this context.
3. (c) The coefficient for parity is different than in the linear model shown in Exercise 8.2. Why might there be a difference?
4. (d) Calculate the residual for the rst observation in the data set.
5. (e) The variance of the residuals is 249.28, and the variance of the birth weights of all babies in the data set is 332.57. Calculate the R^2 and the adjusted R^2. Note that there are 1,236 observations in the data set.

8.4 Absenteeism. Researchers interested in the relationship between absenteeism from school and certain demographic characteristics of children collected data from 146 randomly sampled students in rural New SouthWales, Australia, in a particular school year. Below are three observations from this data set.

	eth	sex	lrn	days
1	0	1	1	2
2	0	1	1	11
⋮	⋮	⋮	⋮	⋮
146	1	0	0	37

The summary table below shows the results of a linear regression model for predicting the average number of days absent based on ethnic background (eth: 0 - aboriginal, 1 - not aboriginal), sex (sex: 0 - female, 1 - male), and learner status (lrn: 0 - average learner, 1 - slow learner).[18]

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	18.93	2.57	7.37	0.0000
eth	-9.11	2.60	-3.51	0.0000
sex	3.10	2.64	1.18	0.2411
lrn	2.15	2.65	0.81	0.4177

(a) Write the equation of the regression line.

(b) Interpret each one of the slopes in this context.

(c) Calculate the residual for the rst observation in the data set: a student who is aboriginal, male, a slow learner, and missed 2 days of school.

(d) The variance of the residuals is 240.57, and the variance of the number of absent days for all students in the data set is 264.17. Calculate the R^2 and the adjusted R^2. Note that there are 146 observations in the data set.

8.5 GPA. A survey of 55 Duke University students asked about their GPA, number of hours they study at night, number of nights they go out, and their gender. Summary output of the regression model is shown below. Note that male is coded as 1.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	3.45	0.35	9.85	0.00
studyweek	0.00	0.00	0.27	0.79
sleepnight	0.01	0.05	0.11	0.91
outnight	0.05	0.05	1.01	0.32
gender	-0.08	0.12	-0.68	0.50

(a) Calculate a 95% con dence interval for the coefficient of gender in the model, and interpret it in the context of the data.

(b) Would you expect a 95% con dence interval for the slope of the remaining variables to include 0? Explain

[18]*W. N. Venables and B. D. Ripley. Modern Applied Statistics with S. Fourth Edition. Data can also be found in the R MASS package. New York: Springer, 2002.*

8.6 Cherry trees. Timber yield is approximately equal to the volume of a tree, however, this value is difficult to measure without rst cutting the tree down. Instead, other variables, such as height and diameter, may be used to predict a tree's volume and yield. Researchers wanting to understand the relationship between these variables for black cherry trees collected data from 31 such trees in the Allegheny National Forest, Pennsylvania. Height is measured in feet, diameter in inches (at 54 inches above ground), and volume in cubic feet.[19]

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-57.99	8.64	-6.71	0.00
height	0.34	0.13	2.61	0.01
diameter	4.71	0.26	17.82	0.00

(a) Calculate a 95% con dence interval for the coefficient of height, and interpret it in the context of the data.

(b) One tree in this sample is 79 feet tall, has a diameter of 11.3 inches, and is 24.2 cubic feet in volume. Determine if the model overestimates or underestimates the volume of this tree, and by how much.

Model selection

8.7 Baby weights, Part IV. Exercise 8.3 considers a model that predicts a newborn's weight using several predictors. Use the regression table below, which summarizes the model, to answer the following questions. If necessary, refer back to Exercise 8.3 for a reminder about the meaning of each variable.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-80.41	14.35	-5.60	0.0000
gestation	0.44	0.03	15.26	0.0000
parity	-3.33	1.13	-2.95	0.0033
age	-0.01	0.09	-0.10	0.9170
height	1.15	0.21	5.63	0.0000
weight	0.05	0.03	1.99	0.0471
smoke	-8.40	0.95	-8.81	0.0000

(a) Determine which variables, if any, do not have a signi cant linear relationship with the outcome and should be candidates for removal from the model. If there is more than one such variable, indicate which one should be removed first.

(b) The summary table below shows the results of the model with the age variable removed. Determine if any other variable(s) should be removed from the model.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-80.64	14.04	-5.74	0.0000
gestation	0.44	0.03	15.28	0.0000
parity	-3.29	1.06	-3.10	0.0020
height	1.15	0.20	5.64	0.0000
weight	0.05	0.03	2.00	0.0459
smoke	-8.38	0.95	-8.82	0.0000

[19]*D.J. Hand. A handbook of small data sets. Chapman & Hall/CRC, 1994.*

8.8 Absenteeism, Part II. Exercise 8.4 considers a model that predicts the number of days absent using three predictors: ethnic background (eth), gender (sex), and learner status (lrn). Use the regression table below to answer the following questions. If necessary, refer back to Exercise 8.4 for additional details about each variable.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	18.93	2.57	7.37	0.0000
eth	-9.11	2.60	-3.51	0.0000
sex	3.10	2.64	1.18	0.2411
lrn	2.15	2.65	0.81	0.4177

(a) Determine which variables, if any, do not have a signi cant linear relationship with the outcome and should be candidates for removal from the model. If there is more than one such variable, indicate which one should be removed first.

(b) The summary table below shows the results of the regression we re t after removing learner status from the model. Determine if any other variable(s) should be removed from the model.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	19.98	2.22	9.01	0.0000
eth	-9.06	2.60	-3.49	0.0006
sex	2.78	2.60	1.07	0.2878

8.9 Baby weights, Part V. Exercise 8.3 provides regression output for the full model (including all explanatory variables available in the data set) for predicting birth weight of babies. In this exercise we consider a forward-selection algorithm and add variables to the model one-at-a-time. The table below shows the p-value and adjusted R^2 of each model where we include only the corresponding predictor. Based on this table, which variable should be added to the model first?

variable	gestation	parity	age	height	weight	smoke
p-value	2.2×10^{-16}	0.1052	0.2375	2.97×10^{-12}	8.2×10^{-8}	2.2×10^{-16}
R^2_{adj}	0.1657	0.0013	0.0003	0.0386	0.0229	0.0569

8.10 Absenteeism, Part III. Exercise 8.4 provides regression output for the full model, including all explanatory variables available in the data set, for predicting the number of days absent from school. In this exercise we consider a forward-selection algorithm and add variables to the model one-at-a-time. The table below shows the p-value and adjusted R^2 of each model where we include only the corresponding predictor. Based on this table, which variable should be added to the model first?

variable	ethnicity	sex	learner status
p-value	0.0007	0.3142	0.5870
R^2_{adj}	0.0714	0.0001	0

Checking model assumptions using graphs

8.11 Baby weights, Part V. Exercise 8.7 presents a regression model for predicting the average birth weight of babies based on length of gestation, parity, height, weight, and smoking status of the mother. Determine if the model assumptions are met using the plots below. If not, describe how to proceed with the analysis.

8.12 GPA and IQ. A regression model for predicting GPA from gender and IQ was fit, and both predictors were found to be statistically signi cant. Using the plots given below, determine if this regression model is appropriate for these data.

Logistic regression

8.13 Possum classi cation, Part I. The common brushtail possum of the Australia region is a bit cuter than its distant cousin, the American opossum (see Figure 7.5 on page 318). We consider 104 brushtail possums from two regions in Australia, where the possums may be considered a random sample from the population. The rst region is Victoria, which is in the eastern half of Australia and traverses the southern coast. The second region consists of New South Wales and Queensland, which make up eastern and northeastern Australia.

We use logistic regression to differentiate between possums in these two regions. The outcome variable, called population, takes value 1 when a possum is from Victoria and 0 when it is from New South Wales or Queensland. We consider ve predictors: sex male (an indicator for a possum being male), head length, skull width, total length, and tail length. Each variable is summarized in a histogram. The full logistic regression model and a reduced model after variable selection are summarized in the table.

Full Model

	Estimate	SE	Z	Pr(>|Z|)
(Intercept)	39.2349	11.5368	3.40	0.0007
sex male	-1.2376	0.6662	-1.86	0.0632
head length	-0.1601	0.1386	-1.16	0.2480
skull width	-0.2012	0.1327	-1.52	0.1294
total length	0.6488	0.1531	4.24	0.0000
tail length	-1.8708	0.3741	-5.00	0.0000

Reduced Model

	Estimate	SE	Z	Pr(>|Z|)
(Intercept)	33.5095	9.9053	3.38	0.0007
sex male	-1.4207	0.6457	-2.20	0.0278
head length				
skull width	-0.2787	0.1226	-2.27	0.0231
total length	0.5687	0.1322	4.30	0.0000
tail length	-1.8057	0.3599	-5.02	0.0000

(a) Examine each of the predictors. Are there any outliers that are likely to have a very large inuence on the logistic regression model?

(b) The summary table for the full model indicates that at least one variable should be eliminated when using the p-value approach for variable selection: head length. The second component of the table summarizes the reduced model following variable selection. Explain why the remaining estimates change between the two models.

8.14 Challenger disaster, Part I. On January 28, 1986, a routine launch was anticipated for the Challenger space shuttle. Seventy-three seconds into the ight, disaster happened: the shuttle broke apart, killing all seven crew members on board. An investigation into the cause of the disaster focused on a critical seal called an O-ring, and it is believed that damage to these O-rings during a shuttle launch may be related to the ambient temperature during the launch. The table below summarizes observational data on O-rings for 23 shuttle missions, where the mission order is based on the temperature at the time of the launch. Temp gives the

temperature in Fahrenheit, Damaged represents the number of damaged O-rings, and Undamaged represents the number of O-rings that were not damaged.

Shuttle Mission	1	2	3	4	5	6	7	8	9	10	11	12
Temperature	53	57	58	63	66	67	67	67	68	69	70	70
Damaged	5	1	1	1	0	0	0	0	0	0	1	0
Undamaged	1	5	5	5	6	6	6	6	6	6	5	6

Shuttle Mission	13	14	15	16	17	18	19	20	21	22	23
Temperature	70	70	72	73	75	75	76	76	78	79	81
Damaged	1	0	0	0	0	1	0	0	0	0	0
Undamaged	5	6	6	6	6	5	6	6	6	6	6

(a) Each column of the table above represents a different shuttle mission. Examine these data and describe what you observe with respect to the relationship between temperatures and damaged O-rings.

(b) Failures have been coded as 1 for a damaged O-ring and 0 for an undamaged O-ring, and a logistic regression model was t to these data. A summary of this model is given below. Describe the key components of this summary table in words.

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	11.6630	3.2963	3.54	0.0004
Temperature	-0.2162	0.0532	-4.07	0.0000

(c) Write out the logistic model using the point estimates of the model parameters.

(d) Based on the model, do you think concerns regarding O-rings are justi ed? Explain.

8.15 Possum classi cation, Part II. A logistic regression model was proposed for classifying common brushtail possums into their two regions in Exercise 8.13. Use the results of the summary table for the reduced model presented in Exercise 8.13 for the questions below. The outcome variable took value 1 if the possum was from Victoria and 0 otherwise.

(a) Write out the form of the model. Also identify which of the following variables are positively associated (when controlling for other variables) with a possum being from Victoria: skull width, total length, and tail length.

(b) Suppose we see a brushtail possum at a zoo in the US, and a sign says the possum had been captured in the wild in Australia, but it doesn't say which part of Australia. However, the sign does indicate that the possum is male, its skull is about 63 mm wide, its tail is 37 cm long, and its total length is 83 cm. What is the reduced model's computed probability that this possum is from Victoria? How confident are you in the model's accuracy of this probability calculation?

8.16 Challenger disaster, Part II. Exercise 8.14 introduced us to O-rings that were identified as a plausible explanation for the breakup of the Challenger space shuttle 73 seconds into takeo in 1986. The investigation found that the ambient temperature at the

time of the shuttle launch was closely related to the damage of O-rings, which are a critical component of the shuttle. See this earlier exercise if you would like to browse the original data.

(a) The data provided in the previous exercise are shown in the plot. The logistic model fit to these data may be written as

$$log(\frac{\hat{p}}{1-\hat{p}} = 11.6630 - 0.2162 \times \text{Temperature} \quad (14.5.1)$$

where $\hat{p}$ is the model-estimated probability that an O-ring will become damaged. Use the model to calculate the probability that an O-ring will become damaged at each of the following ambient temperatures: 51, 53, and 55 degrees Fahrenheit. The model-estimated probabilities for several additional ambient temperatures are provided below, where subscripts indicate the temperature:

$$\hat{p}_{57} = 0.341 \hat{p}_{59} = 0.251 \hat{p}_{61} = 0.179 \hat{p}_{63} = 0.124 \quad (14.5.2)$$

$$\hat{p}_{65} = 0.084 \hat{p}_{67} = 0.056 \hat{p}_{69} = 0.037 \hat{p}_{71} = 0.024 \quad (14.5.3)$$

(b) Add the model-estimated probabilities from part (a) on the plot, then connect these dots using a smooth curve to represent the model-estimated probabilities.

(c) Describe any concerns you may have regarding applying logistic regression in this application, and note any assumptions that are required to accept the model's validity.

Contributors

David M Diez (Google/YouTube), Christopher D Barr (Harvard School of Public Health), Mine Çetinkaya-Rundel (Duke University)

This page titled 14.5: Exercises is shared under a CC BY-SA 3.0 license and was authored, remixed, and/or curated by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel via source content that was edited to the style and standards of the LibreTexts platform.

- **8.E: Multiple and Logistic Regression (Exercises)** by David Diez, Christopher Barr, & Mine Çetinkaya-Rundel has no license indicated. Original source: https://www.openintro.org/book/os.

14.6: Statistical Literacy

Learning Objectives

- Regression Toward the Mean in American Football

In a discussion about the Dallas Cowboy football team, there was a comment that the quarterback threw far more interceptions in the first two games than is typical (there were two interceptions per game). The author correctly pointed out that, because of regression toward the mean, performance in the future is expected to improve. However, the author defined regression toward the mean as, "In nerd land, that basically means that things tend to even out over the long run."

Example 14.6.1: what do you think?

Comment on that definition.

Solution

That definition is sort of correct, but it could be stated more precisely. Things don't always tend to even out in the long run. If a great player has an average game, then things wouldn't even out (to the average of all players) but would regress toward that player's high mean performance.

This page titled 14.6: Statistical Literacy is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **14.9: Statistical Literacy** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

14.E: Regression (Exercises)

General Questions

Q1

What is the equation for a regression line? What does each term in the line refer to? (relevant section)

Q2

The formula for a regression equation based on a sample size of 25 observations is $Y' = 2X + 9$.

a. What would be the predicted score for a person scoring 6 on X?
b. If someone's predicted score was 14, what was this person's score on X? (relevant section)

Q3

What criterion is used for deciding which regression line fits best? (relevant section)

Q4

What does the standard error of the estimate measure? What is the formula for the standard error of the estimate? (relevant section)

Q5

a. In a regression analysis, the sum of squares for the predicted scores is 100 and the sum of squares error is 200, what is R^2?
b. In a different regression analysis, 40% of the variance was explained. The sum of squares total is 1000. What is the sum of squares of the predicted values? (relevant section)

Q6

For the X, Y data below, compute:

a. r and determine if it is significantly different from zero.
b. the slope of the regression line and test if it differs significantly from zero.
c. the 95% confidence interval for the slope.

(relevant section)

X	Y
2	5
4	6
4	7
5	11
6	12

Q7

What assumptions are needed to calculate the various inferential statistics of linear regression? (relevant section)

Q8

The correlation between years of education and salary in a sample of 20 people from a certain company is 0.4. Is this correlation statistically significant at the 0.05 level? (relevant section)

Q9

A sample of X and Y scores is taken, and a regression line is used to predict Y from X. If $SSY' = 300$, $SSE = 500$, and $N = 50$, what is: (relevant section relevant section)

a. SSY?

b. the standard error of the estimate?
c. R^2?

Q10

Using linear regression, find the predicted post-test score for someone with a score of **43** on the pre-test. (relevant section)

Pre	Post
59	56
52	63
44	55
51	50
42	66
42	48
41	58
45	36
27	13
63	50
54	81
44	56
50	64
47	50
55	63
49	57
45	73
57	63
46	46
60	60
65	47
64	73
50	58
74	85
59	44

Q11

The equation for a regression line predicting the number of hours of TV watched by children (Y) from the number of hours of TV watched by their parents (X) is $Y' = 4 + 1.2X$. The sample size is 12.

a. If the standard error of b is 0.4, is the slope statistically significant at the 0.05 level? (relevant section)
b. If the mean of X is 8, what is the mean of Y? (relevant section)

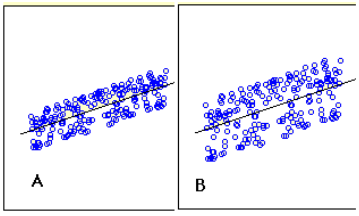
Q12

Based on the table below, compute the regression line that predicts Y from X. (relevant section)

MX	MY	sX	sY	r
10	12	2.5	3.0	-0.6

Q13

Does A or B have a larger standard error of the estimate? (relevant section)



Q14

True/false: If the slope of a simple linear regression line is statistically significant, then the correlation will also always be significant. (relevant section)

Q15

True/false: If the slope of the relationship between X and Y is larger for Population 1 than for Population 2, the correlation will necessarily be larger in Population 1 than in Population 1. Why or why not? (relevant section)

Q16

True/false: If the correlation is 0.8, then 40% of the variance is explained. (relevant section)

Q17

True/false: If the actual Y score was 31, but the predicted score was 28, then the error of prediction is 3. (relevant section)

Questions from Case Studies

The following question is from the Angry Moods (AM) case study.

Q18

(AM#23) Find the regression line for predicting Anger-Out from Control-Out.

a. What is the slope?
b. What is the intercept?
c. Is the relationship at least approximately linear?
d. Test to see if the slope is significantly different from 0.
e. What is the standard error of the estimate?

(relevant section, relevant section, relevant section)

The following question is from the SAT and GPA (SG) case study.

Q19

(SG#3) Find the regression line for predicting the overall university GPA from the high school GPA.

a. What is the slope?
b. What is the y-intercept?
c. If someone had a 2.2 GPA in high school, what is the best estimate of his or her college GPA?
d. If someone had a 4.0 GPA in high school, what is the best estimate of his or her college GPA?

(relevant section)

The following questions are from the Driving (D) case study.

Q20

(D#5) What is the correlation between age and how often the person chooses to drive in inclement weather? Is this correlation statistically significant at the 0.01 level? Are older people more or less likely to report that they drive in inclement weather? (relevant section, relevant section)

Q21

(D#8) What is the correlation between how often a person chooses to drive in inclement weather and the percentage of accidents the person believes occur in inclement weather? Is this correlation significantly different from 0? (relevant section, relevant section)

Q22

(D#10) Use linear regression to predict how often someone rides public transportation in inclement weather from what percentage of accidents that person thinks occur in inclement weather. (Pubtran by Accident)

a. Create a scatter plot of this data and add a regression line.
b. What is the slope?
c. What is the intercept?
d. Is the relationship at least approximately linear?
e. Test if the slope is significantly different from 0.
f. Comment on possible assumption violations for the test of the slope.
g. What is the standard error of the estimate?

(relevant section, relevant section, relevant section)

Selected Answers

S2

a. 21

S5

a. 0.33

S6

b. $b = 1.91$

S9

a. 800

S12

$a = 19.2$

S18

e. 3.45

S19

c. 2.6

S20

$r = 0.43$

S22

b. **0.35**

This page titled 14.E: Regression (Exercises) is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **14.E: Regression (Exercises)** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

CHAPTER OVERVIEW

15: Regression in R

The goal in this chapter is to introduce ***linear regression***, the standard tool that statisticians rely on when analysing the relationship between interval scale predictors and interval scale outcomes. Stripped to its bare essentials, linear regression models are basically a slightly fancier version of the Pearson correlation (Section 5.7) though as we'll see, regression models are much more powerful tools.

15.1: What Is a Linear Regression Model?
15.2: Estimating a Linear Regression Model
15.3: Multiple Linear Regression
15.4: Quantifying the Fit of the Regression Model
15.5: Hypothesis Tests for Regression Models
15.6: Correlations
15.7: Handling Missing Values
15.8: Testing the Significance of a Correlation
15.9: Regarding Regression Coefficients
15.10: Assumptions of Regression
15.11: Model Checking
15.12: Model Selection
15.13: Summary

This page titled 15: Regression in R is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

15.1: What Is a Linear Regression Model?

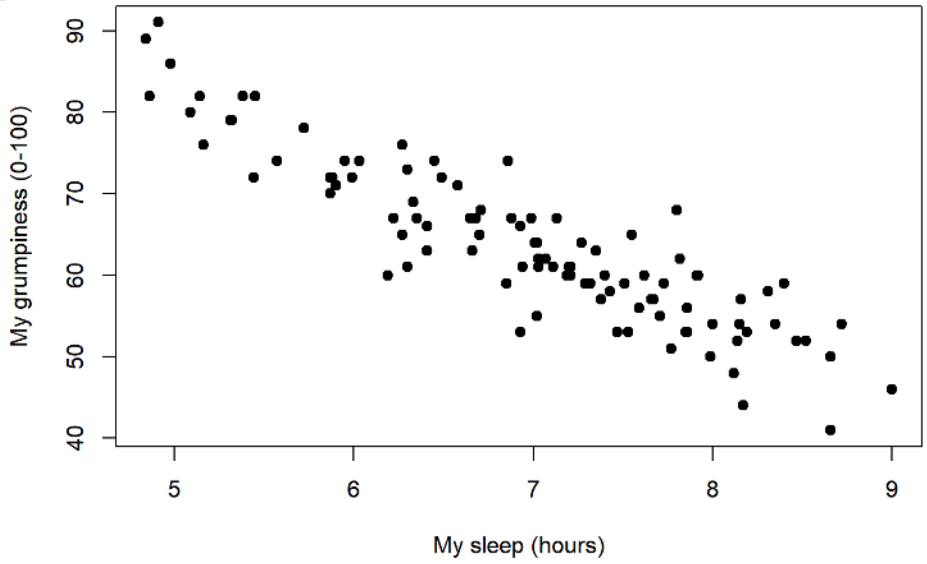


Figure 15.1: Scatterplot showing grumpiness as a function of hours slept.



Figure 15.2: Panel a shows the sleep-grumpiness scatterplot from above with the best fitting regression line drawn over the top. Not surprisingly, the line goes through the middle of the data.



Figure 15.3: In contrast, this plot shows the same data, but with a very poor choice of regression line drawn over the top.

Since the basic ideas in regression are closely tied to correlation, we'll return to the `parenthood.Rdata` file that we were using to illustrate how correlations work. Recall that, in this data set, we were trying to find out why Dan is so very grumpy all the time, and our working hypothesis was that I'm not getting enough sleep. We drew some scatterplots to help us examine the relationship between the amount of sleep I get, and my grumpiness the following day. The actual scatterplot that we draw is the one shown in Figure 15.1, and as we saw previously this corresponds to a correlation of r=−.90, but what we find ourselves secretly imagining is something that looks closer to Figure 15.2. That is, we mentally draw a straight line through the middle of the data. In statistics, this line that we're drawing is called a ***regression line***. Notice that – since we're not idiots – the regression line goes through the middle of the data. We don't find ourselves imagining anything like the rather silly plot shown in Figure 15.3.

This is not highly surprising: the line that I've drawn in Figure 15.3 doesn't "fit" the data very well, so it doesn't make a lot of sense to propose it as a way of summarising the data, right? This is a very simple observation to make, but it turns out to be very powerful when we start trying to wrap just a little bit of maths around it. To do so, let's start with a refresher of some high school maths. The formula for a straight line is usually written like this:

$$y=mx+c$$

Or, at least, that's what it was when I went to high school all those years ago. The two *variables* are x and y, and we have two *coefficients*, m and c. The coefficient m represents the *slope* of the line, and the coefficient c represents the *y-intercept* of the line. Digging further back into our decaying memories of high school (sorry, for some of us high school was a long time ago), we remember that the intercept is interpreted as "the value of y that you get when x=0". Similarly, a slope of m means that if you increase the x-value by 1 unit, then the y-value goes up by m units; a negative slope means that the y-value would go down rather than up. Ah yes, it's all coming back to me now.

Now that we've remembered that, it should come as no surprise to discover that we use the exact same formula to describe a regression line. If Y is the outcome variable (the DV) and X is the predictor variable (the IV), then the formula that describes our regression is written like this:

$$\hat{Y}_i = b_1 X_i + b_0$$

Hm. Looks like the same formula, but there's some extra frilly bits in this version. Let's make sure we understand them. Firstly, notice that I've written X_i and Y_i rather than just plain old X and Y. This is because we want to remember that we're dealing with actual data. In this equation, X_i is the value of predictor variable for the ith observation (i.e., the number of hours of sleep that I got on day i of my little study), and Y_i is the corresponding value of the outcome variable (i.e., my grumpiness on that day). And although I haven't said so explicitly in the equation, what we're assuming is that this formula works for all observations in the data

set (i.e., for all i). Secondly, notice that I wrote $\hat{Y}_i$ and not Yi. This is because we want to make the distinction between the *actual data* Y_i, and the *estimate* $\hat{Y}_i$ (i.e., the prediction that our regression line is making). Thirdly, I changed the letters used to describe the coefficients from m and c to b_1 and b_0. That's just the way that statisticians like to refer to the coefficients in a regression model. I've no idea why they chose b, but that's what they did. In any case b_0 always refers to the intercept term, and b1 refers to the slope.

Excellent, excellent. Next, I can't help but notice that – regardless of whether we're talking about the good regression line or the bad one – the data don't fall perfectly on the line. Or, to say it another way, the data Yi are not identical to the predictions of the regression model $\hat{Y}_i$. Since statisticians love to attach letters, names and numbers to everything, let's refer to the difference between the model prediction and that actual data point as a *residual*, and we'll refer to it as ϵ_i.[214] Written using mathematics, the residuals are defined as:

$$\epsilon_i = Y_i - \hat{Y}_i$$

which in turn means that we can write down the complete linear regression model as:

$$Y_i = b_1 X_i + b_0 + \epsilon_i$$

This page titled 15.1: What Is a Linear Regression Model? is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **15.1: What Is a Linear Regression Model?** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

15.2: Estimating a Linear Regression Model



Figure 15.4: A depiction of the residuals associated with the best fitting regression line



Figure 15.5: The residuals associated with a poor regression line

Okay, now let's redraw our pictures, but this time I'll add some lines to show the size of the residual for all observations. When the regression line is good, our residuals (the lengths of the solid black lines) all look pretty small, as shown in Figure 15.4, but when the regression line is a bad one, the residuals are a lot larger, as you can see from looking at Figure 15.5. Hm. Maybe what we "want" in a regression model is *small* residuals. Yes, that does seem to make sense. In fact, I think I'll go so far as to say that the "best fitting" regression line is the one that has the smallest residuals. Or, better yet, since statisticians seem to like to take squares of everything why not say that …

The estimated regression coefficients, $\hat{b}_0$ *and* $\hat{b}_1$ *are those that minimise the sum of the squared residuals, which we could either write as* $\sum_i \left(Y_i - \hat{Y}_i\right)^2$ *or as* $\sum_i \epsilon_i^2$.

Yes, yes that sounds even better. And since I've indented it like that, it probably means that this is the right answer. And since this is the right answer, it's probably worth making a note of the fact that our regression coefficients are *estimates* (we're trying to guess the parameters that describe a population!), which is why I've added the little hats, so that we get $\hat{b}_0$ and $\hat{b}_1$ rather than b0 and b1. Finally, I should also note that – since there's actually more than one way to estimate a regression model – the more technical name for this estimation process is ***ordinary least squares (OLS) regression***.

At this point, we now have a concrete definition for what counts as our "best" choice of regression coefficients, $\hat{b}_0$ and $\hat{b}_1$. The natural question to ask next is, if our optimal regression coefficients are those that minimise the sum squared residuals, how do we *find* these wonderful numbers? The actual answer to this question is complicated, and it doesn't help you understand the logic of regression.[215] As a result, this time I'm going to let you off the hook. Instead of showing you how to do it the long and tedious way first, and then "revealing" the wonderful shortcut that R provides you with, let's cut straight to the chase… and use the `lm()` function (short for "linear model") to do all the heavy lifting.

15.2.1 Using the `lm()` function

The `lm()` function is a fairly complicated one: if you type `?lm`, the help files will reveal that there are a lot of arguments that you can specify, and most of them won't make a lot of sense to you. At this stage however, there's really only two of them that you care about, and as it turns out you've seen them before:

- `formula`. A formula that specifies the regression model. For the simple linear regression models that we've talked about so far, in which you have a single predictor variable as well as an intercept term, this formula is of the form `outcome ~ predictor`. However, more complicated formulas are allowed, and we'll discuss them later.
- `data`. The data frame containing the variables.

As we saw with `aov()` in Chapter 14, the output of the `lm()` function is a fairly complicated object, with quite a lot of technical information buried under the hood. Because this technical information is used by other functions, it's generally a good idea to create a variable that stores the results of your regression. With this in mind, to run my linear regression, the command I want to use is this:

```
regression.1 <- lm( formula = dan.grump ~ dan.sleep,
                    data = parenthood )
```

Note that I used `dan.grump ~ dan.sleep` as the formula: in the model that I'm trying to estimate, `dan.grump` is the *outcome* variable, and `dan.sleep` is the predictor variable. It's always a good idea to remember which one is which! Anyway, what this does is create an "`lm` object" (i.e., a variable whose class is `"lm"`) called `regression.1`. Let's have a look at what happens when we `print()` it out:

```
print( regression.1 )
```

```
##
## Call:
## lm(formula = dan.grump ~ dan.sleep, data = parenthood)
##
## Coefficients:
## (Intercept)    dan.sleep
##     125.956       -8.937
```

This looks promising. There's two separate pieces of information here. Firstly, R is politely reminding us what the command was that we used to specify the model in the first place, which can be helpful. More importantly from our perspective, however, is the second part, in which R gives us the intercept $\hat{b}_0$ =125.96 and the slope $\hat{b}_1$ =−8.94. In other words, the best-fitting regression line that I plotted in Figure 15.2 has this formula:

$$\hat{Y}_i = -8.94\ X_i + 125.96$$

15.2.2 Interpreting the estimated model

The most important thing to be able to understand is how to interpret these coefficients. Let's start with $\hat{b}_1$, the slope. If we remember the definition of the slope, a regression coefficient of $\hat{b}_1$ =−8.94 means that if I increase X_i by 1, then I'm decreasing Y_i by 8.94. That is, each additional hour of sleep that I gain will improve my mood, reducing my grumpiness by 8.94 grumpiness

points. What about the intercept? Well, since $\hat{b}_0$ corresponds to "the expected value of Y_i when Xi equals 0", it's pretty straightforward. It implies that if I get zero hours of sleep (X_i=0) then my grumpiness will go off the scale, to an insane value of (Y_i=125.96). Best to be avoided, I think.

This page titled 15.2: Estimating a Linear Regression Model is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **15.2: Estimating a Linear Regression Model** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

15.3: Multiple Linear Regression

The simple linear regression model that we've discussed up to this point assumes that there's a single predictor variable that you're interested in, in this case `dan.sleep`. In fact, up to this point, *every* statistical tool that we've talked about has assumed that your analysis uses one predictor variable and one outcome variable. However, in many (perhaps most) research projects you actually have multiple predictors that you want to examine. If so, it would be nice to be able to extend the linear regression framework to be able to include multiple predictors. Perhaps some kind of ***multiple regression*** model would be in order?

Multiple regression is conceptually very simple. All we do is add more terms to our regression equation. Let's suppose that we've got two variables that we're interested in; perhaps we want to use both `dan.sleep` and `baby.sleep` to predict the `dan.grump` variable. As before, we let Y_i refer to my grumpiness on the i-th day. But now we have two X variables: the first corresponding to the amount of sleep I got and the second corresponding to the amount of sleep my son got. So we'll let X_{i1} refer to the hours I slept on the i-th day, and X_{i2} refers to the hours that the baby slept on that day. If so, then we can write our regression model like this:

$$Y_i = b_2 X_{i2} + b_1 X_{i1} + b_0 + \epsilon_i$$

As before, ϵi is the residual associated with the i-th observation, $\epsilon_i = Y_i - \hat{Y}_i$. In this model, we now have three coefficients that need to be estimated: b_0 is the intercept, b_1 is the coefficient associated with my sleep, and b_2 is the coefficient associated with my son's sleep. However, although the number of coefficients that need to be estimated has changed, the basic idea of how the estimation works is unchanged: our estimated coefficients $\hat{b}_0$, $\hat{b}_1$ and $\hat{b}_2$ are those that minimise the sum squared residuals.

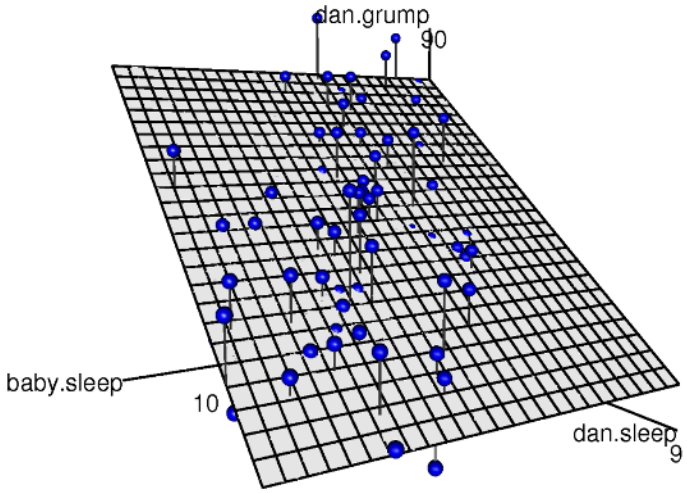


Figure 15.6: A 3D visualisation of a multiple regression model. There are two predictors in the model, `dan.sleep` and `baby.sleep`; the outcome variable is `dan.grump`. Together, these three variables form a 3D space: each observation (blue dots) is a point in this space. In much the same way that a simple linear regression model forms a line in 2D space, this multiple regression model forms a plane in 3D space. When we estimate the regression coefficients, what we're trying to do is find a plane that is as close to all the blue dots as possible.

15.3.1 Doing it in R

Multiple regression in R is no different to simple regression: all we have to do is specify a more complicated `formula` when using the `lm()` function. For example, if we want to use both `dan.sleep` and `baby.sleep` as predictors in our attempt to explain why I'm so grumpy, then the formula we need is this:

```
dan.grump ~ dan.sleep + baby.sleep
```

Notice that, just like last time, I haven't explicitly included any reference to the intercept term in this formula; only the two predictor variables and the outcome. By default, the `lm()` function assumes that the model should include an intercept (though you can get rid of it if you want). In any case, I can create a new regression model – which I'll call `regression.2` – using the following command:

```
regression.2 <- lm( formula = dan.grump ~ dan.sleep + baby.sleep,
                    data = parenthood )
```

And just like last time, if we `print()` out this regression model we can see what the estimated regression coefficients are:

```
print( regression.2 )
```

```
##
## Call:
## lm(formula = dan.grump ~ dan.sleep + baby.sleep, data = parenthood)
##
## Coefficients:
## (Intercept)    dan.sleep   baby.sleep
##   125.96557     -8.95025      0.01052
```

The coefficient associated with `dan.sleep` is quite large, suggesting that every hour of sleep I lose makes me a lot grumpier. However, the coefficient for `baby.sleep` is very small, suggesting that it doesn't really matter how much sleep my son gets; not really. What matters as far as my grumpiness goes is how much sleep *I* get. To get a sense of what this multiple regression model looks like, Figure 15.6 shows a 3D plot that plots all three variables, along with the regression model itself.

15.3.2 Formula for the general case

The equation that I gave above shows you what a multiple regression model looks like when you include two predictors. Not surprisingly, then, if you want more than two predictors all you have to do is add more X terms and more b coefficients. In other words, if you have K predictor variables in the model then the regression equation looks like this:

$$Y_i = \left(\sum_{k=1}^{K} b_k X_{ik}\right) + b_0 + \epsilon_i$$

This page titled 15.3: Multiple Linear Regression is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **15.3: Multiple Linear Regression** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

15.4: Quantifying the Fit of the Regression Model

So we now know how to estimate the coefficients of a linear regression model. The problem is, we don't yet know if this regression model is any good. For example, the `regression.1` model *claims* that every hour of sleep will improve my mood by quite a lot, but it might just be rubbish. Remember, the regression model only produces a prediction $\hat{Y}_i$ about what my mood is like: my actual mood is Y_i. If these two are very close, then the regression model has done a good job. If they are very different, then it has done a bad job.

15.4.1 R^2 value

Once again, let's wrap a little bit of mathematics around this. Firstly, we've got the sum of the squared residuals:

$$SS_{res} = \sum_i (Y_i - \hat{Y}_i)^2$$

which we would hope to be pretty small. Specifically, what we'd like is for it to be very small in comparison to the total variability in the outcome variable,

$$SS_{tot} = \sum_i (Y_i - \bar{Y})^2$$

While we're here, let's calculate these values in R. Firstly, in order to make my R commands look a bit more similar to the mathematical equations, I'll create variables `X` and `Y` :

```
X <- parenthood$dan.sleep  # the predictor
Y <- parenthood$dan.grump  # the outcome
```

Now that we've done this, let's calculate the $\hat{Y}$ values and store them in a variable called `Y.pred` . For the simple model that uses only a single predictor, `regression.1` , we would do the following:

```
Y.pred <- -8.94 * X  +  125.97
```

Okay, now that we've got a variable which stores the regression model predictions for how grumpy I will be on any given day, let's calculate our sum of squared residuals. We would do that using the following command:

```
SS.resid <- sum( (Y - Y.pred)^2 )
print( SS.resid )
```

```
## [1] 1838.722
```

Wonderful. A big number that doesn't mean very much. Still, let's forge boldly onwards anyway, and calculate the total sum of squares as well. That's also pretty simple:

```
SS.tot <- sum( (Y - mean(Y))^2 )
print( SS.tot )
```

```
## [1] 9998.59
```

Hm. Well, it's a much bigger number than the last one, so this does suggest that our regression model was making good predictions. But it's not very interpretable.

Perhaps we can fix this. What we'd like to do is to convert these two fairly meaningless numbers into one number. A nice, interpretable number, which for no particular reason we'll call R^2. What we would like is for the value of R^2 to be equal to 1 if the regression model makes no errors in predicting the data. In other words, if it turns out that the residual errors are zero – that is, if $SS_{res}=0$ – then we expect $R^2=1$. Similarly, if the model is completely useless, we would like R^2 to be equal to 0. What do I mean by

"useless"? Tempting as it is demand that the regression model move out of the house, cut its hair and get a real job, I'm probably going to have to pick a more practical definition: in this case, all I mean is that the residual sum of squares is no smaller than the total sum of squares, SS_{res}=SS_{tot}. Wait, why don't we do exactly that? The formula that provides us with out R^2 value is pretty simple to write down,

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

and equally simple to calculate in R:

```
R.squared <- 1 - (SS.resid / SS.tot)
print( R.squared )
```

```
## [1] 0.8161018
```

The R^2 value, sometimes called the ***coefficient of determination***[216] has a simple interpretation: it is the *proportion* of the variance in the outcome variable that can be accounted for by the predictor. So in this case, the fact that we have obtained R^2=.816 means that the predictor (`my.sleep`) explains 81.6% of the variance in the outcome (`my.grump`).

Naturally, you don't actually need to type in all these commands yourself if you want to obtain the R^2 value for your regression model. As we'll see later on in Section 15.5.3, all you need to do is use the `summary()` function. However, let's put that to one side for the moment. There's another property of R^2 that I want to point out.

15.4.2 relationship between regression and correlation

At this point we can revisit my earlier claim that regression, in this very simple form that I've discussed so far, is basically the same thing as a correlation. Previously, we used the symbol r to denote a Pearson correlation. Might there be some relationship between the value of the correlation coefficient r and the R^2 value from linear regression? Of course there is: the squared correlation r^2 is identical to the R^2 value for a linear regression with only a single predictor. To illustrate this, here's the squared correlation:

```
r <- cor(X, Y)  # calculate the correlation
print( r^2 )    # print the squared correlation
```

```
## [1] 0.8161027
```

Yep, same number. In other words, running a Pearson correlation is more or less equivalent to running a linear regression model that uses only one predictor variable.

15.4.3 adjusted R^2 value

One final thing to point out before moving on. It's quite common for people to report a slightly different measure of model performance, known as "adjusted R^2". The motivation behind calculating the adjusted R^2 value is the observation that adding more predictors into the model will *always* call the R^2 value to increase (or at least not decrease). The adjusted R^2 value introduces a slight change to the calculation, as follows. For a regression model with K predictors, fit to a data set containing N observations, the adjusted R^2 is:

$$\text{adj.}R^2 = 1 - \left(\frac{\text{SS}_{res}}{\text{SS}_{tot}} \times \frac{N-1}{N-K-1} \right)$$

This adjustment is an attempt to take the degrees of freedom into account. The big advantage of the adjusted R^2 value is that when you add more predictors to the model, the adjusted R^2 value will only increase if the new variables improve the model performance more than you'd expect by chance. The big disadvantage is that the adjusted R^2 value *can't* be interpreted in the elegant way that R^2 can. R^2 has a simple interpretation as the proportion of variance in the outcome variable that is explained by the regression model; to my knowledge, no equivalent interpretation exists for adjusted R^2.

An obvious question then, is whether you should report R^2 or adjusted R^2. This is probably a matter of personal preference. If you care more about interpretability, then R^2 is better. If you care more about correcting for bias, then adjusted R^2 is probably better. Speaking just for myself, I prefer R^2: my feeling is that it's more important to be able to interpret your measure of model performance. Besides, as we'll see in Section 15.5, if you're worried that the improvement in R^2 that you get by adding a predictor is just due to chance and not because it's a better model, well, we've got hypothesis tests for that.

This page titled 15.4: Quantifying the Fit of the Regression Model is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **15.4: Quantifying the Fit of the Regression Model** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

15.5: Hypothesis Tests for Regression Models

So far we've talked about what a regression model is, how the coefficients of a regression model are estimated, and how we quantify the performance of the model (the last of these, incidentally, is basically our measure of effect size). The next thing we need to talk about is hypothesis tests. There are two different (but related) kinds of hypothesis tests that we need to talk about: those in which we test whether the regression model as a whole is performing significantly better than a null model; and those in which we test whether a particular regression coefficient is significantly different from zero.

At this point, you're probably groaning internally, thinking that I'm going to introduce a whole new collection of tests. You're probably sick of hypothesis tests by now, and don't want to learn any new ones. Me too. I'm so sick of hypothesis tests that I'm going to shamelessly reuse the F-test from Chapter 14 and the t-test from Chapter 13. In fact, all I'm going to do in this section is show you how those tests are imported wholesale into the regression framework.

15.5.1 Testing the model as a whole

Okay, suppose you've estimated your regression model. The first hypothesis test you might want to try is one in which the null hypothesis that there is *no relationship* between the predictors and the outcome, and the alternative hypothesis is that *the data are distributed in exactly the way that the regression model predicts*. Formally, our "null model" corresponds to the fairly trivial "regression" model in which we include 0 predictors, and only include the intercept term b_0

$$H_0: Y_i = b_0 + \epsilon_i$$

If our regression model has K predictors, the "alternative model" is described using the usual formula for a multiple regression model:

$$H_1 : Y_i = \left(\sum_{k=1}^{K} b_k X_{ik} \right) + b_0 + \epsilon_i$$

How can we test these two hypotheses against each other? The trick is to understand that just like we did with ANOVA, it's possible to divide up the total variance SS_{tot} into the sum of the residual variance SS_{res} and the regression model variance SS_{mod}. I'll skip over the technicalities, since we covered most of them in the ANOVA chapter, and just note that:

$$SS_{mod} = SS_{tot} - SS_{res}$$

And, just like we did with the ANOVA, we can convert the sums of squares in to mean squares by dividing by the degrees of freedom.

$$\text{MS}_{mod} = \frac{\text{SS}_{mod}}{df_{mod}}$$
$$\text{MS}_{res} = \frac{\text{SS}_{res}}{df_{res}}$$

So, how many degrees of freedom do we have? As you might expect, the df associated with the model is closely tied to the number of predictors that we've included. In fact, it turns out that $df_{mod} = K$. For the residuals, the total degrees of freedom is $df_{res} = N - K - 1$.

$$F = \frac{MS_{mod}}{MS_{res}}$$

and the degrees of freedom associated with this are K and N−K−1. This F statistic has exactly the same interpretation as the one we introduced in Chapter 14. Large F values indicate that the null hypothesis is performing poorly in comparison to the alternative hypothesis. And since we already did some tedious "do it the long way" calculations back then, I won't waste your time repeating them. In a moment I'll show you how to do the test in R the easy way, but first, let's have a look at the tests for the individual regression coefficients.

15.5.2 Tests for individual coefficients

The F-test that we've just introduced is useful for checking that the model as a whole is performing better than chance. This is important: if your regression model doesn't produce a significant result for the F-test then you probably don't have a very good regression model (or, quite possibly, you don't have very good data). However, while failing this test is a pretty strong indicator that the model has problems, *passing* the test (i.e., rejecting the null) doesn't imply that the model is good! Why is that, you might be wondering? The answer to that can be found by looking at the coefficients for the `regression.2` model:

```
print( regression.2 )
```

```
##
## Call:
## lm(formula = dan.grump ~ dan.sleep + baby.sleep, data = parenthood)
##
## Coefficients:
## (Intercept)    dan.sleep   baby.sleep
##   125.96557     -8.95025      0.01052
```

I can't help but notice that the estimated regression coefficient for the `baby.sleep` variable is tiny (0.01), relative to the value that we get for `dan.sleep` (-8.95). Given that these two variables are absolutely on the same scale (they're both measured in "hours slept"), I find this suspicious. In fact, I'm beginning to suspect that it's really only the amount of sleep that *I* get that matters in order to predict my grumpiness.

Once again, we can reuse a hypothesis test that we discussed earlier, this time the t-test. The test that we're interested has a null hypothesis that the true regression coefficient is zero (b=0), which is to be tested against the alternative hypothesis that it isn't (b≠0). That is:

$$H_0: b=0$$

$$H_1: b\neq 0$$

How can we test this? Well, if the central limit theorem is kind to us, we might be able to guess that the sampling distribution of $\hat{b}$, the estimated regression coefficient, is a normal distribution with mean centred on b. What that would mean is that if the null hypothesis were true, then the sampling distribution of $\hat{b}$ has mean zero and unknown standard deviation. Assuming that we can come up with a good estimate for the standard error of the regression coefficient, SE ($\hat{b}$), then we're in luck. That's *exactly* the situation for which we introduced the one-sample t way back in Chapter 13. So let's define a t-statistic like this,

$$t = \frac{\hat{b}}{SE(\hat{b})}$$

I'll skip over the reasons why, but our degrees of freedom in this case are df=N−K−1. Irritatingly, the estimate of the standard error of the regression coefficient, SE($\hat{b}$), is not as easy to calculate as the standard error of the mean that we used for the simpler t-tests in Chapter 13. In fact, the formula is somewhat ugly, and not terribly helpful to look at. For our purposes it's sufficient to point out that the standard error of the estimated regression coefficient depends on both the predictor and outcome variables, and is somewhat sensitive to violations of the homogeneity of variance assumption (discussed shortly).

In any case, this t-statistic can be interpreted in the same way as the t-statistics that we discussed in Chapter 13. Assuming that you have a two-sided alternative (i.e., you don't really care if b>0 or b<0), then it's the extreme values of t (i.e., a lot less than zero or a lot greater than zero) that suggest that you should reject the null hypothesis.

15.5.3 Running the hypothesis tests in R

To compute all of the quantities that we have talked about so far, all you need to do is ask for a `summary()` of your regression model. Since I've been using `regression.2` as my example, let's do that:

```
summary( regression.2 )
```

```
##
## Call:
## lm(formula = dan.grump ~ dan.sleep + baby.sleep, data = parenthood)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0345  -2.2198  -0.4016   2.6775  11.7496
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 125.96557    3.04095  41.423   <2e-16 ***
## dan.sleep    -8.95025    0.55346 -16.172   <2e-16 ***
## baby.sleep    0.01052    0.27106   0.039    0.969
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.354 on 97 degrees of freedom
## Multiple R-squared:  0.8161, Adjusted R-squared:  0.8123
## F-statistic: 215.2 on 2 and 97 DF,  p-value: < 2.2e-16
```

The output that this command produces is pretty dense, but we've already discussed everything of interest in it, so what I'll do is go through it line by line. The first line reminds us of what the actual regression model is:

```
Call:
lm(formula = dan.grump ~ dan.sleep + baby.sleep, data = parenthood)
```

You can see why this is handy, since it was a little while back when we actually created the `regression.2` model, and so it's nice to be reminded of what it was we were doing. The next part provides a quick summary of the residuals (i.e., the ϵi values),

```
Residuals:
     Min       1Q   Median       3Q      Max
-11.0345  -2.2198  -0.4016   2.6775  11.7496
```

which can be convenient as a quick and dirty check that the model is okay. Remember, we did assume that these residuals were normally distributed, with mean 0. In particular it's worth quickly checking to see if the median is close to zero, and to see if the first quartile is about the same size as the third quartile. If they look badly off, there's a good chance that the assumptions of regression are violated. These ones look pretty nice to me, so let's move on to the interesting stuff. The next part of the R output looks at the coefficients of the regression model:

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 125.96557    3.04095  41.423   <2e-16 ***
dan.sleep    -8.95025    0.55346 -16.172   <2e-16 ***
baby.sleep    0.01052    0.27106   0.039    0.969
---
Signif. codes:  0 ***0.001 **0.01 *0.05 .0.1   1
```

Each row in this table refers to one of the coefficients in the regression model. The first row is the intercept term, and the later ones look at each of the predictors. The columns give you all of the relevant information. The first column is the actual estimate of b (e.g., 125.96 for the intercept, and -8.9 for the `dan.sleep` predictor). The second column is the standard error estimate $\hat{\sigma}_b$. The

third column gives you the t-statistic, and it's worth noticing that in this table t= $\hat{b}$ /SE($\hat{b}$) every time. Finally, the fourth column gives you the actual p value for each of these tests.[217] The only thing that the table itself doesn't list is the degrees of freedom used in the t-test, which is always N−K−1 and is listed immediately below, in this line:

```
Residual standard error: 4.354 on 97 degrees of freedom
```

The value of df=97 is equal to N−K−1, so that's what we use for our t-tests. In the final part of the output we have the F-test and the R^2 values which assess the performance of the model as a whole

```
Residual standard error: 4.354 on 97 degrees of freedom
Multiple R-squared: 0.8161, Adjusted R-squared: 0.8123
F-statistic: 215.2 on 2 and 97 DF,  p-value: < 2.2e-16
```

So in this case, the model performs significantly better than you'd expect by chance (F(2,97)=215.2, p<.001), which isn't all that surprising: the R^2=.812 value indicate that the regression model accounts for 81.2% of the variability in the outcome measure. However, when we look back up at the t-tests for each of the individual coefficients, we have pretty strong evidence that the `baby.sleep` variable has no significant effect; all the work is being done by the `dan.sleep` variable. Taken together, these results suggest that `regression.2` is actually the wrong model for the data: you'd probably be better off dropping the `baby.sleep` predictor entirely. In other words, the `regression.1` model that we started with is the better model.

This page titled 15.5: Hypothesis Tests for Regression Models is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **15.5: Hypothesis Tests for Regression Models** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

15.6: Correlations

Up to this point we have focused entirely on how to construct descriptive statistics for a single variable. What we haven't done is talked about how to describe the relationships *between* variables in the data. To do that, we want to talk mostly about the ***correlation*** between variables. But first, we need some data.

15.6.1 data

After spending so much time looking at the AFL data, I'm starting to get bored with sports. Instead, let's turn to a topic close to every parent's heart: sleep. The following data set is fictitious, but based on real events. Suppose I'm curious to find out how much my infant son's sleeping habits affect my mood. Let's say that I can rate my grumpiness very precisely, on a scale from 0 (not at all grumpy) to 100 (grumpy as a very, very grumpy old man). And, lets also assume that I've been measuring my grumpiness, my sleeping patterns and my son's sleeping patterns for quite some time now. Let's say, for 100 days. And, being a nerd, I've saved the data as a file called `parenthood.Rdata` . If we load the data…

```
load( "./data/parenthood.Rdata" )
who(TRUE)
```

```
##    -- Name --      -- Class --    -- Size --
##    parenthood      data.frame     100 x 4
##     $dan.sleep     numeric        100
##     $baby.sleep    numeric        100
##     $dan.grump     numeric        100
##     $day           integer        100
```

… we see that the file contains a single data frame called `parenthood` , which contains four variables `dan.sleep` , `baby.sleep` , `dan.grump` and `day` . If we peek at the data using `head()` out the data, here's what we get:

```
head(parenthood,10)
```

```
##    dan.sleep baby.sleep dan.grump day
## 1       7.59      10.18        56   1
## 2       7.91      11.66        60   2
## 3       5.14       7.92        82   3
## 4       7.71       9.61        55   4
## 5       6.68       9.75        67   5
## 6       5.99       5.04        72   6
## 7       8.19      10.45        53   7
## 8       7.19       8.27        60   8
## 9       7.40       6.06        60   9
## 10      6.58       7.09        71  10
```

Next, I'll calculate some basic descriptive statistics:

```
describe( parenthood )
```

```
##              vars   n  mean    sd median trimmed   mad   min    max range
## dan.sleep       1 100  6.97  1.02   7.03    7.00  1.09  4.84   9.00  4.16
## baby.sleep      2 100  8.05  2.07   7.95    8.05  2.33  3.25  12.07  8.82
## dan.grump       3 100 63.71 10.05  62.00   63.16  9.64 41.00  91.00 50.00
## day             4 100 50.50 29.01  50.50   50.50 37.06  1.00 100.00 99.00
##               skew kurtosis   se
## dan.sleep    -0.29    -0.72 0.10
## baby.sleep   -0.02    -0.69 0.21
## dan.grump     0.43    -0.16 1.00
## day           0.00    -1.24 2.90
```

Finally, to give a graphical depiction of what each of the three interesting variables looks like, Figure 5.6 plots histograms.

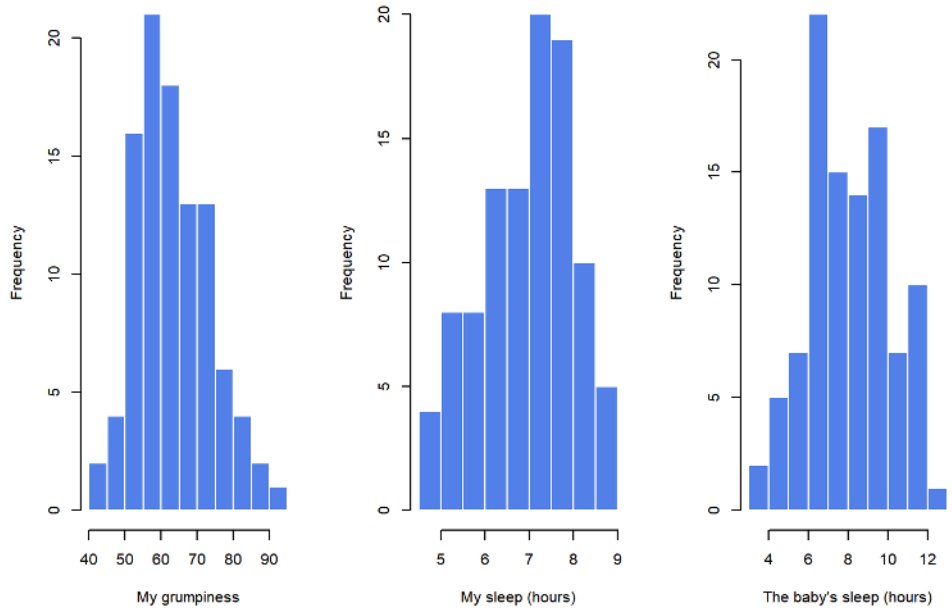


Figure 5.6: Histograms for the three interesting variables in the `parenthood` data set

One thing to note: just because R can calculate dozens of different statistics doesn't mean you should report all of them. If I were writing this up for a report, I'd probably pick out those statistics that are of most interest to me (and to my readership), and then put them into a nice, simple table like the one in Table **??**.[79] Notice that when I put it into a table, I gave everything "human readable" names. This is always good practice. Notice also that I'm not getting enough sleep. This isn't good practice, but other parents tell me that it's standard practice.

Table 5.2: Descriptive statistics for the parenthood data.

variable	min	max	mean	median	std. dev	IQR
Dan's grumpiness	41	91	63.71	62	10.05	14
Dan's hours slept	4.84	9	6.97	7.03	1.02	1.45
Dan's son's hours slept	3.25	12.07	8.05	7.95	2.07	3.21

15.6.2 strength and direction of a relationship

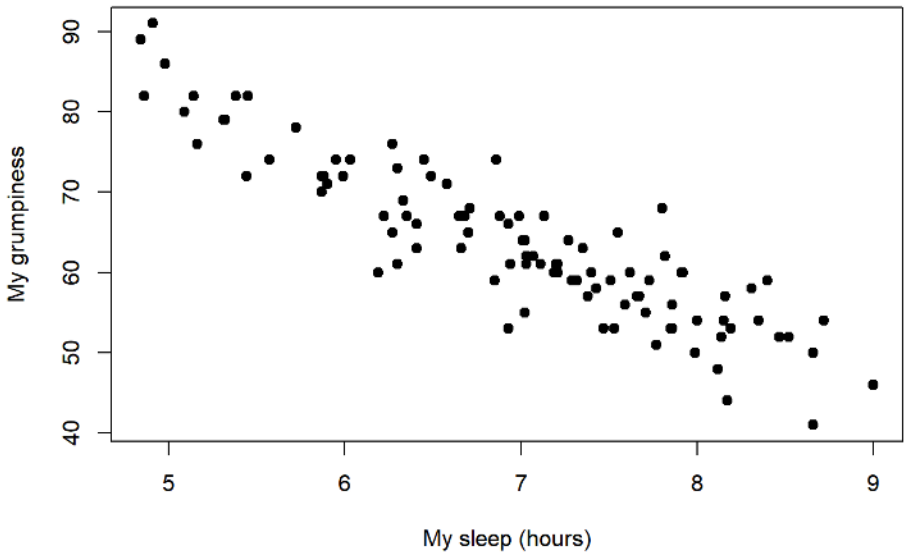


Figure 5.7: Scatterplot showing the relationship between `dan.sleep` and `dan.grump`

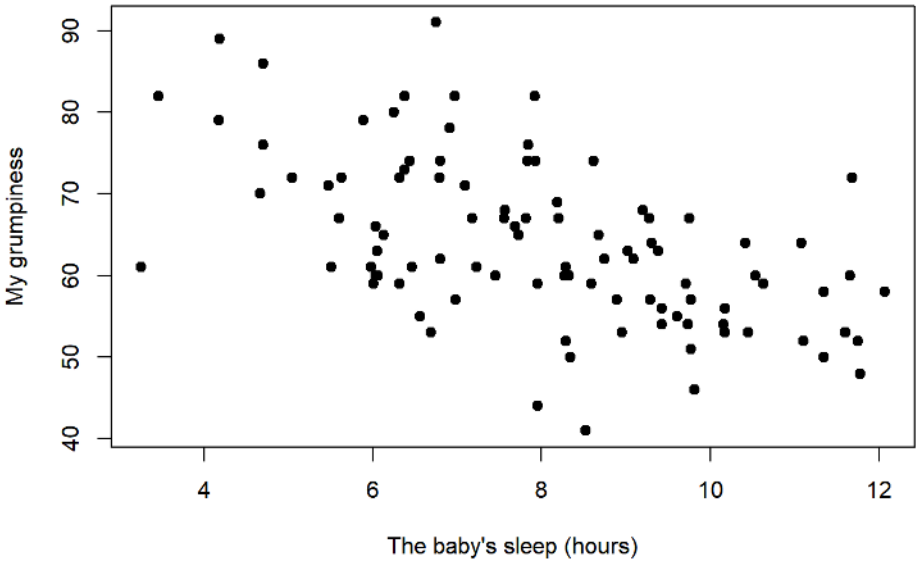


Figure 5.8: Scatterplot showing the relationship between `baby.sleep` and `dan.grump`

We can draw scatterplots to give us a general sense of how closely related two variables are. Ideally though, we might want to say a bit more about it than that. For instance, let's compare the relationship between `dan.sleep` and `dan.grump` (Figure 5.7 with that between `baby.sleep` and `dan.grump` (Figure 5.8. When looking at these two plots side by side, it's clear that the relationship is *qualitatively* the same in both cases: more sleep equals less grump! However, it's also pretty obvious that the relationship between `dan.sleep` and `dan.grump` is *stronger* than the relationship between `baby.sleep` and `dan.grump` . The plot on the left is "neater" than the one on the right. What it feels like is that if you want to predict what my mood is, it'd help you a little bit to know how many hours my son slept, but it'd be *more* helpful to know how many hours I slept.

In contrast, let's consider Figure 5.8 vs. Figure 5.9. If we compare the scatterplot of " `baby.sleep` v `dan.grump` " to the scatterplot of "` `baby.sleep` v `dan.sleep` ", the overall strength of the relationship is the same, but the direction is different. That is, if my son sleeps more, I get *more* sleep (positive relationship, but if he sleeps more then I get *less* grumpy (negative relationship).

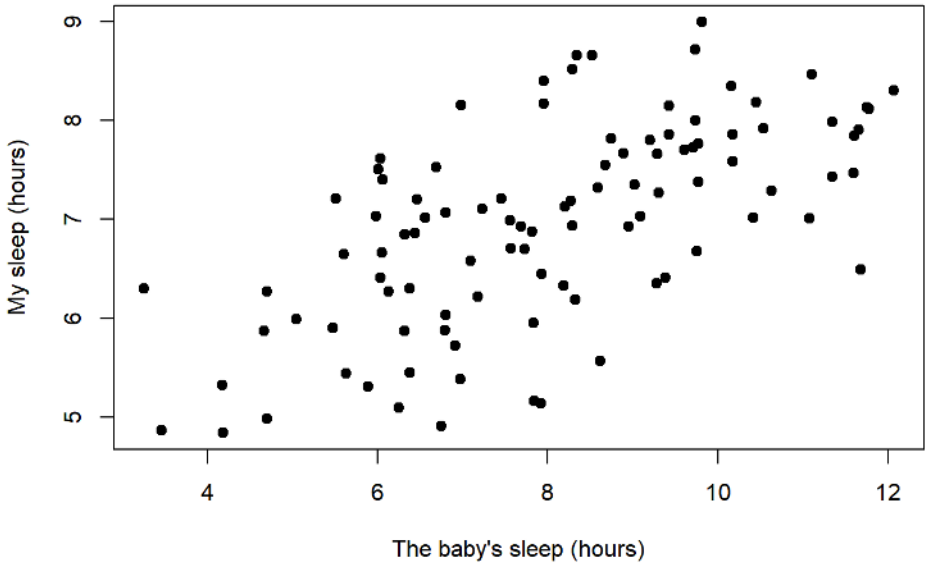


Figure 5.9: Scatterplot showing the relationship between `baby.sleep` and `dan.sleep`

15.6.3 correlation coefficient

We can make these ideas a bit more explicit by introducing the idea of a ***correlation coefficient*** (or, more specifically, Pearson's correlation coefficient), which is traditionally denoted by r. The correlation coefficient between two variables X and Y (sometimes denoted rXY), which we'll define more precisely in the next section, is a measure that varies from −1 to 1. When r=−1 it means that we have a perfect negative relationship, and when r=1 it means we have a perfect positive relationship. When r=0, there's no relationship at all. If you look at Figure 5.10, you can see several plots showing what different correlations look like.

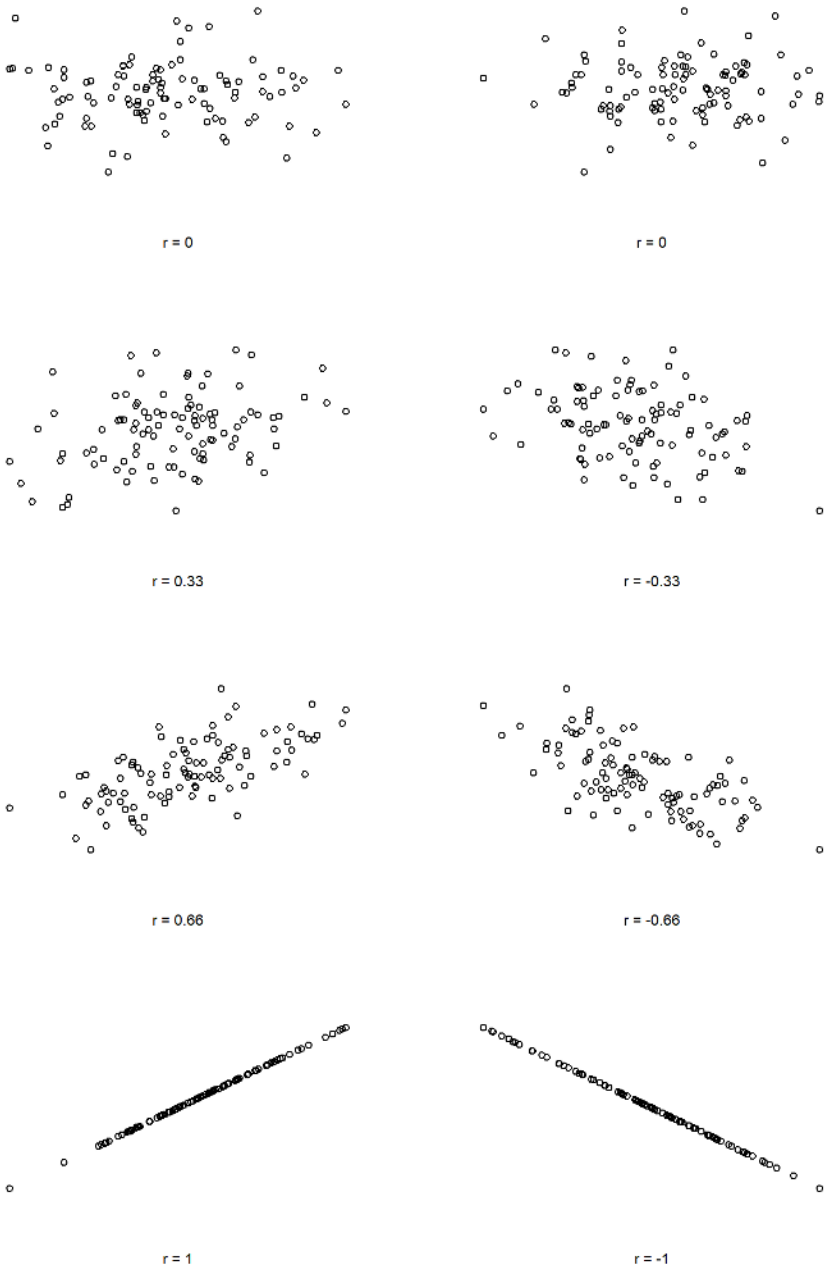


Figure 5.10: Illustration of the effect of varying the strength and direction of a correlation

The formula for the Pearson's correlation coefficient can be written in several different ways. I think the simplest way to write down the formula is to break it into two steps. Firstly, let's introduce the idea of a ***covariance***. The covariance between two variables X and Y is a generalisation of the notion of the variance; it's a mathematically simple way of describing the relationship between two variables that isn't terribly informative to humans:

$$\text{Cov}(X,Y) = \frac{1}{N-1} \sum_{i=1}^{N} \left(X_i - \bar{X}\right) \left(Y_i - \bar{Y}\right)$$

Because we're multiplying (i.e., taking the "product" of) a quantity that depends on X by a quantity that depends on Y and then averaging[80], you can think of the formula for the covariance as an "average cross product" between X and Y. The covariance has the nice property that, if X and Y are entirely unrelated, then the covariance is exactly zero. If the relationship between them is positive (in the sense shown in Figure@reffig:corr) then the covariance is also positive; and if the relationship is negative then the covariance is also negative. In other words, the covariance captures the basic qualitative idea of correlation. Unfortunately, the raw

magnitude of the covariance isn't easy to interpret: it depends on the units in which X and Y are expressed, and worse yet, the actual units that the covariance itself is expressed in are really weird. For instance, if X refers to the `dan.sleep` variable (units: hours) and Y refers to the `dan.grump` variable (units: grumps), then the units for their covariance are "hours × grumps". And I have no freaking idea what that would even mean.

The Pearson correlation coefficient r fixes this interpretation problem by standardising the covariance, in pretty much the exact same way that the z-score standardises a raw score: by dividing by the standard deviation. However, because we have two variables that contribute to the covariance, the standardisation only works if we divide by both standard deviations.[81] In other words, the correlation between X and Y can be written as follows:

$$r_{XY} = \frac{\text{Cov}(X,Y)}{\hat{\sigma}_X \hat{\sigma}_Y}$$

By doing this standardisation, not only do we keep all of the nice properties of the covariance discussed earlier, but the actual values of r are on a meaningful scale: r=1 implies a perfect positive relationship, and r=−1 implies a perfect negative relationship. I'll expand a little more on this point later, in Section@refsec:interpretingcorrelations. But before I do, let's look at how to calculate correlations in R.

15.6.4 Calculating correlations in R

Calculating correlations in R can be done using the `cor()` command. The simplest way to use the command is to specify two input arguments `x` and `y`, each one corresponding to one of the variables. The following extract illustrates the basic usage of the function:[82]

```
cor( x = parenthood$dan.sleep, y = parenthood$dan.grump )
```

```
## [1] -0.903384
```

However, the `cor()` function is a bit more powerful than this simple example suggests. For example, you can also calculate a complete "correlation matrix", between all pairs of variables in the data frame:[83]

```
# correlate all pairs of variables in "parenthood":
cor( x = parenthood )
```

```
##              dan.sleep  baby.sleep   dan.grump         day
## dan.sleep   1.00000000  0.62794934 -0.90338404 -0.09840768
## baby.sleep  0.62794934  1.00000000 -0.56596373 -0.01043394
## dan.grump  -0.90338404 -0.56596373  1.00000000  0.07647926
## day        -0.09840768 -0.01043394  0.07647926  1.00000000
```

15.6.5 Interpreting a correlation

Naturally, in real life you don't see many correlations of 1. So how should you interpret a correlation of, say r=.4? The honest answer is that it really depends on what you want to use the data for, and on how strong the correlations in your field tend to be. A friend of mine in engineering once argued that any correlation less than .95 is completely useless (I think he was exaggerating, even for engineering). On the other hand there are real cases – even in psychology – where you should really expect correlations that strong. For instance, one of the benchmark data sets used to test theories of how people judge similarities is so clean that any theory that can't achieve a correlation of at least .9 really isn't deemed to be successful. However, when looking for (say) elementary correlates of intelligence (e.g., inspection time, response time), if you get a correlation above .3 you're doing very very well. In short, the interpretation of a correlation depends a lot on the context. That said, the rough guide in Table **??** is pretty typical.

```
knitr::kable(
rbind(
c("-1.0 to -0.9" ,"Very strong", "Negative"),
c("-0.9 to -0.7", "Strong", "Negative") ,
c("-0.7 to -0.4", "Moderate", "Negative") ,
c("-0.4 to -0.2", "Weak", "Negative"),
c("-0.2 to 0","Negligible", "Negative") ,
c("0 to 0.2","Negligible", "Positive"),
c("0.2 to 0.4", "Weak", "Positive"),
c("0.4 to 0.7", "Moderate", "Positive"),
c("0.7 to 0.9", "Strong", "Positive"),
c("0.9 to 1.0", "Very strong", "Positive")), col.names=c("Correlation", "Strength", "
  booktabs = TRUE)
```

Correlation	Strength	Direction
-1.0 to -0.9	Very strong	Negative
-0.9 to -0.7	Strong	Negative
-0.7 to -0.4	Moderate	Negative
-0.4 to -0.2	Weak	Negative
-0.2 to 0	Negligible	Negative
0 to 0.2	Negligible	Positive
0.2 to 0.4	Weak	Positive
0.4 to 0.7	Moderate	Positive
0.7 to 0.9	Strong	Positive
0.9 to 1.0	Very strong	Positive

However, something that can never be stressed enough is that you should *always* look at the scatterplot before attaching any interpretation to the data. A correlation might not mean what you think it means. The classic illustration of this is "Anscombe's Quartet" (**???** Anscombe1973), which is a collection of four data sets. Each data set has two variables, an X and a Y. For all four data sets the mean value for X is 9 and the mean for Y is 7.5. The, standard deviations for all X variables are almost identical, as are those for the the Y variables. And in each case the correlation between X and Y is r=0.816. You can verify this yourself, since the dataset comes distributed with R. The commands would be:

```
cor( anscombe$x1, anscombe$y1 )
```

```
## [1] 0.8164205
```

```
cor( anscombe$x2, anscombe$y2 )
```

```
## [1] 0.8162365
```

and so on.

You'd think that these four data setswould look pretty similar to one another. They do not. If we draw scatterplots of X against Y for all four variables, as shown in Figure 5.11 we see that all four of these are *spectacularly* different to each other.

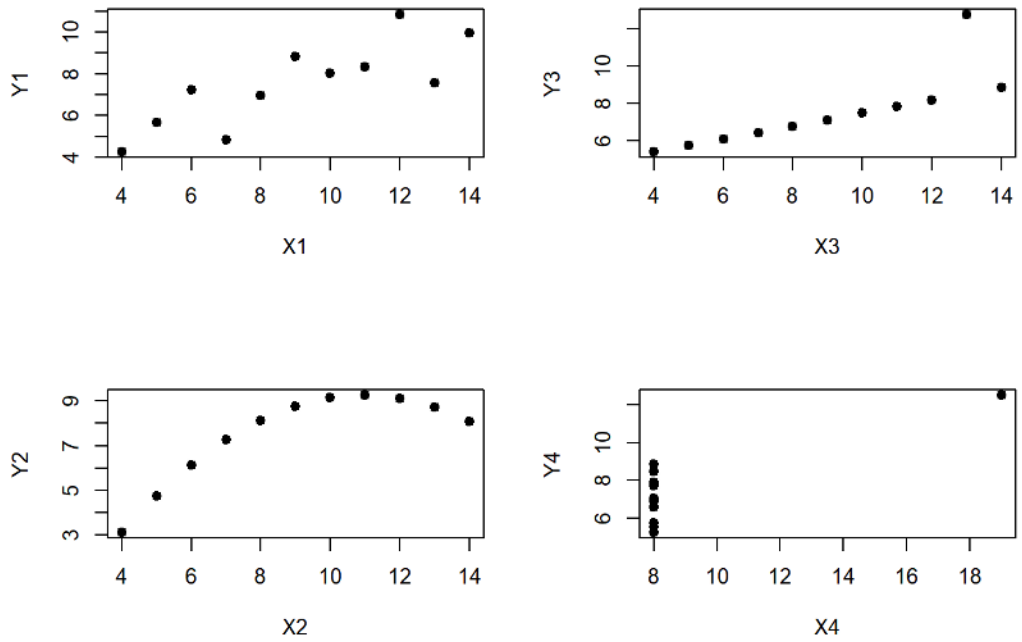


Figure 5.11: Anscombe's quartet. All four of these data sets have a Pearson correlation of r=.816, but they are qualitatively different from one another.

The lesson here, which so very many people seem to forget in real life is "*always graph your raw data*". This will be the focus of Chapter 6.

15.6.6 Spearman's rank correlations

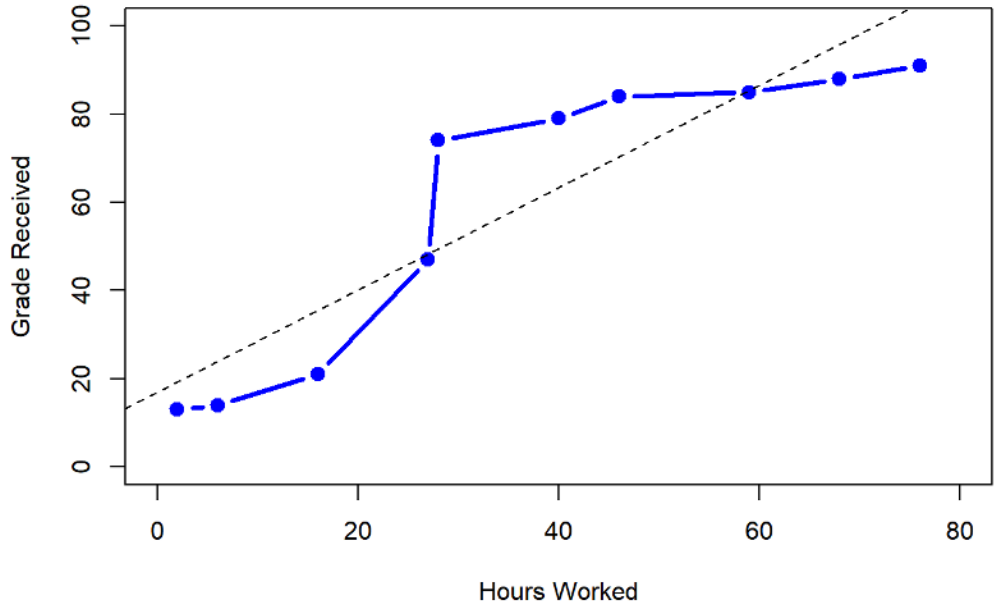


Figure 5.12: The relationship between hours worked and grade received, for a toy data set consisting of only 10 students (each circle corresponds to one student). The dashed line through the middle shows the linear relationship between the two variables. This produces a strong Pearson correlation of r=.91. However, the interesting thing to note here is that there's actually a perfect monotonic relationship between the two variables: in this toy example at least, increasing the hours worked always increases the grade received, as illustrated by the solid line. This is reflected in a Spearman correlation of rho=1. With such a small data set, however, it's an open question as to which version better describes the actual relationship involved.

The Pearson correlation coefficient is useful for a lot of things, but it does have shortcomings. One issue in particular stands out: what it actually measures is the strength of the *linear* relationship between two variables. In other words, what it gives you is a measure of the extent to which the data all tend to fall on a single, perfectly straight line. Often, this is a pretty good approximation to what we mean when we say "relationship", and so the Pearson correlation is a good thing to calculation. Sometimes, it isn't.

One very common situation where the Pearson correlation isn't quite the right thing to use arises when an increase in one variable X really is reflected in an increase in another variable Y, but the nature of the relationship isn't necessarily linear. An example of this might be the relationship between effort and reward when studying for an exam. If you put in zero effort (X) into learning a subject, then you should expect a grade of 0% (Y). However, a little bit of effort will cause a *massive* improvement: just turning up to lectures means that you learn a fair bit, and if you just turn up to classes, and scribble a few things down so your grade might rise to 35%, all without a lot of effort. However, you just don't get the same effect at the other end of the scale. As everyone knows, it takes *a lot* more effort to get a grade of 90% than it takes to get a grade of 55%. What this means is that, if I've got data looking at study effort and grades, there's a pretty good chance that Pearson correlations will be misleading.

To illustrate, consider the data plotted in Figure 5.12, showing the relationship between hours worked and grade received for 10 students taking some class. The curious thing about this – highly fictitious – data set is that increasing your effort *always* increases your grade. It might be by a lot or it might be by a little, but increasing effort will never decrease your grade. The data are stored in `effort.Rdata` :

```
> load( "effort.Rdata" )
> who(TRUE)
   -- Name --    -- Class --    -- Size --
   effort        data.frame     10 x 2
    $hours       numeric        10
    $grade       numeric        10
```

The raw data look like this:

```
> effort
   hours grade
1      2    13
2     76    91
3     40    79
4      6    14
5     16    21
6     28    74
7     27    47
8     59    85
9     46    84
10    68    88
```

If we run a standard Pearson correlation, it shows a strong relationship between hours worked and grade received,

```
> cor( effort$hours, effort$grade )
[1] 0.909402
```

but this doesn't actually capture the observation that increasing hours worked *always* increases the grade. There's a sense here in which we want to be able to say that the correlation is *perfect* but for a somewhat different notion of what a "relationship" is. What we're looking for is something that captures the fact that there is a perfect ***ordinal relationship*** here. That is, if student 1 works more hours than student 2, then we can guarantee that student 1 will get the better grade. That's not what a correlation of r=.91 says at all.

How should we address this? Actually, it's really easy: if we're looking for ordinal relationships, all we have to do is treat the data as if it were ordinal scale! So, instead of measuring effort in terms of "hours worked", lets rank all 10 of our students in order of hours worked. That is, student 1 did the least work out of anyone (2 hours) so they get the lowest rank (rank = 1). Student 4 was the next laziest, putting in only 6 hours of work in over the whole semester, so they get the next lowest rank (rank = 2). Notice that I'm using "rank =1" to mean "low rank". Sometimes in everyday language we talk about "rank = 1" to mean "top rank" rather than "bottom rank". So be careful: you can rank "from smallest value to largest value" (i.e., small equals rank 1) or you can rank "from largest value to smallest value" (i.e., large equals rank 1). In this case, I'm ranking from smallest to largest, because that's the default way that R does it. But in real life, it's really easy to forget which way you set things up, so you have to put a bit of effort into remembering!

Okay, so let's have a look at our students when we rank them from worst to best in terms of effort and reward:

	rank (hours worked)	rank (grade received)
student	1	1
student	2	10
student	3	6
student	4	2
student	5	3
student	6	5
student	7	4
student	8	8
student	9	7
student	10	9

Hm. These are *identical*. The student who put in the most effort got the best grade, the student with the least effort got the worst grade, etc. We can get R to construct these rankings using the `rank()` function, like this:

```
> hours.rank <- rank( effort$hours )   # rank students by hours worked
> grade.rank <- rank( effort$grade )   # rank students by grade received
```

As the table above shows, these two rankings are identical, so if we now correlate them we get a perfect relationship:

```
> cor( hours.rank, grade.rank )
[1] 1
```

What we've just re-invented is ***Spearman's rank order correlation***, usually denoted ρ to distinguish it from the Pearson correlation r. We can calculate Spearman's ρ using R in two different ways. Firstly we could do it the way I just showed, using the `rank()` function to construct the rankings, and then calculate the Pearson correlation on these ranks. However, that's way too much effort to do every time. It's much easier to just specify the `method` argument of the `cor()` function.

```
> cor( effort$hours, effort$grade, method = "spearman")
[1] 1
```

The default value of the `method` argument is `"pearson"`, which is why we didn't have to specify it earlier on when we were doing Pearson correlations.

15.6.7 `correlate()` function

As we've seen, the `cor()` function works pretty well, and handles many of the situations that you might be interested in. One thing that many beginners find frustrating, however, is the fact that it's not built to handle non-numeric variables. From a statistical perspective, this is perfectly sensible: Pearson and Spearman correlations are only designed to work for numeric variables, so the `cor()` function spits out an error.

Here's what I mean. Suppose you were keeping track of how many `hours` you worked in any given day, and counted how many `tasks` you completed. If you were doing the tasks for money, you might also want to keep track of how much `pay` you got for each job. It would also be sensible to keep track of the `weekday` on which you actually did the work: most of us don't work as much on Saturdays or Sundays. If you did this for 7 weeks, you might end up with a data set that looks like this one:

```
> load("work.Rdata")

> who(TRUE)
   -- Name --    -- Class --    -- Size --
   work          data.frame     49 x 7
    $hours       numeric        49
    $tasks       numeric        49
    $pay         numeric        49
    $day         integer        49
    $weekday     factor         49
    $week        numeric        49
    $day.type    factor         49

> head(work)
  hours tasks pay day   weekday week day.type
1   7.2    14  41   1   Tuesday    1  weekday
2   7.4    11  39   2 Wednesday    1  weekday
3   6.6    14  13   3  Thursday    1  weekday
4   6.5    22  47   4    Friday    1  weekday
5   3.1     5   4   5  Saturday    1  weekend
6   3.0     7  12   6    Sunday    1  weekend
```

Obviously, I'd like to know something about how all these variables correlate with one another. I could correlate `hours` with `pay` quite using `cor()`, like so:

```
> cor(work$hours,work$pay)
[1] 0.7604283
```

But what if I wanted a quick and easy way to calculate all pairwise correlations between the numeric variables? I can't just input the `work` data frame, because it contains two factor variables, `weekday` and `day.type`. If I try this, I get an error:

```
> cor(work)
Error in cor(work) : 'x' must be numeric
```

It order to get the correlations that I want using the `cor()` function, is create a new data frame that doesn't contain the factor variables, and then feed that new data frame into the `cor()` function. It's not actually very hard to do that, and I'll talk about how to do it properly in Section@refsec:subsetdataframe. But it would be nice to have some function that is smart enough to just ignore the factor variables. That's where the `correlate()` function in the `lsr` package can be handy. If you feed it a data frame that contains factors, it knows to ignore them, and returns the pairwise correlations only between the numeric variables:

```
> correlate(work)

CORRELATIONS
============
- correlation type:  pearson
- correlations shown only when both variables are numeric

          hours  tasks   pay    day weekday   week day.type
hours         .  0.800 0.760 -0.049       .  0.018        .
tasks     0.800      . 0.720 -0.072       . -0.013        .
pay       0.760  0.720     .  0.137       .  0.196        .
day      -0.049 -0.072 0.137      .       .  0.990        .
weekday       .      .     .      .       .      .        .
week      0.018 -0.013 0.196  0.990       .      .        .
day.type      .      .     .      .       .      .        .
```

The output here shows a `.` whenever one of the variables is non-numeric. It also shows a `.` whenever a variable is correlated with itself (it's not a meaningful thing to do). The `correlate()` function can also do Spearman correlations, by specifying the `corr.method` to use:

```
> correlate( work, corr.method="spearman" )

CORRELATIONS
============
- correlation type:  spearman
- correlations shown only when both variables are numeric

          hours  tasks   pay    day weekday   week day.type
hours         .  0.805 0.745 -0.047       .  0.010        .
tasks     0.805      . 0.730 -0.068       . -0.008        .
pay       0.745  0.730     .  0.094       .  0.154        .
day      -0.047 -0.068 0.094      .       .  0.990        .
weekday       .      .     .      .       .      .        .
week      0.010 -0.008 0.154  0.990       .      .        .
day.type      .      .     .      .       .      .        .
```

Obviously, there's no new functionality in the `correlate()` function, and any advanced R user would be perfectly capable of using the `cor()` function to get these numbers out. But if you're not yet comfortable with extracting a subset of a data frame, the `correlate()` function is for you.

This page titled 15.6: Correlations is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **5.7: Correlations** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

15.7: Handling Missing Values

There's one last topic that I want to discuss briefly in this chapter, and that's the issue of ***missing data***. Real data sets very frequently turn out to have missing values: perhaps someone forgot to fill in a particular survey question, for instance. Missing data can be the source of a lot of tricky issues, most of which I'm going to gloss over. However, at a minimum, you need to understand the basics of handling missing data in R.

15.7.1 single variable case

Let's start with the simplest case, in which you're trying to calculate descriptive statistics for a single variable which has missing data. In R, this means that there will be `NA` values in your data vector. Let's create a variable like that:

```
> partial <- c(10, 20, NA, 30)
```

Let's assume that you want to calculate the mean of this variable. By default, R assumes that you want to calculate the mean using all four elements of this vector, which is probably the safest thing for a dumb automaton to do, but it's rarely what you actually want. Why not? Well, remember that the basic interpretation of `NA` is "I don't know what this number is". This means that `1 + NA = NA` : if I add 1 to some number that I don't know (i.e., the `NA`) then the answer is *also* a number that I don't know. As a consequence, if you don't explicitly tell R to ignore the `NA` values, and the data set does have missing values, then the output will itself be a missing value. If I try to calculate the mean of the `partial` vector, without doing anything about the missing value, here's what happens:

```
> mean( x = partial )
[1] NA
```

Technically correct, but deeply unhelpful.

To fix this, all of the descriptive statistics functions that I've discussed in this chapter (with the exception of `cor()` which is a special case I'll discuss below) have an optional argument called `na.rm` , which is shorthand for "remove NA values". By default, `na.rm = FALSE` , so R does nothing about the missing data problem. Let's try setting `na.rm = TRUE` and see what happens:

When calculating sums and means when missing data are present (i.e., when there are `NA` values) there's actually an additional argument to the function that you should be aware of. This argument is called `na.rm` , and is a logical value indicating whether R should ignore (or "remove") the missing data for the purposes of doing the calculations. By default, R assumes that you want to keep the missing values, so unless you say otherwise it will set `na.rm = FALSE` . However, R assumes that `1 + NA = NA` : if I add 1 to some number that I don't know (i.e., the `NA`) then the answer is *also* a number that I don't know. As a consequence, if you don't explicitly tell R to ignore the `NA` values, and the data set does have missing values, then the output will itself be a missing value. This is illustrated in the following extract:

```
> mean( x = partial, na.rm = TRUE )
[1] 20
```

Notice that the mean is `20` (i.e., `60 / 3`) and *not* `15` . When R ignores a `NA` value, it genuinely ignores it. In effect, the calculation above is identical to what you'd get if you asked for the mean of the three-element vector `c(10, 20, 30)` .

As indicated above, this isn't unique to the `mean()` function. Pretty much all of the other functions that I've talked about in this chapter have an `na.rm` argument that indicates whether it should ignore missing values. However, its behaviour is the same for all these functions, so I won't waste everyone's time by demonstrating it separately for each one.

15.7.2 Missing values in pairwise calculations

I mentioned earlier that the `cor()` function is a special case. It doesn't have an `na.rm` argument, because the story becomes a lot more complicated when more than one variable is involved. What it does have is an argument called `use` which does

roughly the same thing, but you need to think little more carefully about what you want this time. To illustrate the issues, let's open up a data set that has missing values, `parenthood2.Rdata`. This file contains the same data as the original parenthood data, but with some values deleted. It contains a single data frame, `parenthood2`:

```
> load( "parenthood2.Rdata" )
> print( parenthood2 )
  dan.sleep baby.sleep dan.grump day
1      7.59         NA        56   1
2      7.91      11.66        60   2
3      5.14       7.92        82   3
4      7.71       9.61        55   4
5      6.68       9.75        NA   5
6      5.99       5.04        72   6
BLAH BLAH BLAH
```

If I calculate my descriptive statistics using the `describe()` function

```
> describe( parenthood2 )
           var   n  mean    sd median trimmed   mad   min    max    BLAH
dan.sleep    1  91  6.98  1.02   7.03    7.02  1.13  4.84   9.00    BLAH
baby.sleep   2  89  8.11  2.05   8.20    8.13  2.28  3.25  12.07    BLAH
dan.grump    3  92 63.15  9.85  61.00   62.66 10.38 41.00  89.00    BLAH
day          4 100 50.50 29.01  50.50   50.50 37.06  1.00 100.00    BLAH
```

we can see from the `n` column that there are 9 missing values for `dan.sleep`, 11 missing values for `baby.sleep` and 8 missing values for `dan.grump`.[84] Suppose what I would like is a correlation matrix. And let's also suppose that I don't bother to tell R how to handle those missing values. Here's what happens:

```
> cor( parenthood2 )
           dan.sleep baby.sleep dan.grump day
dan.sleep          1         NA        NA  NA
baby.sleep        NA          1        NA  NA
dan.grump         NA         NA         1  NA
day               NA         NA        NA   1
```

Annoying, but it kind of makes sense. If I don't *know* what some of the values of `dan.sleep` and `baby.sleep` actually are, then I can't possibly *know* what the correlation between these two variables is either, since the formula for the correlation coefficient makes use of every single observation in the data set. Once again, it makes sense: it's just not particularly *helpful*.

To make R behave more sensibly in this situation, you need to specify the `use` argument to the `cor()` function. There are several different values that you can specify for this, but the two that we care most about in practice tend to be `"complete.obs"` and `"pairwise.complete.obs"`. If we specify `use = "complete.obs"`, R will completely ignore all cases (i.e., all rows in our `parenthood2` data frame) that have any missing values at all. So, for instance, if you look back at the extract earlier when I used the `head()` function, notice that observation 1 (i.e., day 1) of the `parenthood2` data set is missing the value for `baby.sleep`, but is otherwise complete? Well, if you choose `use = "complete.obs"` R will ignore that row completely: that is, even when it's trying to calculate the correlation between `dan.sleep` and `dan.grump`, observation 1 will be ignored, because the value of `baby.sleep` is missing for that observation. Here's what we get:

```
> cor(parenthood2, use = "complete.obs")
             dan.sleep baby.sleep   dan.grump         day
dan.sleep   1.00000000  0.6394985 -0.89951468  0.06132891
baby.sleep  0.63949845  1.0000000 -0.58656066  0.14555814
dan.grump  -0.89951468 -0.5865607  1.00000000 -0.06816586
day         0.06132891  0.1455581 -0.06816586  1.00000000
```

The other possibility that we care about, and the one that tends to get used more often in practice, is to set `use = "pairwise.complete.obs"`. When we do that, R only looks at the variables that it's trying to correlate when determining what to drop. So, for instance, since the only missing value for observation 1 of `parenthood2` is for `baby.sleep` R will only drop observation 1 when `baby.sleep` is one of the variables involved: and so R keeps observation 1 when trying to correlate `dan.sleep` and `dan.grump`. When we do it this way, here's what we get:

```
> cor(parenthood2, use = "pairwise.complete.obs")
             dan.sleep  baby.sleep    dan.grump          day
dan.sleep   1.00000000  0.61472303 -0.903442442 -0.076796665
baby.sleep  0.61472303  1.00000000 -0.567802669  0.058309485
dan.grump  -0.90344244 -0.56780267  1.000000000  0.005833399
day        -0.07679667  0.05830949  0.005833399  1.000000000
```

Similar, but not quite the same. It's also worth noting that the `correlate()` function (in the `lsr` package) automatically uses the "pairwise complete" method:

```
> correlate(parenthood2)

CORRELATIONS
============
- correlation type:  pearson
- correlations shown only when both variables are numeric

           dan.sleep baby.sleep dan.grump    day
dan.sleep          .      0.615    -0.903 -0.077
baby.sleep     0.615          .    -0.568  0.058
dan.grump     -0.903     -0.568         .  0.006
day           -0.077      0.058     0.006      .
```

The two approaches have different strengths and weaknesses. The "pairwise complete" approach has the advantage that it keeps more observations, so you're making use of more of your data and (as we'll discuss in tedious detail in Chapter 10 and it improves the reliability of your estimated correlation. On the other hand, it means that every correlation in your correlation matrix is being computed from a slightly different set of observations, which can be awkward when you want to compare the different correlations that you've got.

So which method should you use? It depends a lot on *why* you think your values are missing, and probably depends a little on how paranoid you are. For instance, if you think that the missing values were "chosen" completely randomly[85] then you'll probably want to use the pairwise method. If you think that missing data are a cue to thinking that the whole observation might be rubbish (e.g., someone just selecting arbitrary responses in your questionnaire), but that there's no pattern to which observations are "rubbish" then it's probably safer to keep only those observations that are complete. If you think there's something systematic going on, in that some observations are more likely to be missing than others, then you have a much trickier problem to solve, and one that is beyond the scope of this book.

This page titled 15.7: Handling Missing Values is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **5.8: Handling Missing Values** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

15.8: Testing the Significance of a Correlation

15.8.1 Hypothesis tests for a single correlation

I don't want to spend too much time on this, but it's worth very briefly returning to the point I made earlier, that Pearson correlations are basically the same thing as linear regressions with only a single predictor added to the model. What this means is that the hypothesis tests that I just described in a regression context can also be applied to correlation coefficients. To see this, let's take a `summary()` of the `regression.1` model:

```
summary( regression.1 )
```

```
##
## Call:
## lm(formula = dan.grump ~ dan.sleep, data = parenthood)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.025  -2.213  -0.399   2.681  11.750
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 125.9563     3.0161   41.76   <2e-16 ***
## dan.sleep    -8.9368     0.4285  -20.85   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.332 on 98 degrees of freedom
## Multiple R-squared:  0.8161, Adjusted R-squared:  0.8142
## F-statistic: 434.9 on 1 and 98 DF,  p-value: < 2.2e-16
```

The important thing to note here is the t test associated with the predictor, in which we get a result of $t(98)=-20.85$, $p<.001$. Now let's compare this to the output of a different function, which goes by the name of `cor.test()`. As you might expect, this function runs a hypothesis test to see if the observed correlation between two variables is significantly different from 0. Let's have a look:

```
cor.test( x = parenthood$dan.sleep, y = parenthood$dan.grump )
```

```
##
##  Pearson's product-moment correlation
##
## data:  parenthood$dan.sleep and parenthood$dan.grump
## t = -20.854, df = 98, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9340614 -0.8594714
## sample estimates:
##       cor
## -0.903384
```

Again, the key thing to note is the line that reports the hypothesis test itself, which seems to be saying that t(98)=−20.85, p<.001. Hm. Looks like it's exactly the same test, doesn't it? And that's exactly what it is. The test for the significance of a correlation is identical to the t test that we run on a coefficient in a regression model.

15.8.2 Hypothesis tests for all pairwise correlations

Okay, one more digression before I return to regression properly. In the previous section I talked about the `cor.test()` function, which lets you run a hypothesis test on a single correlation. The `cor.test()` function is (obviously) an extension of the `cor()` function, which we talked about in Section 5.7. However, the `cor()` function isn't restricted to computing a single correlation: you can use it to compute *all* pairwise correlations among the variables in your data set. This leads people to the natural question: can the `cor.test()` function do the same thing? Can we use `cor.test()` to run hypothesis tests for all possible parwise correlations among the variables in a data frame?

The answer is no, and there's a very good reason for this. Testing a single correlation is fine: if you've got some reason to be asking "is A related to B?", then you should absolutely run a test to see if there's a significant correlation. But if you've got variables A, B, C, D and E and you're thinking about testing the correlations among all possible pairs of these, a statistician would want to ask: what's your hypothesis? If you're in the position of wanting to test all possible pairs of variables, then you're pretty clearly on a fishing expedition, hunting around in search of significant effects when you don't actually have a clear research hypothesis in mind. This is *dangerous*, and the authors of `cor.test()` obviously felt that they didn't want to support that kind of behaviour.

On the other hand… a somewhat less hardline view might be to argue we've encountered this situation before, back in Section 14.5 when we talked about *post hoc tests* in ANOVA. When running post hoc tests, we didn't have any specific comparisons in mind, so what we did was apply a correction (e.g., Bonferroni, Holm, etc) in order to avoid the possibility of an inflated Type I error rate. From this perspective, it's okay to run hypothesis tests on all your pairwise correlations, but you must treat them as post hoc analyses, and if so you need to apply a correction for multiple comparisons. That's what the `correlate()` function in the `lsr` package does. When we use the `correlate()` function in Section 5.7 all it did was print out the correlation matrix. But you can get it to output the results of all the pairwise tests as well by specifying `test=TRUE` . Here's what happens with the `parenthood` data:

```
library(lsr)
```

```
## Warning: package 'lsr' was built under R version 3.5.2
```

```
correlate(parenthood, test=TRUE)
```

```
##
## CORRELATIONS
## ============
## - correlation type:  pearson
## - correlations shown only when both variables are numeric
##
##            dan.sleep    baby.sleep    dan.grump       day
## dan.sleep          .         0.628***    -0.903*** -0.098
## baby.sleep     0.628***          .       -0.566*** -0.010
## dan.grump     -0.903***     -0.566***         .     0.076
## day           -0.098        -0.010        0.076         .
##
## ---
## Signif. codes: . = p < .1, * = p<.05, ** = p<.01, *** = p<.001
##
##
## p-VALUES
## ========
## - total number of tests run:  6
## - correction for multiple testing:  holm
##
##            dan.sleep baby.sleep dan.grump   day
## dan.sleep          .      0.000     0.000 0.990
## baby.sleep     0.000          .     0.000 0.990
## dan.grump      0.000      0.000         . 0.990
## day            0.990      0.990     0.990     .
##
##
## SAMPLE SIZES
## ============
##
##            dan.sleep baby.sleep dan.grump day
## dan.sleep        100        100       100 100
## baby.sleep       100        100       100 100
## dan.grump        100        100       100 100
## day              100        100       100 100
```

The output here contains three matrices. First it prints out the correlation matrix. Second it prints out a matrix of p-values, using the Holm method[218] to correct for multiple comparisons. Finally, it prints out a matrix indicating the sample size (number of pairwise complete cases) that contributed to each correlation.

So there you have it. If you really desperately want to do pairwise hypothesis tests on your correlations, the `correlate()` function will let you do it. But please, **please** be careful. I can't count the number of times I've had a student panicking in my office because they've run these pairwise correlation tests, and they get one or two significant results that don't make any sense. For some reason, the moment people see those little significance stars appear, they feel compelled to throw away all common sense and assume that the results must correspond to something real that requires an explanation. In most such cases, my experience has been that the right answer is "it's a Type I error".

This page titled 15.8: Testing the Significance of a Correlation is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **15.6: Testing the Significance of a Correlation** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

15.9: Regarding Regression Coefficients

Before moving on to discuss the assumptions underlying linear regression and what you can do to check if they're being met, there's two more topics I want to briefly discuss, both of which relate to the regression coefficients. The first thing to talk about is calculating confidence intervals for the coefficients; after that, I'll discuss the somewhat murky question of how to determine which of predictor is most important.

15.9.1 Confidence intervals for the coefficients

Like any population parameter, the regression coefficients b cannot be estimated with complete precision from a sample of data; that's part of why we need hypothesis tests. Given this, it's quite useful to be able to report confidence intervals that capture our uncertainty about the true value of b. This is especially useful when the research question focuses heavily on an attempt to find out *how* strongly variable X is related to variable Y, since in those situations the interest is primarily in the regression weight b. Fortunately, confidence intervals for the regression weights can be constructed in the usual fashion,

$$CI(b) = \hat{b} \pm (t_{crit} \; x \; SE(\hat{b}))$$

where SE($\hat{b}$) is the standard error of the regression coefficient, and t_{crit} is the relevant critical value of the appropriate t distribution. For instance, if it's a 95% confidence interval that we want, then the critical value is the 97.5th quantile of a t distribution with N−K−1 degrees of freedom. In other words, this is basically the same approach to calculating confidence intervals that we've used throughout. To do this in R we can use the `confint()` function. There arguments to this function are

- `object` . The regression model (`lm` object) for which confidence intervals are required.
- `parm` . A vector indicating which coefficients we should calculate intervals for. This can be either a vector of numbers or (more usefully) a character vector containing variable names. By default, all coefficients are included, so usually you don't bother specifying this argument.
- `level` . A number indicating the confidence level that should be used. As is usually the case, the default value is 0.95, so you wouldn't usually need to specify this argument.

So, suppose I want 99% confidence intervals for the coefficients in the `regression.2` model. I could do this using the following command:

```
confint( object = regression.2,
         level = .99)
```

```
##                   0.5 %      99.5 %
## (Intercept) 117.9755724 133.9555593
## dan.sleep   -10.4044419  -7.4960575
## baby.sleep   -0.7016868   0.7227357
```

Simple enough.

15.9.2 Calculating standardised regression coefficients

One more thing that you might want to do is to calculate "standardised" regression coefficients, often denoted β. The rationale behind standardised coefficients goes like this. In a lot of situations, your variables are on fundamentally different scales. Suppose, for example, my regression model aims to predict people's IQ scores, using their educational attainment (number of years of education) and their income as predictors. Obviously, educational attainment and income are not on the same scales: the number of years of schooling can only vary by 10s of years, whereas income would vary by 10,000s of dollars (or more). The units of measurement have a big influence on the regression coefficients: the b coefficients only make sense when interpreted in light of the units, both of the predictor variables and the outcome variable. This makes it very difficult to compare the coefficients of different predictors. Yet there are situations where you really do want to make comparisons between different coefficients. Specifically, you might want some kind of standard measure of which predictors have the strongest relationship to the outcome. This is what ***standardised coefficients*** aim to do.

The basic idea is quite simple: the standardised coefficients are the coefficients that you would have obtained if you'd converted all the variables to z-scores before running the regression.[219] The idea here is that, by converting all the predictors to z-scores, they all go into the regression on the same scale, thereby removing the problem of having variables on different scales. Regardless of what the original variables were, a β value of 1 means that an increase in the predictor of 1 standard deviation will produce a corresponding 1 standard deviation increase in the outcome variable. Therefore, if variable A has a larger absolute value of β than variable B, it is deemed to have a stronger relationship with the outcome. Or at least that's the idea: it's worth being a little cautious here, since this does rely very heavily on the assumption that "a 1 standard deviation change" is fundamentally the same kind of thing for all variables. It's not always obvious that this is true.

Leaving aside the interpretation issues, let's look at how it's calculated. What you could do is standardise all the variables yourself and then run a regression, but there's a much simpler way to do it. As it turns out, the β coefficient for a predictor X and outcome Y has a very simple formula, namely

$$\beta_X = b_X \times \frac{\sigma_X}{\sigma_Y}$$

where σ_X is the standard deviation of the predictor, and σY is the standard deviation of the outcome variable Y. This makes matters a lot simpler. To make things even simpler, the `lsr` package includes a function `standardCoefs()` that computes the β coefficients.

```
standardCoefs( regression.2 )
```

```
##                      b        beta
## dan.sleep  -8.95024973 -0.90474809
## baby.sleep  0.01052447  0.00217223
```

This clearly shows that the `dan.sleep` variable has a much stronger effect than the `baby.sleep` variable. However, this is a perfect example of a situation where it would probably make sense to use the original coefficients b rather than the standardised coefficients β. After all, my sleep and the baby's sleep are *already* on the same scale: number of hours slept. Why complicate matters by converting these to z-scores?

This page titled 15.9: Regarding Regression Coefficients is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **15.7: Regarding Regression Coefficients** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

15.10: Assumptions of Regression

The linear regression model that I've been discussing relies on several assumptions. In Section 15.9 we'll talk a lot more about how to check that these assumptions are being met, but first, let's have a look at each of them.

- *Normality*. Like half the models in statistics, standard linear regression relies on an assumption of normality. Specifically, it assumes that the *residuals* are normally distributed. It's actually okay if the predictors X and the outcome Y are non-normal, so long as the residuals ϵ are normal. See Section 15.9.3.
- *Linearity*. A pretty fundamental assumption of the linear regression model is that relationship between X and Y actually be linear! Regardless of whether it's a simple regression or a multiple regression, we assume that the relatiships involved are linear. See Section 15.9.4.
- *Homogeneity of variance*. Strictly speaking, the regression model assumes that each residual ϵ_i is generated from a normal distribution with mean 0, and (more importantly for the current purposes) with a standard deviation σ that is the same for every single residual. In practice, it's impossible to test the assumption that every residual is identically distributed. Instead, what we care about is that the standard deviation of the residual is the same for all values of $\hat{Y}$, and (if we're being especially paranoid) all values of every predictor X in the model. See Section 15.9.5.
- *Uncorrelated predictors*. The idea here is that, is a multiple regression model, you don't want your predictors to be too strongly correlated with each other. This isn't "technically" an assumption of the regression model, but in practice it's required. Predictors that are too strongly correlated with each other (referred to as "collinearity") can cause problems when evaluating the model. See Section 15.9.6
- *Residuals are independent of each other*. This is really just a "catch all" assumption, to the effect that "there's nothing else funny going on in the residuals". If there is something weird (e.g., the residuals all depend heavily on some other unmeasured variable) going on, it might screw things up.
- *No "bad" outliers*. Again, not actually a technical assumption of the model (or rather, it's sort of implied by all the others), but there is an implicit assumption that your regression model isn't being too strongly influenced by one or two anomalous data points; since this raises questions about the adequacy of the model, and the trustworthiness of the data in some cases. See Section 15.9.2.

This page titled 15.10: Assumptions of Regression is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **15.8: Assumptions of Regression** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

15.11: Model Checking

The main focus of this section is ***regression diagnostics***, a term that refers to the art of checking that the assumptions of your regression model have been met, figuring out how to fix the model if the assumptions are violated, and generally to check that nothing "funny" is going on. I refer to this as the "art" of model checking with good reason: it's not easy, and while there are a lot of fairly standardised tools that you can use to diagnose and maybe even cure the problems that ail your model (if there are any, that is!), you really do need to exercise a certain amount of judgment when doing this. It's easy to get lost in all the details of checking this thing or that thing, and it's quite exhausting to try to remember what all the different things are. This has the very nasty side effect that a lot of people get frustrated when trying to learn *all* the tools, so instead they decide not to do *any* model checking. This is a bit of a worry!

In this section, I describe several different things you can do to check that your regression model is doing what it's supposed to. It doesn't cover the full space of things you could do, but it's still much more detailed than what I see a lot of people doing in practice; and I don't usually cover all of this in my intro stats class myself. However, I do think it's important that you get a sense of what tools are at your disposal, so I'll try to introduce a bunch of them here. Finally, I should note that this section draws quite heavily from the Fox and Weisberg (2011) text, the book associated with the `car` package. The `car` package is notable for providing some excellent tools for regression diagnostics, and the book itself talks about them in an admirably clear fashion. I don't want to sound too gushy about it, but I do think that Fox and Weisberg (2011) is well worth reading.

15.11.1 Three kinds of residuals

The majority of regression diagnostics revolve around looking at the residuals, and by now you've probably formed a sufficiently pessimistic theory of statistics to be able to guess that – precisely *because* of the fact that we care a lot about the residuals – there are several different kinds of residual that we might consider. In particular, the following three kinds of residual are referred to in this section: "ordinary residuals", "standardised residuals", and "Studentised residuals". There is a fourth kind that you'll see referred to in some of the Figures, and that's the "Pearson residual": however, for the models that we're talking about in this chapter, the Pearson residual is identical to the ordinary residual.

The first and simplest kind of residuals that we care about are ***ordinary residuals***. These are the actual, raw residuals that I've been talking about throughout this chapter. The ordinary residual is just the difference between the fitted value $\hat{Y}_i$ and the observed value Y_i. I've been using the notation ϵi to refer to the i-th ordinary residual, and by gum I'm going to stick to it. With this in mind, we have the very simple equation

$$\epsilon_i = Y_i - \hat{Y}_i$$

This is of course what we saw earlier, and unless I specifically refer to some other kind of residual, this is the one I'm talking about. So there's nothing new here: I just wanted to repeat myself. In any case, you can get R to output a vector of ordinary residuals, you can use a command like this:

```
residuals( object = regression.2 )
```

```
##           1           2           3           4           5           6
##  -2.1403095   4.7081942   1.9553640  -2.0602806   0.7194888  -0.4066133
##           7           8           9          10          11          12
##   0.2269987  -1.7003077   0.2025039   3.8524589   3.9986291  -4.9120150
##          13          14          15          16          17          18
##   1.2060134   0.4946578  -2.6579276  -0.3966805   3.3538613   1.7261225
##          19          20          21          22          23          24
##  -0.4922551  -5.6405941  -0.4660764   2.7238389   9.3653697   0.2841513
##          25          26          27          28          29          30
##  -0.5037668  -1.4941146   8.1328623   1.9787316  -1.5126726   3.5171148
##          31          32          33          34          35          36
##  -8.9256951  -2.8282946   6.1030349  -7.5460717   4.5572128 -10.6510836
##          37          38          39          40          41          42
##  -5.6931846   6.3096506  -2.1082466  -0.5044253   0.1875576   4.8094841
##          43          44          45          46          47          48
##  -5.4135163  -6.2292842  -4.5725232  -5.3354601   3.9950111   2.1718745
##          49          50          51          52          53          54
##  -3.4766440   0.4834367   6.2839790   2.0109396  -1.5846631  -2.2166613
##          55          56          57          58          59          60
##   2.2033140   1.9328736  -1.8301204  -1.5401430   2.5298509  -3.3705782
##          61          62          63          64          65          66
##  -2.9380806   0.6590736  -0.5917559  -8.6131971   5.9781035   5.9332979
##          67          68          69          70          71          72
##  -1.2341956   3.0047669  -1.0802468   6.5174672  -3.0155469   2.1176720
##          73          74          75          76          77          78
##   0.6058757  -2.7237421  -2.2291472  -1.4053822   4.7461491  11.7495569
##          79          80          81          82          83          84
##   4.7634141   2.6620908 -11.0345292  -0.7588667   1.4558227  -0.4745727
##          85          86          87          88          89          90
##   8.9091201  -1.1409777   0.7555223  -0.4107130   0.8797237  -1.4095586
##          91          92          93          94          95          96
##   3.1571385  -3.4205757  -5.7228699  -2.2033958  -3.8647891   0.4982711
##          97          98          99         100
##  -5.5249495   4.1134221  -8.2038533   5.6800859
```

One drawback to using ordinary residuals is that they're always on a different scale, depending on what the outcome variable is and how good the regression model is. That is, Unless you've decided to run a regression model without an intercept term, the ordinary residuals will have mean 0; but the variance is different for every regression. In a lot of contexts, especially where you're only interested in the *pattern* of the residuals and not their actual values, it's convenient to estimate the ***standardised residuals***, which are normalised in such a way as to have standard deviation 1. The way we calculate these is to divide the ordinary residual by an estimate of the (population) standard deviation of these residuals. For technical reasons, mumble mumble, the formula for this is:

$$\epsilon_i' = \frac{\epsilon_i}{\hat{\sigma}\sqrt{1-h_i}}$$

where $\hat{\sigma}$ in this context is the estimated population standard deviation of the ordinary residuals, and h_i is the "hat value" of the ith observation. I haven't explained hat values to you yet (but have no fear,[220] it's coming shortly), so this won't make a lot of sense. For now, it's enough to interpret the standardised residuals as if we'd converted the ordinary residuals to z-scores. In fact, that is more or less the truth, it's just that we're being a bit fancier. To get the standardised residuals, the command you want is this:

```
rstandard( model = regression.2 )
```

```
##           1           2           3           4           5           6
## -0.49675845  1.10430571  0.46361264 -0.47725357  0.16756281 -0.09488969
##           7           8           9          10          11          12
##  0.05286626 -0.39260381  0.04739691  0.89033990  0.95851248 -1.13898701
##          13          14          15          16          17          18
##  0.28047841  0.11519184 -0.61657092 -0.09191865  0.77692937  0.40403495
##          19          20          21          22          23          24
## -0.11552373 -1.31540412 -0.10819238  0.62951824  2.17129803  0.06586227
##          25          26          27          28          29          30
## -0.11980449 -0.34704024  1.91121833  0.45686516 -0.34986350  0.81233165
##          31          32          33          34          35          36
## -2.08659993 -0.66317843  1.42930082 -1.77763064  1.07452436 -2.47385780
##          37          38          39          40          41          42
## -1.32715114  1.49419658 -0.49115639 -0.11674947  0.04401233  1.11881912
##          43          44          45          46          47          48
## -1.27081641 -1.46422595 -1.06943700 -1.24659673  0.94152881  0.51069809
##          49          50          51          52          53          54
## -0.81373349  0.11412178  1.47938594  0.46437962 -0.37157009 -0.51609949
##          55          56          57          58          59          60
##  0.51800753  0.44813204 -0.42662358 -0.35575611  0.58403297 -0.78022677
##          61          62          63          64          65          66
## -0.67833325  0.15484699 -0.13760574 -2.05662232  1.40238029  1.37505125
##          67          68          69          70          71          72
## -0.28964989  0.69497632 -0.24945316  1.50709623 -0.69864682  0.49071427
##          73          74          75          76          77          78
##  0.14267297 -0.63246560 -0.51972828 -0.32509811  1.10842574  2.72171671
##          79          80          81          82          83          84
##  1.09975101  0.62057080 -2.55172097 -0.17584803  0.34340064 -0.11158952
##          85          86          87          88          89          90
##  2.10863391 -0.26386516  0.17624445 -0.09504416  0.20450884 -0.32730740
##          91          92          93          94          95          96
##  0.73475640 -0.79400855 -1.32768248 -0.51940736 -0.91512580  0.11661226
##          97          98          99         100
## -1.28069115  0.96332849 -1.90290258  1.31368144
```

Note that this function uses a different name for the input argument, but it's still just a linear regression object that the function wants to take as its input here.

The third kind of residuals are ***Studentised residuals*** (also called "jackknifed residuals") and they're even fancier than standardised residuals. Again, the idea is to take the ordinary residual and divide it by some quantity in order to estimate some standardised notion of the residual, but the formula for doing the calculations this time is subtly different:

$$\epsilon_i^* = \frac{\epsilon_i}{\hat{\sigma}_{(-i)}\sqrt{1-h_i}}$$

Notice that our estimate of the standard deviation here is written $\hat{\sigma}_{-i}$. What this corresponds to is the estimate of the residual standard deviation that you *would have obtained*, if you just deleted the ith observation from the data set. This sounds like the sort of thing that would be a nightmare to calculate, since it seems to be saying that you have to run N new regression models (even a modern computer might grumble a bit at that, especially if you've got a large data set). Fortunately, some terribly clever person has shown that this standard deviation estimate is actually given by the following equation:

$$\hat{\sigma}_{(-i)} = \hat{\sigma}\sqrt{\frac{N-K-1-\epsilon_i'^2}{N-K-2}}$$

Isn't that a pip? Anyway, the command that you would use if you wanted to pull out the Studentised residuals for our regression model is

```
rstudent( model = regression.2 )
```

```
##           1           2           3           4           5           6
## -0.49482102  1.10557030  0.46172854 -0.47534555  0.16672097 -0.09440368
##           7           8           9          10          11          12
##  0.05259381 -0.39088553  0.04715251  0.88938019  0.95810710 -1.14075472
##          13          14          15          16          17          18
##  0.27914212  0.11460437 -0.61459001 -0.09144760  0.77533036  0.40228555
##          19          20          21          22          23          24
## -0.11493461 -1.32043609 -0.10763974  0.62754813  2.21456485  0.06552336
##          25          26          27          28          29          30
## -0.11919416 -0.34546127  1.93818473  0.45499388 -0.34827522  0.81089646
##          31          32          33          34          35          36
## -2.12403286 -0.66125192  1.43712830 -1.79797263  1.07539064 -2.54258876
##          37          38          39          40          41          42
## -1.33244515  1.50388257 -0.48922682 -0.11615428  0.04378531  1.12028904
##          43          44          45          46          47          48
## -1.27490649 -1.47302872 -1.07023828 -1.25020935  0.94097261  0.50874322
##          49          50          51          52          53          54
## -0.81230544  0.11353962  1.48863006  0.46249410 -0.36991317 -0.51413868
##          55          56          57          58          59          60
##  0.51604474  0.44627831 -0.42481754 -0.35414868  0.58203894 -0.77864171
##          61          62          63          64          65          66
## -0.67643392  0.15406579 -0.13690795 -2.09211556  1.40949469  1.38147541
##          67          68          69          70          71          72
## -0.28827768  0.69311245 -0.24824363  1.51717578 -0.69679156  0.48878534
##          73          74          75          76          77          78
##  0.14195054 -0.63049841 -0.51776374 -0.32359434  1.10974786  2.81736616
##          79          80          81          82          83          84
##  1.10095270  0.61859288 -2.62827967 -0.17496714  0.34183379 -0.11101996
##          85          86          87          88          89          90
##  2.14753375 -0.26259576  0.17536170 -0.09455738  0.20349582 -0.32579584
##          91          92          93          94          95          96
##  0.73300184 -0.79248469 -1.33298848 -0.51744314 -0.91435205  0.11601774
##          97          98          99         100
## -1.28498273  0.96296745 -1.92942389  1.31867548
```

Before moving on, I should point out that you don't often need to manually extract these residuals yourself, even though they are at the heart of almost all regression diagnostics. That is, the `residuals()`, `rstandard()` and `rstudent()` functions are all useful to *know* about, but most of the time the various functions that run the diagnostics will take care of these calculations for you. Even so, it's always nice to know how to actually get hold of these things yourself in case you ever need to do something non-standard.

15.11.2 Three kinds of anomalous data

One danger that you can run into with linear regression models is that your analysis might be disproportionately sensitive to a smallish number of "unusual" or "anomalous" observations. I discussed this idea previously in Section 6.5.2 in the context of discussing the outliers that get automatically identified by the `boxplot()` function, but this time we need to be much more precise. In the context of linear regression, there are three conceptually distinct ways in which an observation might be called "anomalous". All three are interesting, but they have rather different implications for your analysis.

The first kind of unusual observation is an ***outlier***. The definition of an outlier (in this context) is an observation that is very different from what the regression model predicts. An example is shown in Figure 15.7. In practice, we operationalise this concept by saying that an outlier is an observation that has a very large Studentised residual, ϵ_i^*. Outliers are interesting: a big outlier *might* correspond to junk data – e.g., the variables might have been entered incorrectly, or some other defect may be detectable. Note that you shouldn't throw an observation away just because it's an outlier. But the fact that it's an outlier is often a cue to look more closely at that case, and try to find out why it's so different.

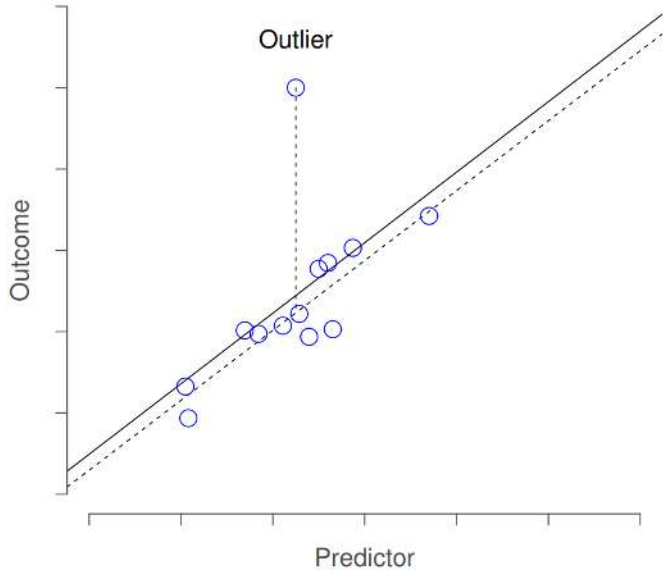


Figure 15.7: An illustration of outliers. The dotted lines plot the regression line that would have been estimated without the anomalous observation included, and the corresponding residual (i.e., the Studentised residual). The solid line shows the regression line with the anomalous observation included. The outlier has an unusual value on the outcome (y axis location) but not the predictor (x axis location), and lies a long way from the regression line.

The second way in which an observation can be unusual is if it has high ***leverage***: this happens when the observation is very different from all the other observations. This doesn't necessarily have to correspond to a large residual: if the observation happens to be unusual on all variables in precisely the same way, it can actually lie very close to the regression line. An example of this is shown in Figure 15.8. The leverage of an observation is operationalised in terms of its *hat value*, usually written hi. The formula for the hat value is rather complicated[221] but its interpretation is not: h_i is a measure of the extent to which the i-th observation is "in control" of where the regression line ends up going. You can extract the hat values using the following command:

```
hatvalues( model = regression.2 )
```


https://stats.libretexts.org/@go/page/36203

```
##           1           2           3           4           5           6
## 0.02067452 0.04105320 0.06155445 0.01685226 0.02734865 0.03129943
##           7           8           9          10          11          12
## 0.02735579 0.01051224 0.03698976 0.01229155 0.08189763 0.01882551
##          13          14          15          16          17          18
## 0.02462902 0.02718388 0.01964210 0.01748592 0.01691392 0.03712530
##          19          20          21          22          23          24
## 0.04213891 0.02994643 0.02099435 0.01233280 0.01853370 0.01804801
##          25          26          27          28          29          30
## 0.06722392 0.02214927 0.04472007 0.01039447 0.01381812 0.01105817
##          31          32          33          34          35          36
## 0.03468260 0.04048248 0.03814670 0.04934440 0.05107803 0.02208177
##          37          38          39          40          41          42
## 0.02919013 0.05928178 0.02799695 0.01519967 0.04195751 0.02514137
##          43          44          45          46          47          48
## 0.04267879 0.04517340 0.03558080 0.03360160 0.05019778 0.04587468
##          49          50          51          52          53          54
## 0.03701290 0.05331282 0.04814477 0.01072699 0.04047386 0.02681315
##          55          56          57          58          59          60
## 0.04556787 0.01856997 0.02919045 0.01126069 0.01012683 0.01546412
##          61          62          63          64          65          66
## 0.01029534 0.04428870 0.02438944 0.07469673 0.04135090 0.01775697
##          67          68          69          70          71          72
## 0.04217616 0.01384321 0.01069005 0.01340216 0.01716361 0.01751844
##          73          74          75          76          77          78
## 0.04863314 0.02158623 0.02951418 0.01411915 0.03276064 0.01684599
##          79          80          81          82          83          84
## 0.01028001 0.02920514 0.01348051 0.01752758 0.05184527 0.04583604
##          85          86          87          88          89          90
## 0.05825858 0.01359644 0.03054414 0.01487724 0.02381348 0.02159418
##          91          92          93          94          95          96
## 0.02598661 0.02093288 0.01982480 0.05063492 0.05907629 0.03682026
##          97          98          99         100
## 0.01817919 0.03811718 0.01945603 0.01373394
```

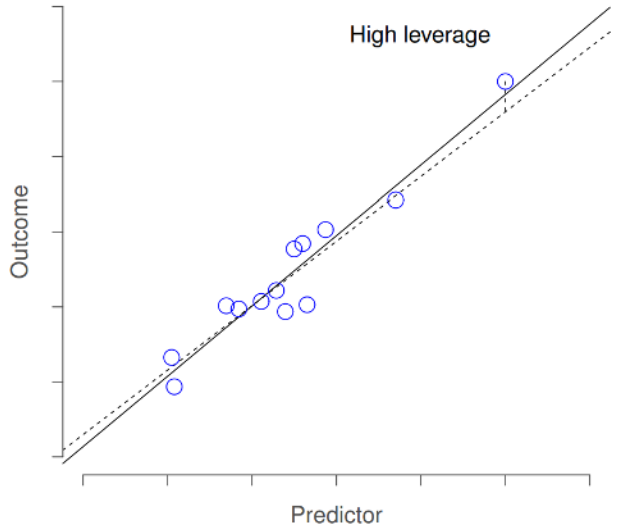


Figure 15.8: An illustration of high leverage points. The anomalous observation in this case is unusual both in terms of the predictor (x axis) and the outcome (y axis), but this unusualness is highly consistent with the pattern of correlations that exists among the other observations; as a consequence, the observation falls very close to the regression line and does not distort it.

In general, if an observation lies far away from the other ones in terms of the predictor variables, it will have a large hat value (as a rough guide, high leverage is when the hat value is more than 2-3 times the average; and note that the sum of the hat values is constrained to be equal to K+1). High leverage points are also worth looking at in more detail, but they're much less likely to be a cause for concern unless they are also outliers. % guide from Venables and Ripley.

This brings us to our third measure of unusualness, the ***influence*** of an observation. A high influence observation is an outlier that has high leverage. That is, it is an observation that is very different to all the other ones in some respect, and also lies a long way from the regression line. This is illustrated in Figure 15.9. Notice the contrast to the previous two figures: outliers don't move the regression line much, and neither do high leverage points. But something that is an outlier and has high leverage… that has a big effect on the regression line.

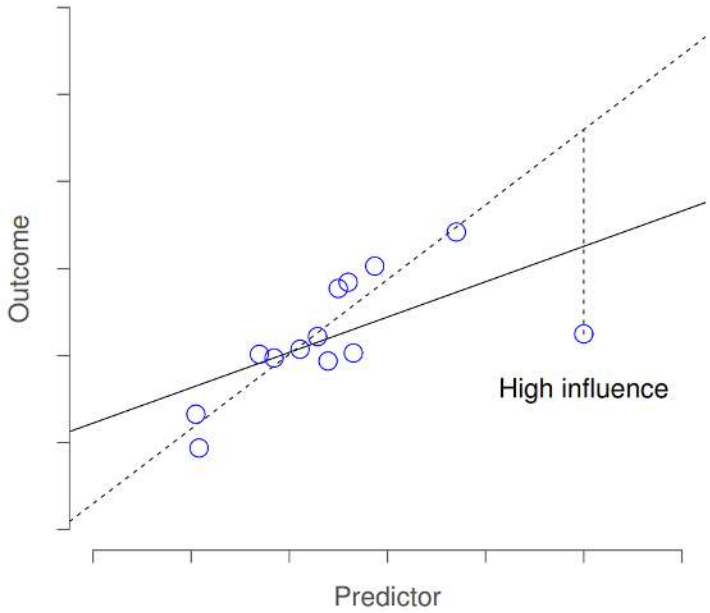


Figure 15.9: An illustration of high influence points. In this case, the anomalous observation is highly unusual on the predictor variable (x axis), and falls a long way from the regression line. As a consequence, the regression line is highly distorted, even though (in this case) the anomalous observation is entirely typical in terms of the outcome variable (y axis).

That's why we call these points high influence; and it's why they're the biggest worry. We operationalise influence in terms of a measure known as ***Cook's distance***,

$$D_i = \frac{\epsilon_i^{*2}}{K+1} \times \frac{h_i}{1-h_i}$$

Notice that this is a multiplication of something that measures the outlier-ness of the observation (the bit on the left), and something that measures the leverage of the observation (the bit on the right). In other words, in order to have a large Cook's distance, an observation must be a fairly substantial outlier *and* have high leverage. In a stunning turn of events, you can obtain these values using the following command:

```
cooks.distance( model = regression.2 )
```

```
##            1            2            3            4            5
## 1.736512e-03 1.740243e-02 4.699370e-03 1.301417e-03 2.631557e-04
##            6            7            8            9           10
## 9.697585e-05 2.620181e-05 5.458491e-04 2.876269e-05 3.288277e-03
##           11           12           13           14           15
## 2.731835e-02 8.296919e-03 6.621479e-04 1.235956e-04 2.538915e-03
##           16           17           18           19           20
## 5.012283e-05 3.461742e-03 2.098055e-03 1.957050e-04 1.780519e-02
##           21           22           23           24           25
## 8.367377e-05 1.649478e-03 2.967594e-02 2.657610e-05 3.448032e-04
##           26           27           28           29           30
## 9.093379e-04 5.699951e-02 7.307943e-04 5.716998e-04 2.459564e-03
##           31           32           33           34           35
## 5.214331e-02 6.185200e-03 2.700686e-02 5.467345e-02 2.071643e-02
##           36           37           38           39           40
## 4.606378e-02 1.765312e-02 4.689817e-02 2.316122e-03 7.012530e-05
##           41           42           43           44           45
## 2.827824e-05 1.076083e-02 2.399931e-02 3.381062e-02 1.406498e-02
##           46           47           48           49           50
## 1.801086e-02 1.561699e-02 4.179986e-03 8.483514e-03 2.444787e-04
##           51           52           53           54           55
## 3.689946e-02 7.794472e-04 1.941235e-03 2.446230e-03 4.270361e-03
##           56           57           58           59           60
## 1.266609e-03 1.824212e-03 4.804705e-04 1.163181e-03 3.187235e-03
##           61           62           63           64           65
## 1.595512e-03 3.703826e-04 1.577892e-04 1.138165e-01 2.827715e-02
##           66           67           68           69           70
## 1.139374e-02 1.231422e-03 2.260006e-03 2.241322e-04 1.028479e-02
##           71           72           73           74           75
## 2.841329e-03 1.431223e-03 3.468538e-04 2.941757e-03 2.738249e-03
##           76           77           78           79           80
## 5.045357e-04 1.387108e-02 4.230966e-02 4.187440e-03 3.861831e-03
##           81           82           83           84           85
## 2.965826e-02 1.838888e-04 2.149369e-03 1.993929e-04 9.168733e-02
##           86           87           88           89           90
## 3.198994e-04 3.262192e-04 4.547383e-05 3.400893e-04 7.881487e-04
##           91           92           93           94           95
## 4.801204e-03 4.493095e-03 1.188427e-02 4.796360e-03 1.752666e-02
##           96           97           98           99          100
## 1.732793e-04 1.012302e-02 1.225818e-02 2.394964e-02 8.010508e-03
```

As a rough guide, Cook's distance greater than 1 is often considered large (that's what I typically use as a quick and dirty rule), though a quick scan of the internet and a few papers suggests that 4/N has also been suggested as a possible rule of thumb.

As hinted above, you don't usually need to make use of these functions, since you can have R automatically draw the critical plots.[222] For the `regression.2` model, these are the plots showing Cook's distance (Figure 15.10) and the more detailed breakdown showing the scatter plot of the Studentised residual against leverage (Figure 15.11). To draw these, we can use the `plot()` function. When the main argument `x` to this function is a linear model object, it will draw one of six different plots, each of which is quite useful for doing regression diagnostics. You specify which one you want using the `which` argument (a number between 1 and 6). If you don't do this then R will draw all six. The two plots of interest to us in this context are generated using the following commands:

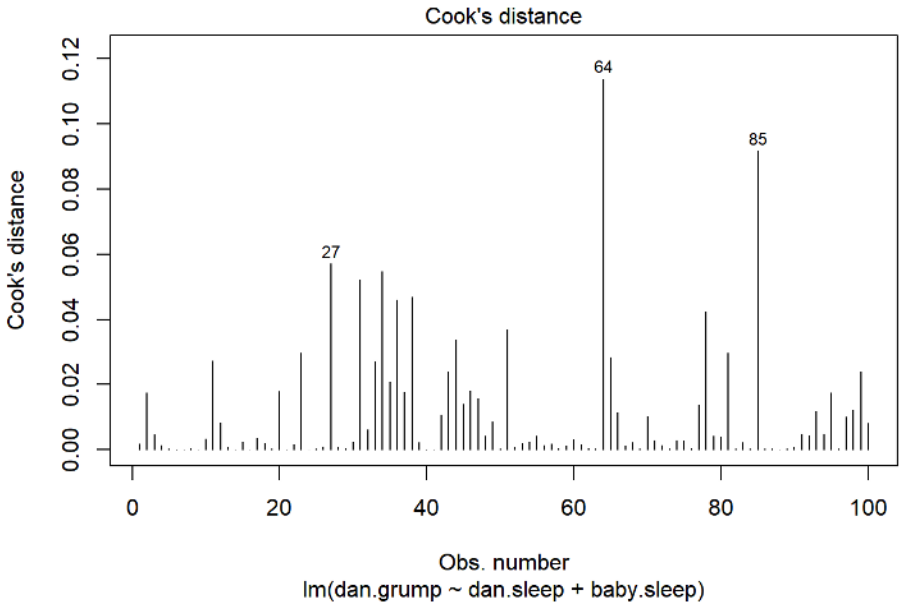


Figure 15.10: Cook's distance for every observation. This is one of the standard regression plots produced by the `plot()` function when the input is a linear regression object. It is obtained by setting `which=4`

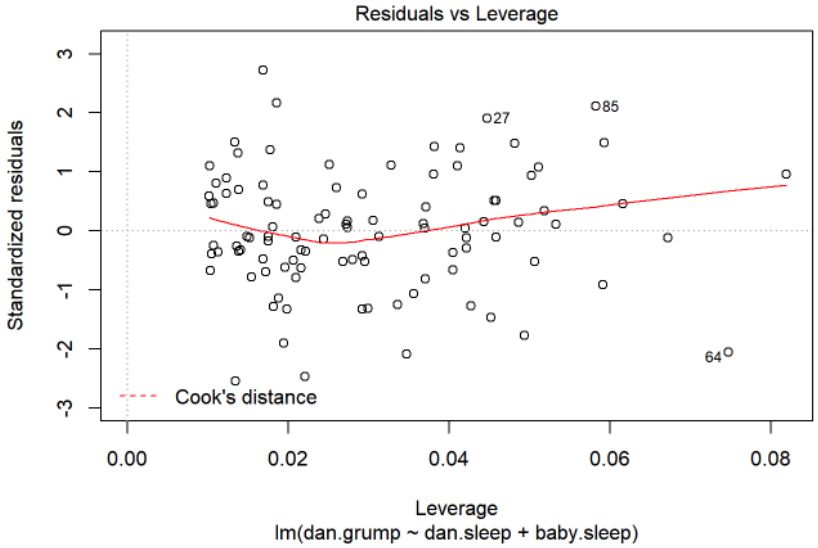


Figure 15.11: Residuals versus leverage. This is one of the standard regression plots produced by the `plot()` function when the input is a linear regression object. It is obtained by setting `which=5` .

An obvious question to ask next is, if you do have large values of Cook's distance, what should you do? As always, there's no hard and fast rules. Probably the first thing to do is to try running the regression with that point excluded and see what happens to the model performance and to the regression coefficients. If they really are substantially different, it's time to start digging into your

data set and your notes that you no doubt were scribbling as your ran your study; try to figure out *why* the point is so different. If you start to become convinced that this one data point is badly distorting your results, you might consider excluding it, but that's less than ideal unless you have a solid explanation for why this particular case is qualitatively different from the others and therefore deserves to be handled separately.[223] To give an example, let's delete the observation from day 64, the observation with the largest Cook's distance for the `regression.2` model. We can do this using the `subset` argument:

```
lm( formula = dan.grump ~ dan.sleep + baby.sleep,  # same formula
    data = parenthood,       # same data frame...
    subset = -64             # ...but observation 64 is deleted
 )
```

```
##
## Call:
## lm(formula = dan.grump ~ dan.sleep + baby.sleep, data = parenthood,
##     subset = -64)
##
## Coefficients:
## (Intercept)    dan.sleep   baby.sleep
##    126.3553      -8.8283      -0.1319
```

As you can see, those regression coefficients have barely changed in comparison to the values we got earlier. In other words, we really don't have any problem as far as anomalous data are concerned.

15.11.3 Checking the normality of the residuals

Like many of the statistical tools we've discussed in this book, regression models rely on a normality assumption. In this case, we assume that the residuals are normally distributed. The tools for testing this aren't fundamentally different to those that we discussed earlier in Section 13.9. Firstly, I firmly believe that it never hurts to draw an old fashioned histogram. The command I use might be something like this:

```
hist( x = residuals( regression.2 ),   # data are the residuals
      xlab = "Value of residual",      # x-axis label
      main = "",                       # no title
      breaks = 20                      # lots of breaks
 )
```

The resulting plot is shown in Figure 15.12, and as you can see the plot looks pretty damn close to normal, almost unnaturally so.

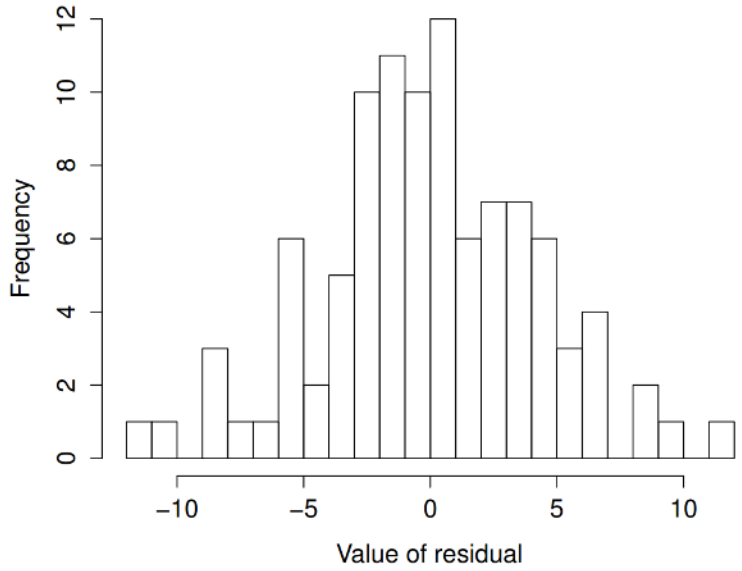


Figure 15.12: A histogram of the (ordinary) residuals in the `regression.2` model. These residuals look very close to being normally distributed, much moreso than is typically seen with real data. This shouldn't surprise you… they aren't real data, and they aren't real residuals!

I could also run a Shapiro-Wilk test to check, using the `shapiro.test()` function; the W value of .99, at this sample size, is non-significant (p=.84), again suggesting that the normality assumption isn't in any danger here. As a third measure, we might also want to draw a QQ-plot using the `qqnorm()` function. The QQ plot is an excellent one to draw, and so you might not be surprised to discover that it's one of the regression plots that we can produce using the `plot()` function:

```
plot( x = regression.2, which = 2 )   # Figure @ref{fig:regressionplot2}
```

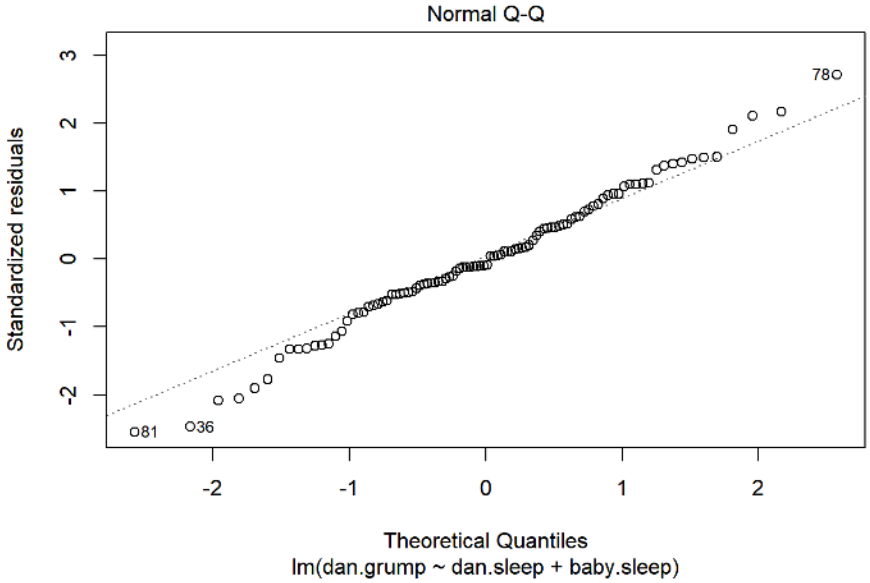


Figure 15.13: Plot of the theoretical quantiles according to the model, against the quantiles of the standardised residuals. This is one of the standard regression plots produced by the `plot()` function when the input is a linear regression object. It is obtained by setting `which=2` .

The output is shown in Figure 15.13, showing the standardised residuals plotted as a function of their theoretical quantiles according to the regression model. The fact that the output appends the model specification to the picture is nice.

15.11.4 Checking the linearity of the relationship

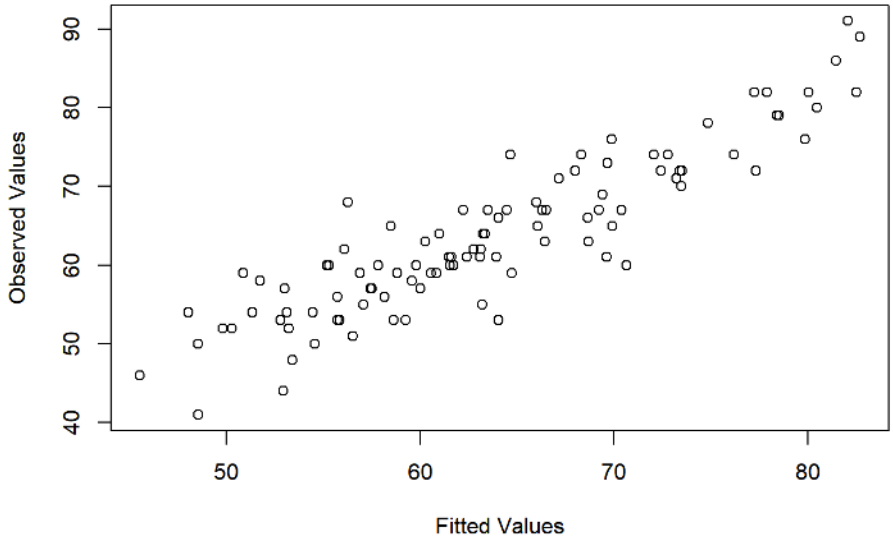


Figure 15.14: Plot of the fitted values against the observed values of the outcome variable. A straight line is what we're hoping to see here. This looks pretty good, suggesting that there's nothing grossly wrong, but there could be hidden subtle issues.

The third thing we might want to test is the linearity of the relationships between the predictors and the outcomes. There's a few different things that you might want to do in order to check this. Firstly, it never hurts to just plot the relationship between the fitted values $\hat{Y}_i$ and the observed values Y_i for the outcome variable, as illustrated in Figure 15.14. To draw this we could use the `fitted.values()` function to extract the $\hat{Y}_i$ values in much the same way that we used the `residuals()` function to extract the ϵ_i values. So the commands to draw this figure might look like this:

```
yhat.2 <- fitted.values( object = regression.2 )
plot( x = yhat.2,
      y = parenthood$dan.grump,
      xlab = "Fitted Values",
      ylab = "Observed Values"
)
```

One of the reasons I like to draw these plots is that they give you a kind of "big picture view". If this plot looks approximately linear, then we're probably not doing too badly (though that's not to say that there aren't problems). However, if you can see big departures from linearity here, then it strongly suggests that you need to make some changes.

In any case, in order to get a more detailed picture it's often more informative to look at the relationship between the fitted values and the residuals themselves. Again, we could draw this plot using low level commands, but there's an easier way. Just `plot()` the regression model, and select `which = 1` :

```
plot(x = regression.2, which = 1)
```

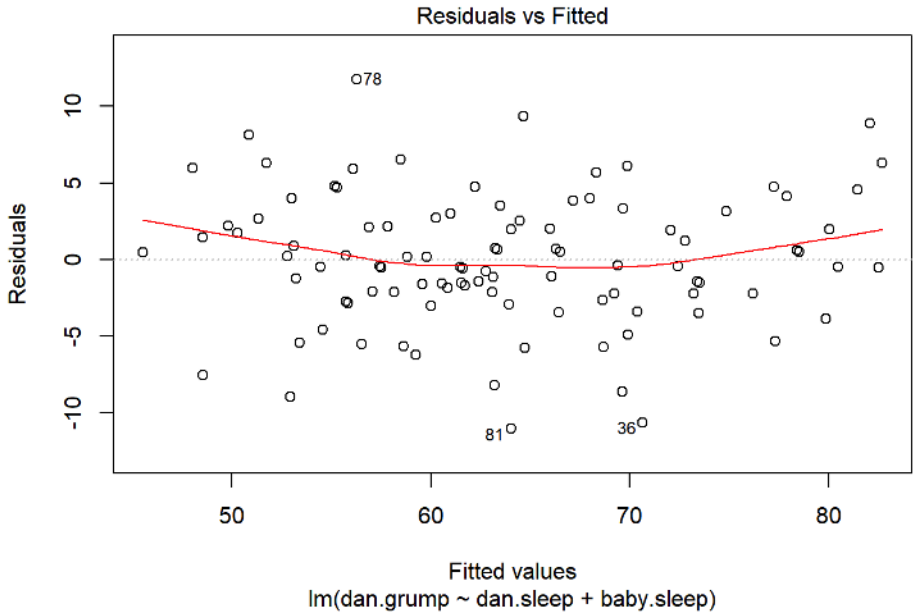


Figure 15.15: Plot of the fitted values against the residuals for `regression.2`, with a line showing the relationship between the two. If this is horizontal and straight, then we can feel reasonably confident that the "average residual" for all "fitted values" is more or less the same. This is one of the standard regression plots produced by the `plot()` function when the input is a linear regression object. It is obtained by setting `which=1`.

The output is shown in Figure 15.15. As you can see, not only does it draw the scatterplot showing the fitted value against the residuals, it also plots a line through the data that shows the relationship between the two. Ideally, this should be a straight, perfectly horizontal line. There's some hint of curvature here, but it's not clear whether or not we be concerned.

A somewhat more advanced version of the same plot is produced by the `residualPlots()` function in the `car` package. This function not only draws plots comparing the fitted values to the residuals, it does so for each individual predictor. The command is and the resulting plots are shown in Figure 15.16.

```
residualPlots( model = regression.2 )
```

```
## Loading required package: carData
```

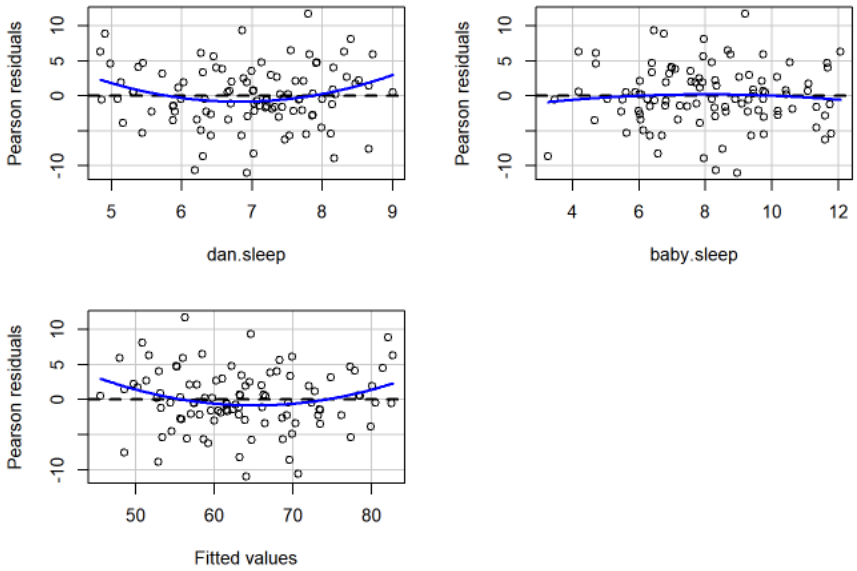


Figure 15.16: Plot of the fitted values against the residuals for `regression.2`, along with similar plots for the two predictors individually. This plot is produced by the `residualPlots()` function in the `car` package. Note that it refers to the residuals as "Pearson residuals", but in this context these are the same as ordinary residuals.

```
##             Test stat Pr(>|Test stat|)
## dan.sleep      2.1604          0.03323 *
## baby.sleep    -0.5445          0.58733
## Tukey test     2.1615          0.03066 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that this function also reports the results of a bunch of ***curvature tests***. For a predictor variable X in some regression model, this test is equivalent to adding a new predictor to the model corresponding to X^2, and running the t-test on the b coefficient associated with this new predictor. If it comes up significant, it implies that there is some nonlinear relationship between the variable and the residuals.

The third line here is the ***Tukey test***, which is basically the same test, except that instead of squaring one of the predictors and adding it to the model, you square the fitted-value. In any case, the fact that the curvature tests have come up significant is hinting that the curvature that we can see in Figures 15.15 and 15.16 is genuine;[224] although it still bears remembering that the pattern in Figure 15.14 is pretty damn straight: in other words the deviations from linearity are pretty small, and probably not worth worrying about.

In a lot of cases, the solution to this problem (and many others) is to transform one or more of the variables. We discussed the basics of variable transformation in Sections 7.2 and (mathfunc), but I do want to make special note of one additional possibility that I didn't mention earlier: the Box-Cox transform. The Box-Cox function is a fairly simple one, but it's very widely used

$$f(x, \lambda) = \frac{x^\lambda - 1}{\lambda}$$

for all values of λ except λ=0. When λ=0 we just take the natural logarithm (i.e., ln(x)). You can calculate it using the `boxCox()` function in the `car` package. Better yet, if what you're trying to do is convert a data to normal, or as normal as possible, there's the `powerTransform()` function in the `car` package that can estimate the best value of λ. Variable transformation is another topic that deserves a fairly detailed treatment, but (again) due to deadline constraints, it will have to wait until a future version of this book.

15.11.5 Checking the homogeneity of variance

The regression models that we've talked about all make a homogeneity of variance assumption: the variance of the residuals is assumed to be constant. The "default" plot that R provides to help with doing this (`which = 3` when using `plot()`) shows a plot of the square root of the size of the residual $\sqrt{|\epsilon_i|}$, as a function of the fitted value $\hat{Y}_i$. We can produce the plot using the following command,

```
plot(x = regression.2, which = 3)
```

and the resulting plot is shown in Figure 15.17. Note that this plot actually uses the standardised residuals (i.e., converted to z scores) rather than the raw ones, but it's immaterial from our point of view. What we're looking to see here is a straight, horizontal line running through the middle of the plot.

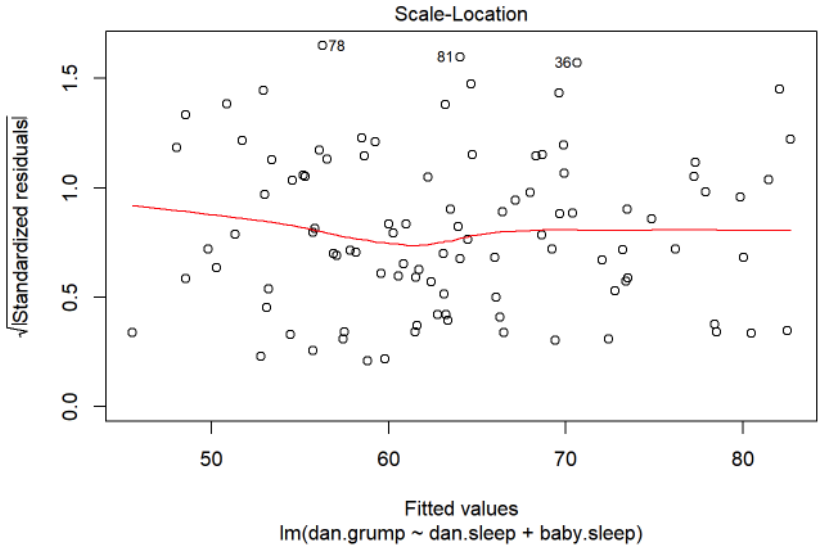


Figure 15.17: Plot of the fitted values (model predictions) against the square root of the abs standardised residuals. This plot is used to diagnose violations of homogeneity of variance. If the variance is really constant, then the line through the middle should be horizontal and flat. This is one of the standard regression plots produced by the `plot()` function when the input is a linear regression object. It is obtained by setting `which=3` .

A slightly more formal approach is to run hypothesis tests. The `car` package provides a function called `ncvTest()` (***non-constant variance test***) that can be used for this purpose (Cook and Weisberg 1983). I won't explain the details of how it works, other than to say that the idea is that what you do is run a regression to see if there is a relationship between the squared residuals ϵ_i and the fitted values $\hat{Y}_i$, or possibly to run a regression using all of the original predictors instead of just $\hat{Y}_i$.[225] Using the default settings, the `ncvTest()` looks for a relationship between $\hat{Y}_i$ and the variance of the residuals, making it a straightforward analogue of Figure 15.17. So if we run it for our model,

```
ncvTest( regression.2 )
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.09317511, Df = 1, p = 0.76018
```

We see that our original impression was right: there's no violations of homogeneity of variance in this data.

It's a bit beyond the scope of this chapter to talk too much about how to deal with violations of homogeneity of variance, but I'll give you a quick sense of what you need to consider. The *main* thing to worry about, if homogeneity of variance is violated, is that the standard error estimates associated with the regression coefficients are no longer entirely reliable, and so your t tests for the coefficients aren't quite right either. A simple fix to the problem is to make use of a "heteroscedasticity corrected covariance

matrix" when estimating the standard errors. These are often called ***sandwich estimators***, for reasons that only make sense if you understand the maths at a low level[226] have implemented as the default in the `hccm()` function is a tweak on this, proposed by Long and Ervin (2000). This version uses $\Sigma = \text{diag}\left(\epsilon_i^2 / (1 - h_i^2)\right)$, where hi is the ith hat value. Gosh, regression is *fun*, isn't it?] You don't need to understand what this means (not for an introductory class), but it might help to note that there's a `hccm()` function in the `car()` package that does it. Better yet, you don't even need to use it. You can use the `coeftest()` function in the `lmtest` package, but you need the `car` package loaded:

```
library(lmtest)
library(car)
coeftest( regression.2, vcov= hccm )
```

```
##
## t test of coefficients:
##
##               Estimate Std. Error  t value Pr(>|t|)
## (Intercept) 125.965566   3.247285  38.7910   <2e-16 ***
## dan.sleep    -8.950250   0.615820 -14.5339   <2e-16 ***
## baby.sleep    0.010524   0.291565   0.0361   0.9713
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Not surprisingly, these t tests are pretty much identical to the ones that we saw when we used the `summary(regression.2)` command earlier; because the homogeneity of variance assumption wasn't violated. But if it had been, we might have seen some more substantial differences.

15.11.6 Checking for collinearity

The last kind of regression diagnostic that I'm going to discuss in this chapter is the use of ***variance inflation factors*** (VIFs), which are useful for determining whether or not the predictors in your regression model are too highly correlated with each other. There is a variance inflation factor associated with each predictor X_k in the model, and the formula for the k-th VIF is:

$$\text{VIF}_k = \frac{1}{1 - R^2_{(-k)}}$$

where $R^2_{(-k)}$ refers to R-squared value you would get if you ran a regression using X_k as the outcome variable, and all the other X variables as the predictors. The idea here is that $R^2_{(-k)}$ is a very good measure of the extent to which X_k is correlated with all the other variables in the model. Better yet, the square root of the VIF is pretty interpretable: it tells you how much wider the confidence interval for the corresponding coefficient b_k is, relative to what you would have expected if the predictors are all nice and uncorrelated with one another. If you've only got two predictors, the VIF values are always going to be the same, as we can see if we use the `vif()` function (`car` package)…

```
vif( mod = regression.2 )
```

```
##  dan.sleep baby.sleep
##   1.651038   1.651038
```

And since the square root of 1.65 is 1.28, we see that the correlation between our two predictors isn't causing much of a problem.

To give a sense of how we could end up with a model that has bigger collinearity problems, suppose I were to run a much less interesting regression model, in which I tried to predict the `day` on which the data were collected, as a function of all the other variables in the data set. To see why this would be a bit of a problem, let's have a look at the correlation matrix for all four variables:

```
cor( parenthood )
```

```
##              dan.sleep  baby.sleep   dan.grump         day
## dan.sleep   1.00000000  0.62794934 -0.90338404 -0.09840768
## baby.sleep  0.62794934  1.00000000 -0.56596373 -0.01043394
## dan.grump  -0.90338404 -0.56596373  1.00000000  0.07647926
## day        -0.09840768 -0.01043394  0.07647926  1.00000000
```

We have some fairly large correlations between some of our predictor variables! When we run the regression model and look at the VIF values, we see that the collinearity is causing a lot of uncertainty about the coefficients. First, run the regression…

```
regression.3 <- lm( day ~ baby.sleep + dan.sleep + dan.grump, parenthood )
```

and second, look at the VIFs…

```
vif( regression.3 )
```

```
## baby.sleep  dan.sleep  dan.grump
##   1.651064   6.102337   5.437903
```

Yep, that's some mighty fine collinearity you've got there.

This page titled 15.11: Model Checking is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **15.9: Model Checking** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

15.12: Model Selection

One fairly major problem that remains is the problem of "model selection". That is, if we have a data set that contains several variables, which ones should we include as predictors, and which ones should we not include? In other words, we have a problem of ***variable selection***. In general, model selection is a complex business, but it's made somewhat simpler if we restrict ourselves to the problem of choosing a subset of the variables that ought to be included in the model. Nevertheless, I'm not going to try covering even this reduced topic in a lot of detail. Instead, I'll talk about two broad principles that you need to think about; and then discuss one concrete tool that R provides to help you select a subset of variables to include in your model. Firstly, the two principles:

- It's nice to have an actual substantive basis for your choices. That is, in a lot of situations you the researcher have good reasons to pick out a smallish number of possible regression models that are of theoretical interest; these models will have a sensible interpretation in the context of your field. Never discount the importance of this. Statistics serves the scientific process, not the other way around.
- To the extent that your choices rely on statistical inference, there is a trade off between simplicity and goodness of fit. As you add more predictors to the model, you make it more complex; each predictor adds a new free parameter (i.e., a new regression coefficient), and each new parameter increases the model's capacity to "absorb" random variations. So the goodness of fit (e.g., R^2) continues to rise as you add more predictors no matter what. If you want your model to be able to generalise well to new observations, you need to avoid throwing in too many variables.

This latter principle is often referred to as ***Ockham's razor***, and is often summarised in terms of the following pithy saying: *do not multiply entities beyond necessity*. In this context, it means: don't chuck in a bunch of largely irrelevant predictors just to boost your R^2. Hm. Yeah, the original was better.

In any case, what we need is an actual mathematical criterion that will implement the qualitative principle behind Ockham's razor in the context of selecting a regression model. As it turns out there are several possibilities. The one that I'll talk about is the ***Akaike information criterion*** (AIC; Akaike 1974) simply because it's the default one used in the R function `step()`. In the context of a linear regression model (and ignoring terms that don't depend on the model in any way!), the AIC for a model that has K predictor variables plus an intercept is:[227]

$$\text{AIC} = \frac{\text{SS}_{res}^2}{\hat{\sigma}} + 2K$$

The smaller the AIC value, the better the model performance is. If we ignore the low level details, it's fairly obvious what the AIC does: on the left we have a term that increases as the model predictions get worse; on the right we have a term that increases as the model complexity increases. The best model is the one that fits the data well (low residuals; left hand side) using as few predictors as possible (low K; right hand side). In short, this is a simple implementation of Ockham's razor.

15.12.1 Backward elimination

Okay, let's have a look at the `step()` function at work. In this example I'll keep it simple and use only the basic ***backward elimination*** approach. That is, start with the complete regression model, including all possible predictors. Then, at each "step" we try all possible ways of removing one of the variables, and whichever of these is best (in terms of lowest AIC value) is accepted. This becomes our new regression model; and we then try all possible deletions from the new model, again choosing the option with lowest AIC. This process continues until we end up with a model that has a lower AIC value than any of the other possible models that you could produce by deleting one of its predictors. Let's see this in action. First, I need to define the model from which the process starts.

```
full.model <- lm( formula = dan.grump ~ dan.sleep + baby.sleep + day,
                  data = parenthood
 )
```

That's nothing terribly new: yet another regression. Booooring. Still, we do need to do it: the `object` argument to the `step()` function will be this regression model. With this in mind, I would call the `step()` function using the following

command:

```
step( object = full.model,     # start at the full model
      direction = "backward"   # allow it remove predictors but not add them
)
```

```
## Start:  AIC=299.08
## dan.grump ~ dan.sleep + baby.sleep + day
##
##              Df Sum of Sq    RSS    AIC
## - baby.sleep  1       0.1 1837.2 297.08
## - day         1       1.6 1838.7 297.16
## <none>                    1837.1 299.08
## - dan.sleep   1    4909.0 6746.1 427.15
##
## Step:  AIC=297.08
## dan.grump ~ dan.sleep + day
##
##             Df Sum of Sq    RSS    AIC
## - day        1       1.6 1838.7 295.17
## <none>                   1837.2 297.08
## - dan.sleep  1    8103.0 9940.1 463.92
##
## Step:  AIC=295.17
## dan.grump ~ dan.sleep
##
##             Df Sum of Sq    RSS    AIC
## <none>                   1838.7 295.17
## - dan.sleep  1    8159.9 9998.6 462.50
```

```
##
## Call:
## lm(formula = dan.grump ~ dan.sleep, data = parenthood)
##
## Coefficients:
## (Intercept)    dan.sleep
##     125.956       -8.937
```

although in practice I didn't need to specify `direction` because `"backward"` is the default. The output is somewhat lengthy, so I'll go through it slowly. Firstly, the output reports the AIC value for the current best model:

```
Start:  AIC=299.08
dan.grump ~ dan.sleep + baby.sleep + day
```

That's our starting point. Since small AIC values are good, we want to see if we can get a value smaller than 299.08 by deleting one of those three predictors. So what R does is try all three possibilities, calculate the AIC values for each one, and then print out a short table with the results:

```
             Df Sum of Sq    RSS    AIC
- baby.sleep  1       0.1 1837.2 297.08
- day         1       1.6 1838.7 297.16
<none>                    1837.1 299.08
- dan.sleep   1    4909.0 6746.1 427.15
```

To read this table, it helps to note that the text in the left hand column is telling you what *change* R made to the regression model. So the line that reads `<none>` is the actual model we started with, and you can see on the right hand side that this still corresponds to an AIC value of 299.08 (obviously). The other three rows in the table correspond to the other three models that it looked at: it tried removing the `baby.sleep` variable, which is indicated by `- baby.sleep`, and this produced an AIC value of 297.08. That was the best of the three moves, so it's at the top of the table. So, this move is accepted, and now we start again. There are two predictors left in the model, `dan.sleep` and `day`, so it tries deleting those:

```
Step:  AIC=297.08
dan.grump ~ dan.sleep + day

            Df Sum of Sq    RSS    AIC
- day        1       1.6 1838.7 295.17
<none>                   1837.2 297.08
- dan.sleep  1    8103.0 9940.1 463.92
```

Okay, so what we can see is that removing the `day` variable lowers the AIC value from 297.08 to 295.17. So R decides to keep that change too, and moves on:

```
Step:  AIC=295.17
dan.grump ~ dan.sleep

            Df Sum of Sq    RSS    AIC
<none>                   1838.7 295.17
- dan.sleep  1    8159.9 9998.6 462.50
```

This time around, there's no further deletions that can actually improve the AIC value. So the `step()` function stops, and prints out the result of the best regression model it could find:

```
Call:
lm(formula = dan.grump ~ dan.sleep, data = parenthood)

Coefficients:
(Intercept)    dan.sleep
    125.956       -8.937
```

which is (perhaps not all that surprisingly) the `regression.1` model that we started with at the beginning of the chapter.

15.12.2 Forward selection

As an alternative, you can also try ***forward selection***. This time around we start with the smallest possible model as our start point, and only consider the possible additions to the model. However, there's one complication: you also need to tell `step()` what the largest possible model you're willing to entertain is, using the `scope` argument. The simplest usage is like this:

```
null.model <- lm( dan.grump ~ 1, parenthood )   # intercept only.
step( object = null.model,     # start with null.model
      direction = "forward",   # only consider "addition" moves
      scope =  dan.grump ~ dan.sleep + baby.sleep + day  # largest model allowed
)
```

```
## Start:  AIC=462.5
## dan.grump ~ 1
##
##              Df Sum of Sq    RSS    AIC
## + dan.sleep   1    8159.9 1838.7 295.17
## + baby.sleep  1    3202.7 6795.9 425.89
## <none>                    9998.6 462.50
## + day         1      58.5 9940.1 463.92
##
## Step:  AIC=295.17
## dan.grump ~ dan.sleep
##
##              Df Sum of Sq    RSS    AIC
## <none>                    1838.7 295.17
## + day         1   1.55760 1837.2 297.08
## + baby.sleep  1   0.02858 1838.7 297.16
```

```
##
## Call:
## lm(formula = dan.grump ~ dan.sleep, data = parenthood)
##
## Coefficients:
## (Intercept)    dan.sleep
##     125.956       -8.937
```

If I do this, the output takes on a similar form, but now it only considers addition (`+`) moves rather than deletion (`-`) moves:

```
Start:  AIC=462.5
dan.grump ~ 1

             Df Sum of Sq    RSS    AIC
+ dan.sleep   1    8159.9 1838.7 295.17
+ baby.sleep  1    3202.7 6795.9 425.89
<none>                    9998.6 462.50
+ day         1      58.5 9940.1 463.92

Step:  AIC=295.17
dan.grump ~ dan.sleep

             Df Sum of Sq    RSS    AIC
<none>                    1838.7 295.17
+ day         1   1.55760 1837.2 297.08
+ baby.sleep  1   0.02858 1838.7 297.16

Call:
lm(formula = dan.grump ~ dan.sleep, data = parenthood)

Coefficients:
(Intercept)    dan.sleep
    125.956       -8.937
```

As you can see, it's found the same model. In general though, forward and backward selection don't always have to end up in the same place.

15.12.3 caveat

Automated variable selection methods are seductive things, especially when they're bundled up in (fairly) simple functions like `step()`. They provide an element of objectivity to your model selection, and that's kind of nice. Unfortunately, they're sometimes used as an excuse for thoughtlessness. No longer do you have to think carefully about which predictors to add to the model and what the theoretical basis for their inclusion might be… everything is solved by the magic of AIC. And if we start throwing around phrases like Ockham's razor, well, it sounds like everything is wrapped up in a nice neat little package that no-one can argue with.

Or, perhaps not. Firstly, there's very little agreement on what counts as an appropriate model selection criterion. When I was taught backward elimination as an undergraduate, we used F-tests to do it, because that was the default method used by the software. The default in the `step()` function is AIC, and since this is an introductory text that's the only method I've described, but the AIC is hardly the Word of the Gods of Statistics. It's an approximation, derived under certain assumptions, and it's guaranteed to work only for large samples when those assumptions are met. Alter those assumptions and you get a different criterion, like the BIC for instance. Take a different approach again and you get the NML criterion. Decide that you're a Bayesian and you get model selection based on posterior odds ratios. Then there are a bunch of regression specific tools that I haven't mentioned. And so on. All of these different methods have strengths and weaknesses, and some are easier to calculate than others (AIC is probably the easiest of the lot, which might account for its popularity). Almost all of them produce the same answers when the answer is "obvious" but there's a fair amount of disagreement when the model selection problem becomes hard.

What does this mean in practice? Well, you *could* go and spend several years teaching yourself the theory of model selection, learning all the ins and outs of it; so that you could finally decide on what you personally think the right thing to do is. Speaking as someone who actually did that, I wouldn't recommend it: you'll probably come out the other side even more confused than when you started. A better strategy is to show a bit of common sense… if you're staring at the results of a `step()` procedure, and the

model that makes sense is close to having the smallest AIC, but is narrowly defeated by a model that doesn't make any sense… trust your instincts. Statistical model selection is an inexact tool, and as I said at the beginning, *interpretability matters*.

15.12.4 Comparing two regression models

An alternative to using automated model selection procedures is for the researcher to explicitly select two or more regression models to compare to each other. You can do this in a few different ways, depending on what research question you're trying to answer. Suppose we want to know whether or not the amount of sleep that my son got has any relationship to my grumpiness, over and above what we might expect from the amount of sleep that I got. We also want to make sure that the day on which we took the measurement has no influence on the relationship. That is, we're interested in the relationship between `baby.sleep` and `dan.grump`, and from that perspective `dan.sleep` and `day` are nuisance variable or ***covariates*** that we want to control for. In this situation, what we would like to know is whether `dan.grump ~ dan.sleep + day + baby.sleep` (which I'll call Model 1, or `M1`) is a better regression model for these data than `dan.grump ~ dan.sleep + day` (which I'll call Model 0, or `M0`). There are two different ways we can compare these two models, one based on a model selection criterion like AIC, and the other based on an explicit hypothesis test. I'll show you the AIC based approach first because it's simpler, and follows naturally from the `step()` function that we saw in the last section. The first thing I need to do is actually run the regressions:

```
M0 <- lm( dan.grump ~ dan.sleep + day, parenthood )
M1 <- lm( dan.grump ~ dan.sleep + day + baby.sleep, parenthood )
```

Now that I have my regression models, I could use the `summary()` function to run various hypothesis tests and other useful statistics, just as we have discussed throughout this chapter. However, since the current focus on model comparison, I'll skip this step and go straight to the AIC calculations. Conveniently, the `AIC()` function in R lets you input several regression models, and it will spit out the AIC values for each of them:[228]

```
AIC( M0, M1 )
```

```
##    df      AIC
## M0  4 582.8681
## M1  5 584.8646
```

Since Model 0 has the smaller AIC value, it is judged to be the better model for these data.

A somewhat different approach to the problem comes out of the hypothesis testing framework. Suppose you have two regression models, where one of them (Model 0) contains a *subset* of the predictors from the other one (Model 1). That is, Model 1 contains all of the predictors included in Model 0, plus one or more additional predictors. When this happens we say that Model 0 is ***nested*** within Model 1, or possibly that Model 0 is a ***submodel*** of Model 1. Regardless of the terminology what this means is that we can think of Model 0 as a null hypothesis and Model 1 as an alternative hypothesis. And in fact we can construct an F test for this in a fairly straightforward fashion. We can fit both models to the data and obtain a residual sum of squares for both models. I'll denote these as $SS_{res}^{(0)}$ and $SS_{res}^{(1)}$ respectively. The superscripting here just indicates which model we're talking about. Then our F statistic is

$$F = \frac{\left(\mathrm{SS}_{res}^{(0)} - \mathrm{SS}_{res}^{(1)}\right)/k}{\left(\mathrm{SS}_{res}^{(1)}\right)/(N-p-1)}$$

where N is the number of observations, p is the number of predictors in the full model (not including the intercept), and k is the difference in the number of parameters between the two models.[229] The degrees of freedom here are k and N−p−1. Note that it's often more convenient to think about the difference between those two SS values as a sum of squares in its own right. That is:

$$\mathrm{SS}_{\Delta} = \mathrm{SS}_{res}^{(0)} - \mathrm{SS}_{res}^{(1)}$$

The reason why this his helpful is that we can express SSΔ a measure of the extent to which the two models make different predictions about the the outcome variable. Specifically:

$$SS_\Delta = \sum_i (\hat{y}_i^{(1)} - \hat{y}_i^{(0)})^2$$

where $\hat{y}_i^{(0)}$ is the fitted value for y_i according to model M0 and $\hat{y}_i^{(1)}$ is the is the fitted value for yi according to model M_1.

Okay, so that's the hypothesis test that we use to compare two regression models to one another. Now, how do we do it in R? The answer is to use the `anova()` function. All we have to do is input the two models that we want to compare (null model first):

```
anova( M0, M1 )
```

```
## Analysis of Variance Table
##
## Model 1: dan.grump ~ dan.sleep + day
## Model 2: dan.grump ~ dan.sleep + day + baby.sleep
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     97 1837.2
## 2     96 1837.1  1  0.063688 0.0033 0.9541
```

Note that, just like we saw with the output from the `step()` function, R has used the acronym `RSS` to refer to the residual sum of squares from each model. That is, RSS in this output corresponds to SS_{res} in the formula above. Since we have p>.05 we retain the null hypothesis (`M0`). This approach to regression, in which we add all of our covariates into a null model, and then *add* the variables of interest into an alternative model, and then compare the two models in hypothesis testing framework, is often referred to as ***hierarchical regression***.

This page titled 15.12: Model Selection is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **15.10: Model Selection** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

15.13: Summary

- Basic ideas in linear regression and how regression models are estimated (Sections 15.1 and 15.2).
- Multiple linear regression (Section 15.3).
- Measuring the overall performance of a regression model using R^2 (Section 15.4)
- Hypothesis tests for regression models (Section 15.5)
- Calculating confidence intervals for regression coefficients, and standardised coefficients (Section 15.7)
- The assumptions of regression (Section 15.8) and how to check them (Section 15.9)
- Selecting a regression model (Section 15.10)

References

Fox, J., and S. Weisberg. 2011. *An R Companion to Applied Regression*. 2nd ed. Los Angeles: Sage.

Cook, R. D., and S. Weisberg. 1983. "Diagnostics for Heteroscedasticity in Regression." *Biometrika* 70: 1–10.

Long, J.S., and L.H. Ervin. 2000. "Using Heteroscedasticity Consistent Standard Errors in Thee Linear Regression Model." *The American Statistician* 54: 217–24.

Akaike, H. 1974. "A New Look at the Statistical Model Identification." *IEEE Transactions on Automatic Control* 19: 716–23.

214. The ϵ symbol is the Greek letter epsilon. It's traditional to use ϵ_i or e_i to denote a residual.
215. Or at least, I'm assuming that it doesn't help most people. But on the off chance that someone reading this is a proper kung fu master of linear algebra (and to be fair, I always have a few of these people in my intro stats class), it *will* help *you* to know that the solution to the estimation problem turns out to be $\hat{b} = (X^T X)^{-1} X^T y$, where $\hat{b}$ is a vector containing the estimated regression coefficients, X is the "design matrix" that contains the predictor variables (plus an additional column containing all ones; strictly X is a matrix of the regressors, but I haven't discussed the distinction yet), and y is a vector containing the outcome variable. For everyone else, this isn't exactly helpful, and can be downright scary. However, since quite a few things in linear regression can be written in linear algebra terms, you'll see a bunch of footnotes like this one in this chapter. If you can follow the maths in them, great. If not, ignore it.
216. And by "sometimes" I mean "almost never". In practice everyone just calls it "R-squared".
217. Note that, although R has done multiple tests here, it hasn't done a Bonferroni correction or anything. These are standard one-sample t-tests with a two-sided alternative. If you want to make corrections for multiple tests, you need to do that yourself.
218. You can change the kind of correction it applies by specifying the `p.adjust.method` argument.
219. Strictly, you standardise all the *regressors*: that is, every "thing" that has a regression coefficient associated with it in the model. For the regression models that I've talked about so far, each predictor variable maps onto exactly one regressor, and vice versa. However, that's not actually true in general: we'll see some examples of this in Chapter 16. But for now, we don't need to care too much about this distinction.
220. Or have no hope, as the case may be.
221. Again, for the linear algebra fanatics: the "hat matrix" is defined to be that matrix H that converts the vector of observed values y into a vector of fitted values $\hat{y}$, such that $\hat{y}$=Hy. The name comes from the fact that this is the matrix that "puts a hat on y". The hat *value* of the i-th observation is the i-th diagonal element of this matrix (so technically I should be writing it as hii rather than hi). Oh, and in case you care, here's how it's calculated: $H = X(X^T X)^{-1} X^T$. Pretty, isn't it?
222. Though special mention should be made of the `influenceIndexPlot()` and `influencePlot()` functions in the `car` package. These produce somewhat more detailed pictures than the default plots that I've shown here. There's also an `outlierTest()` function that tests to see if any of the Studentised residuals are significantly larger than would be expected by chance.
223. An alternative is to run a "robust regression"; I'll discuss robust regression in a later version of this book.
224. And, if you take the time to check the `residualPlots()` for `regression.1`, it's pretty clear that this isn't some wacky distortion being caused by the fact that `baby.sleep` is a useless predictor variable. It's an actual nonlinearity in the relationship between `dan.sleep` and `dan.grump`.
225. Note that the underlying mechanics of the test aren't the same as the ones I've described for regressions; the goodness of fit is assessed using what's known as a score-test not an F-test, and the test statistic is (approximately) $\chi 2$ distributed if there's no

relationship

226. Again, a footnote that should be read only by the two readers of this book that love linear algebra (mmmm… I love the smell of matrix computations in the morning; smells like… nerd). In these estimators, the covariance matrix for b is given by $(X^T X)^{-1} X^T \sum X (X^T X)^{-1}$. See, it's a "sandwich"? Assuming you think that $(X^T X)^{-1}$ ="bread" and XTΣX="filling", that is. Which of course everyone does, right? In any case, the usual estimator is what you get when you set $\sum = \hat{\sigma}^2 I$. The corrected version that I learned originally uses $diag(\epsilon_i^2)$ (White 1980). However, the version that Fox and Weisberg (2011)
227. Note, however, that the `step()` function computes the full version of AIC, including the irrelevant constants that I've dropped here. As a consequence this equation won't correctly describe the AIC values that you see in the outputs here. However, if you calculate the AIC values using my formula for two different regression models and take the difference between them, this will be the same as the differences between AIC values that `step()` reports. In practice, this is all you care about: the actual value of an AIC statistic isn't very informative, but the differences between two AIC values *are* useful, since these provide a measure of the extent to which one model outperforms another.
228. While I'm on this topic I should point out that there is also a function called `BIC()` which computes the Bayesian information criterion (BIC) for the models. So you could type `BIC(M0,M1)` and get a very similar output. In fact, while I'm not particularly impressed with either AIC or BIC as model selection methods, if you do find yourself using one of these two, the empirical evidence suggests that BIC is the better criterion of the two. In most simulation studies that I've seen, BIC does a much better job of selecting the correct model.
229. It's worth noting in passing that this same F statistic can be used to test a much broader range of hypotheses than those that I'm mentioning here. Very briefly: notice that the nested model M0 corresponds to the full model M1 when we constrain some of the regression coefficients to zero. It is sometimes useful to construct submodels by placing other kinds of constraints on the regression coefficients. For instance, maybe two different coefficients might have to sum to zero, or something like that. You can construct hypothesis tests for those kind of constraints too, but it is somewhat more complicated and the sampling distribution for F can end up being something known as the non-central F distribution, which is waaaaay beyond the scope of this book! All I want to do is alert you to this possibility.

This page titled 15.13: Summary is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **15.11: Summary** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

CHAPTER OVERVIEW

16: Research Design

A research design is the set of methods and procedures used in collecting and analyzing measures of the variables specified in the research problem research. The design of a study defines the study type (descriptive, correlational, semi-experimental, experimental, review, meta-analytic) and sub-type (e.g., descriptive-longitudinal case study), research problem, hypotheses, independent and dependent variables, experimental design, and, if applicable, data collection methods and a statistical analysis plan. Research design is the framework that has been created to find answers to research questions.

16.1: Scientific Method
16.2: Measurement
16.3: Data Collection
16.4: Sampling Bias
16.5: Experimental Designs
16.6: Causation
16.7: Statistical Literacy
16.E: Research Design (Exercises)

Flowchart of four phases (enrollment, intervention allocation, follow-up, and data analysis) of a parallel randomized trial of two groups. Image use with permission (CC BYT-SA 3.0; PrevMedFellow).

Contributors and Attributions

- Online Statistics Education: A Multimedia Course of Study (http://onlinestatbook.com/). Project Leader: David M. Lane, Rice University.
- Wikipedia

This page titled 16: Research Design is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

16.1: Scientific Method

Learning Objectives

- Brief discussion of the most important principles of the scientific method

This section contains a brief discussion of the most important principles of the scientific method. A thorough treatment of the philosophy of science is beyond the scope of this work.

One of the hallmarks of the scientific method is that it depends on empirical data. To be a proper scientific investigation the data must be collected systematically. However, scientific investigation does not necessarily require experimentation in the sense of manipulating variables and observing the results. Observational studies in the fields of astronomy, developmental psychology, and ethology are common and provide valuable scientific information.

Theories and explanations are very important in science. Theories in science can never be proved since one can never be 100% certain that a new empirical finding inconsistent with the theory will never be found.

Scientific theories must be potentially disconfirmable. If a theory can accommodate all possible results then it is not a scientific theory. Therefore, a scientific theory should lead to testable hypotheses. If a hypothesis is disconfirmed, then the theory from which the hypothesis was deduced is incorrect. For example, the secondary reinforcement theory of attachment states that an infant becomes attached to its parent by means of a pairing of the parent with a primary reinforcer (food). It is through this "secondary reinforcement" that the child-parent bond forms. The secondary reinforcement theory has been disconfirmed by numerous experiments. Perhaps the most notable is one in which infant monkeys were fed by a surrogate wire mother while a surrogate cloth mother was available. The infant monkeys formed no attachment to the wire monkeys and frequently clung to the cloth surrogate mothers.

History of Attachment Theory

If a hypothesis derived from a theory is confirmed then the theory has survived a test and it becomes more useful and better thought of by the researchers in the field. A theory is not confirmed when correct hypotheses are derived from it.

A key difference between scientific explanations and faith-based explanations is simply that faith-based explanations are based on faith and do not need to be testable. This does not mean that an explanation that cannot be tested is incorrect in some cosmic sense. It just means that it is not a scientific explanation.

The method of investigation in which a hypothesis is developed from a theory and then confirmed or disconfirmed involves deductive reasoning. However, deductive reasoning does not explain where the theory came from in the first place. In general, a theory is developed by a scientist who is aware of many empirical findings on a topic of interest. Then, through a generally poorly understood process called "induction" the scientist develops a way to explain all or most of the findings within a relatively simple framework or theory.

An important attribute of a good scientific theory is that it is parsimonious. That is, that it is simple in the sense that it uses relatively few constructs to explain many empirical findings. A theory that it so complex that it has as many assumptions as it has predictions is not very valuable.

Although strictly speaking, disconfirming an hypothesis deduced from a theory disconfirms the theory, it rarely leads to the abandonment of the theory. Instead, the theory will probably be modified to accommodate the inconsistent finding. If the theory has to be modified over and over to accommodate new findings, the theory generally becomes less and less parsimonious. This can lead to discontent with the theory and the search for a new theory. If a new theory is developed that can explain the same facts in a more parsimonious way, then the new theory will eventually supercede the old theory.

This page titled 16.1: Scientific Method is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **6.1: Scientific Method** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

16.2: Measurement

Learning Objectives

- Describe reliability in terms of true scores and error
- Define the standard error of measurement and state why it is valuable
- Distinguish between reliability and validity
- State the how reliability determines the upper limit to validity

The collection of data involves measurement. Measurement of some characteristics such as height and weight are relatively straightforward. The measurement of psychological attributes such as self esteem can be complex. A good measurement scale should be both reliable and valid. These concepts will be discussed in turn.

Reliability

The notion of reliability revolves around whether you would get at least approximately the same result if you measure something twice with the same measurement instrument. A common way to define reliability is the correlation between parallel forms of a test. Letting "test" represent a parallel form of the test, the symbol $r_{test,test}$ is used to denote the reliability of the test.

True Scores and Error

Assume you wish to measure a person's mean response time to the onset of a stimulus. For simplicity, assume that there is no learning over tests which, of course, is not really true. The person is given $1,000$ trials on the task and you obtain the response time on each trial.

The mean response time over the $1,000$ trials can be thought of as the person's "true" score, or at least a very good approximation of it. Theoretically, the true score is the mean that would be approached as the number of trials increases indefinitely.

An individual response time can be thought of as being composed of two parts: the true score and the error of measurement. Thus if the person's true score were 345 and their response on one of the trials were 358, then the error of measurement would be 13. Similarly, if the response time were 340, the error of measurement would be -5.

Now consider the more realistic example of a class of students taking a 100-point true/false exam. Let's assume that each student knows the answer to some of the questions and has no idea about the other questions. For the sake of simplicity, we are assuming there is no partial knowledge of any of the answers and for a given question a student either knows the answer or guesses. Finally, assume the test is scored such that a student receives one point for a correct answer and loses a point for an incorrect answer. In this example, a student's true score is the number of questions they know the answer to and their error score is their score on the questions they guessed on. For example, assume a student knew 90 of the answers and guessed correctly on 7 of the remaining 10 (and therefore incorrectly on 3). Their true score would be 90 since that is the number of answers they knew. Their error score would be $7-3=4$ and therefore their actual test score would be $90+4$.

Every test score can be thought of as the sum of two independent components, the true score and the error score. This can be written as:

$$y_{test} = y_{true} + y_{error} \tag{16.2.1}$$

The following expression follows directly from the Variance Sum Law:

$$\sigma^2_{Test} = \sigma^2_{True} + \sigma^2_{Error} \tag{16.2.2}$$

Reliability in Terms of True Scores and Error

It can be shown that the reliability of a test, $r_{test,test}$, is the ratio of true-score variance to test-score variance. This can be written as:

$$r_{test,test} = \frac{\sigma^2_{True}}{\sigma^2_{Test}} = \frac{\sigma^2_{True}}{\sigma^2_{True} + \sigma^2_{Error}} \tag{16.2.3}$$

PDF of derivation

It is important to understand the implications of the role the variance of true scores plays in the definition of reliability: If a test were given in two populations for which the variance of the true scores differed, the reliability of the test would be higher in the population with the higher true-score variance. Therefore, reliability is not a property of a test *per se* but the reliability of a test in a given population.

Assessing Error of Measurement

The reliability of a test does not show directly how close the test scores are to the true scores. That is, it does not reveal how much a person's test score would vary across parallel forms of test. By definition, the mean over a large number of parallel tests would be the true score. The standard deviation of a person's test scores would indicate how much the test scores vary from the true score. This standard deviation is called the standard error of measurement. In practice, it is not practical to give a test over and over to the same person and/or assume that there are no practice effects. Instead, the following formula is used to estimate the standard error of measurement.

$$S_{measurement} = S_{test}\sqrt{1 - r_{test,test}} \tag{16.2.4}$$

where $S_{measurement}$is the standard error of measurement, S_{test} is the standard deviation of the test scores, and $r_{test,test}$ is the reliability of the test. Taking the extremes, if the reliability is 0 then the standard error of measurement is equal to the standard deviation of the test; if the reliability is perfect (1.0) then the standard error of measurement is 0.

Increasing Reliability

It is important to make measures as reliable as is practically possible. Suppose an investigator is studying the relationship between spatial ability and a set of other variables. The higher the reliability of the test of spatial ability, the higher the correlations will be. Similarly, if an experimenter seeks to determine whether a particular exercise regiment decreases blood pressure, the higher the reliability of the measure of blood pressure, the more sensitive the experiment. More precisely, the higher the reliability the higher the power of the experiment. Power is covered in detail here. Finally, if a test is being used to select students for college admission or employees for jobs, the higher the reliability of the test the stronger will be the relationship to the criterion.

Two basic ways of increasing reliability are

1. to improve the quality of the items and
2. to increase the number of items.

Items that are either too easy so that almost everyone gets them correct or too difficult so that almost no one gets them correct are not good items: they provide very little information. In most contexts, items which about half the people get correct are the best (other things being equal).

Items that do not correlate with other items can usually be improved. Sometimes the item is confusing or ambiguous.

Increasing the number of items increases reliability in the manner shown by the following formula:

$$r_{new,new} = \frac{kr_{test,test}}{1 + (k-1)r_{test,test}} \tag{16.2.5}$$

where k is the factor by which the test length is increased, $r_{new,new}$ is the reliability of the new longer test, and $r_{test,test}$ is the current reliability. For example, if a test with 50 items has a reliability of 0.70 then the reliability of a test that is 1.5 times longer (75 items) would be calculated as follows:

$$r_{new,new} = \frac{(1.5)(0.70)}{1 + (1.5-1)(0.70)} \tag{16.2.6}$$

which equals 0.78. Thus increasing the number of items from 50 to 75 would increase the reliability from 0.70 to 0.78.

It is important to note that this formula assumes the new items have the same characteristics as the old items. Obviously adding poor items would not increase the reliability as expected and might even decrease the reliability.

More Information on Reliability from William Trochim's Knowledge Source

Validity

The validity of a test refers to whether the test measures what it is supposed to measure. The three most common types of validity are face validity, empirical validity, and construct validity. We consider these types of validity below.

- **Face Validity:** A test's face validity refers to whether the test appears to measure what it is supposed to measure. That is, does the test "on its face" appear to measure what it is supposed to be measuring. An Asian history test consisting of a series of questions about Asian history would have high face validity. If the test included primarily questions about American history then it would have little or no face validity as a test of Asian history.
- **Predictive Validity:** Predictive validity (sometimes called empirical validity) refers to a test's ability to predict the relevant behavior. For example, the main way in which SAT tests are validated is by their ability to predict college grades. Thus, to the extent these tests are successful at predicting college grades they are said to possess predictive validity.
- **Construct Validity:** Construct validity is more difficult to define. In general, a test has construct validity if its pattern of correlations with other measures is in line with the construct it is purporting to measure. Construct validity can be established by showing a test has both convergent and divergent validity. A test has convergent validity if it correlates with other tests that are also measures of the construct in question. Divergent validity is established by showing the test does not correlate highly with tests of other constructs. Of course, some constructs may overlap so the establishment of convergent and divergent validity can be complex.

To take an example, suppose one wished to establish the construct validity of a new test of spatial ability. Convergent and divergent validity could be established by showing the test correlates relatively highly with other measures of spatial ability but less highly with tests of verbal ability or social intelligence.

Reliability and Predictive Validity

The reliability of a test limits the size of the correlation between the test and other measures. In general, the correlation of a test with another measure will be lower than the test's reliability. After all, how could a test correlate with something else as high as it correlates with a parallel form of itself? Theoretically it is possible for a test to correlate as high as the square root of the reliability with another measure. For example, if a test has a reliability of 0.81 then it could correlate as high as 0.90 with another measure. This could happen if the other measure were a perfectly reliable test of the same construct as the test in question. In practice, this is very unlikely.

A correlation above the upper limit set by reliabilities can act as a red flag. For example, Vul, Harris, Winkielman, and Paschler (2009) found that in many studies the correlations between various fMRI activation patterns and personality measures were higher than their reliabilities would allow. A careful examination of these studies revealed serious flaws in the way the data were analyzed.

Vul, E., Harris, C., Winkielman, P., & Paschler, H. (2009) Puzzlingly High Correlations in fMRI Studies of Emotion, Personality, and Social Cognition. *Perspectives on Psychological Science*, *4*, 274-290.

This page titled 16.2: Measurement is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **6.2: Measurement** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

16.3: Data Collection

Learning Objectives

- Describe how a variable such as height should be recorded
- Choose a good response scale for a questionnaire

Most statistical analyses require that your data be in numerical rather than verbal form (you can't punch letters into your calculator). Therefore, data collected in verbal form must be coded so that it is represented by numbers. To illustrate, consider the data in Table **16.3.1**.

Table **16.3.1**: *Example Data*

Student Name	Hair Color	Gender	Major	Height	Computer Experience
Norma	Brown	Female	Psychology	5'4"	Lots
Amber	Blonde	Female	Social Science	5'7"	Very little
Paul	Blonde	Male	History	6'1"	Moderate
Christopher	Black	Male	Biology	5'10"	Lots
Sonya	Brown	Female	Psychology	5'4"	Little

Can you conduct statistical analyses on the above data or must you re-code it in some way? For example, how would you go about computing the average height of the 5 students. You cannot enter students' heights in their current form into a statistical program -- the computer would probably give you an error message because it does not understand notation such as $5'4''$. One solution is to change all the numbers to inches. So, $5'4''$ becomes $(5 \times 12) + 4 = 64$, and $6'1''$ becomes $(6 \times 12) + 1 = 73$, and so forth. In this way, you are converting height in feet and inches to simply height in inches. From there, it is very easy to ask a statistical program to calculate the mean height in inches for the 5 students.

You may ask, "Why not simply ask subjects to write their height in inches in the first place?" Well, the number one rule of data collection is to ask for information in such a way as it will be most accurately reported. Most people know their height in feet and inches and cannot quickly and accurately convert it into inches "on the fly." So, in order to preserve data accuracy, it is best for researchers to make the necessary conversions.

Let's take another example. Suppose you wanted to calculate the mean amount of computer experience for the five students shown in Table **16.3.1**. One way would be to convert the verbal descriptions to numbers as shown in Table **16.3.2**. Thus, "Very Little" would be converted to "1" and "Little" would be converted to "2."

Table **16.3.2**: *Conversion of verbal descriptions to numbers.*

1	2	3	4	5
Very Little	Little	Moderate	Lots	Very Lots

Example 16.3.1: How much information should I record?

Say you are volunteering at a track meet at your college, and your job is to record each runner's time as they pass the finish line for each race. Their times are shown in large red numbers on a digital clock with eight digits to the right of the decimal point, and you are told to record the entire number in your tablet. Thinking eight decimal places is a bit excessive, you only record runners' times to one decimal place. The track meet begins, and runner number one finishes with a time of 22.93219780 seconds. You dutifully record her time in your tablet, but only to one decimal place, that is 22.9. Race number two finishes and you record 32.7 for the winning runner. The fastest time in Race number three is 25.6. Race number four winning time is 22.9, Race number five is…. But wait! You suddenly realize your mistake; you now have a tie between runner one and runner four

for the title of Fastest Overall Runner! You should have recorded more information from the digital clock -- that information is now lost, and you cannot go back in time and record running times to more decimal places.

The point is that you should think very carefully about the scales and specificity of information needed in your research before you begin collecting data. If you believe you might need additional information later but are not sure, measure it; you can always decide to not use some of the data, or "collapse" your data down to lower scales if you wish, but you cannot expand your data set to include more information after the fact. In this example, you probably would not need to record eight digits to the right of the decimal point. But recording only one decimal digit is clearly too few.

Example 16.3.2

Pretend for a moment that you are teaching five children in middle school (yikes!), and you are trying to convince them that they must study more in order to earn better grades. To prove your point, you decide to collect actual data from their recent math exams, and, toward this end, you develop a questionnaire to measure their study time and subsequent grades. You might develop a questionnaire which looks like the following:

1. Please write your name: _____
2. Please indicate how much you studied for this math exam:
   a lot……moderate……..…….little
3. Please circle the grade you received on the math exam: $A\ B\ C\ D\ F$

Given the above questionnaire, your obtained data might look like the following:

Name	Amount Studied	Grade
John	Little	C
Sally	Moderate	B
Alexander	Lots	A
Linda	Moderate	A
Thomas	Little	B

Eyeballing the data, it seems as if the children who studied more received better grades, but it's difficult to tell. "Little," "lots," and "B," are imprecise, qualitative terms. You could get more precise information by asking specifically how many hours they studied and their exact score on the exam. The data then might look as follows:

	Name			Hours studied			% Correct	
	John		5		71			
	Sally		9		83			
	Alexander		13		97			
	Linda		12		91			
	Thomas		7		85			

Of course, this assumes the students would know how many hours they studied. Rather than trust the students' memories, you might ask them to keep a log of their study time as they study.

Contributors and Attributions

- Online Statistics Education: A Multimedia Course of Study (http://onlinestatbook.com/). Project Leader: David M. Lane, Rice University.

- Heidi Zeimer

This page titled 16.3: Data Collection is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **6.3: Data Collection** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

16.4: Sampling Bias

Learning Objectives

- Recognize sampling bias
- Distinguish among self-selection bias, undercoverage bias, and survivorship bias

Descriptions of various types of sampling such as simple random sampling and stratified random sampling are covered in another section. This section discusses various types of sampling biases including self-selection bias and survivorship bias. Examples of other sampling biases that are not easily categorized will also be given.

It is important to keep in mind that sampling bias refers to the method of sampling, not the sample itself. There is no guarantee that random sampling will result in a sample representative of the population just as not every sample obtained using a biased sampling method will be greatly non-representative of the population.

Self-Selection Bias

Imagine that a university newspaper ran an ad asking for students to volunteer for a study in which intimate details of their sex lives would be discussed. Clearly the sample of students who would volunteer for such a study would not be representative of the students at the university. Similarly, an online survey about computer use is likely to attract people more interested in technology than is typical. In both of these examples, people who "self-select" themselves for the experiment are likely to differ in important ways from the population the experimenter wishes to draw conclusions about. Many of the admittedly "non-scientific" polls taken on television or web sites suffer greatly from self-selection bias.

A self-selection bias can result when the non-random component occurs after the potential subject has enlisted in the experiment. Considering again the hypothetical experiment in which subjects are to be asked intimate details of their sex lives, assume that the subjects did not know what the experiment was going to be about until they showed up.Many of the subjects would then likely leave the experiment resulting in a biased sample.

Undercoverage Bias

A common type of sampling bias is to sample too few observations from a segment of the population. A commonly-cited example of undercoverage is the poll taken by the Literary Digest in 1936 that indicated that Landon would win an election against Roosevelt by a large margin when, in fact, it was Roosevelt who won by a large margin. A common explanation is that poorer people were undercovered because they were less likely to have telephones and that this group was more likely to support Roosevelt.

A detailed analysis by Squire (1988) showed that it was not just an undercoverage bias that resulted in the faulty prediction of the election results. He concluded that, in addition to the undercoverage described above, there was a nonresponse bias (a form of self-selection bias) such that those favoring Landon were more likely to return their survey than were those favoring Roosevelt.

Survivorship Bias

Survivorship bias occurs when the observations recorded at the end of the investigation are a non-random set of those present at the beginning of the investigation. The gains in stock funds is an area in which survivorship bias often plays a role. The basic problem is that poorly-performing funds are often either eliminated or merged into other funds. Suppose one considers a sample of stock funds that exist in the present and then calculates the mean 10-year appreciation of those funds. Can these results be validly generalized to other stock funds of the same type? The problem is that the poorly-performing stock funds that are not still in existence (did not survive for 10 years) are not included and therefore there is a bias toward selecting better-performing funds. There is good evidence that this survivorship bias is substantial (Malkiel, 1995).

In World War II, the statistician Abraham Wald analyzed the distribution of hits from anti-aircraft fire on aircraft returning from missions. The idea was that this information would be useful for deciding where to place extra armor. A naive approach would be to put armor at locations that were frequently hit to reduce the damage there. However, this would ignore the survivorship bias occurring because only a subset of aircraft return. Wald's approach was the opposite: if there were few hits in a certain location on

returning planes, then hits in that location were likely to bring a plane down. Therefore, he recommended that locations without hits on the returning planes should be given extra armor. A detailed and mathematical description of Wald's work can be found in Mangel and Samaniego (**1984**.)

References

1. Malkiel, B. G. (1995) Returns from investing in equity mutual funds 1971 to 1991. *The Journal of Finance*, *50*, 549-572.
2. Mangel, M. & Samaniego, F. J. (1984) Abraham Wald's work on aircraft survivability. *Journal of the American Statistical Association*, *79*, 259-267.
3. Squire, P. (1988) Why the 1936 Literary Digest poll failed. *Public Opinion Quarterly*, *52*, 125-133.

This page titled 16.4: Sampling Bias is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **6.4: Sampling Bias** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

16.5: Experimental Designs

Learning Objectives

- Distinguish between between-subject and within-subject designs
- State the advantages of within-subject designs
- Define "multi-factor design" and "factorial design"
- Identify the levels of a variable in an experimental design
- Describe when counterbalancing is used

There are many ways an experiment can be designed. For example, subjects can all be tested under each of the treatment conditions or a different group of subjects can be used for each treatment. An experiment might have just one independent variable or it might have several. This section describes basic experimental designs and their advantages and disadvantages.

Between-Subjects Designs

In a between-subjects design, the various experimental treatments are given to different groups of subjects. For example, in the "Teacher Ratings" case study, subjects were randomly divided into two groups. Subjects were all told they were going to see a video of an instructor's lecture after which they would rate the quality of the lecture. The groups differed in that the subjects in one group were told that prior teaching evaluations indicated that the instructor was charismatic whereas subjects in the other group were told that the evaluations indicated the instructor was punitive. In this experiment, the independent variable is "Condition" and has two levels (charismatic teacher and punitive teacher). It is a between-subjects variable because different subjects were used for the two levels of the independent variable: subjects were in either the "charismatic teacher" or the "punitive teacher" condition. Thus the comparison of the charismatic-teacher condition with the punitive-teacher condition is a comparison between the subjects in one condition with the subjects in the other condition.

The two conditions were treated exactly the same except for the instructions they received. Therefore, it would appear that any difference between conditions should be attributed to the treatments themselves. However, this ignores the possibility of chance differences between the groups. That is, by chance, the raters in one condition might have, on average, been more lenient than the raters in the other condition. Randomly assigning subjects to treatments ensures that all differences between conditions are chance differences; it does not ensure there will be no differences. The key question, then, is how to distinguish real differences from chance differences. The field of inferential statistics answers just this question. The inferential statistics applicable to testing the difference between the means of the two conditions can be found here. Analyzing the data from this experiment reveals that the ratings in the charismatic-teacher condition were higher than those in the punitive-teacher condition. Using inferential statistics, it can be calculated that the probability of finding a difference as large or larger than the one obtained if the treatment had no effect is only 0.018. Therefore it seems likely that the treatment had an effect and it is not the case that all differences were chance differences.

Independent variables often have several levels. For example, in the "Smiles and Leniency" case study the independent variable is "type of smile" and there are four levels of this independent variable:

1. false smile
2. felt smile
3. miserable smile
4. a neutral control

Keep in mind that although there are four levels, there is only one independent variable. Designs with more than one independent variable are considered next.

Multi-Factor Between-Subject Designs

In the "Bias Against Associates of the Obese" experiment, the qualifications of potential job applicants were judged. Each applicant was accompanied by an associate. The experiment had two independent variables: the weight of the associate (obese or average) and the applicant's relationship to the associate (girl friend or acquaintance). This design can be described as an Associate's Weight

(2) x Associate's Relationship (2) factorial design. The numbers in parentheses represent the number of levels of the independent variable. The design was a factorial design because all four combinations of associate's weight and associate's relationship were included. The dependent variable was a rating of the applicant's qualifications (on a 9-point scale).

If two separate experiments had been conducted, one to test the effect of Associate's Weight and one to test the effect of Associate's Relationship then there would be no way to assess whether the effect of Associate's Weight depended on the Associate's Relationship. One might imagine that the Associate's Weight would have a larger effect if the associate were a girl friend rather than merely an acquaintance. A factorial design allows this question to be addressed. When the effect of one variable does differ depending on the level of the other variable then it is said that there is an interaction between the variables.

Factorial designs can have three or more independent variables. In order to be a between-subjects design there must be a separate group of subjects for each combination of the levels of the independent variables.

Within-Subjects Designs

A within-subjects design differs from a between-subjects design in that the same subjects perform at all levels of the independent variable. For example consider the "ADHD Treatment" case study. In this experiment, subjects diagnosed as having attention deficit disorder were each tested on a delay of gratification task after receiving methylphenidate (MPH). All subjects were tested four times, once after receiving one of the four doses. Since each subject was tested under *each* of the four levels of the independent variable "dose," the design is a within-subjects design and dose is a within-subjects variable. Within-subjects designs are sometimes called repeated-measures designs.

Counterbalancing

In a within-subject design it is important not to confound the order in which a task is performed with the experimental treatment. For example, consider the problem that would have occurred if, in the ADHD study, every subject had received the doses in the same order starting with the lowest and continuing to the highest. It is not unlikely that experience with the delay of gratification task would have an effect. If practice on this task leads to better performance, then it would appear that higher doses caused the better performance when, in fact, it was the practice that caused the better performance.

One way to address this problem is to counterbalance the order of presentations. In other words, subjects would be given the doses in different orders in such a way that each dose was given in each sequential position an equal number of times. An example of counterbalancing is shown in Table 16.5.1.

Table **16.5.1**: *Counterbalanced order for four subjects*

Subject	0 mg/kg	0.15 mg/kg	0.30 mg/kg	0.60 mg/kg
1	First	Second	Third	Fourth
2	Second	Third	Fourth	First
3	Third	Fourth	First	Second
4	Fourth	First	Second	Third

It should be kept in mind that counterbalancing is not a satisfactory solution if there are complex dependencies between which treatment precedes which and the dependent variable. In these cases, it is usually better to use a between-subjects design than a within-subjects design.

Advantage of Within-Subjects Designs

An advantage of within-subjects designs is that individual differences in subjects' overall levels of performance are controlled. This is important because subjects invariably will differ greatly from one another. In an experiment on problem solving, some subjects will be better than others regardless of the condition they are in. Similarly, in a study of blood pressure some subjects will have higher blood pressure than others regardless of the condition. Within-subjects designs control these individual differences by comparing the scores of a subject in one condition to the scores of the same subject in other conditions. In this sense each subject serves as his or her own control. This typically gives within-subjects designs considerably more power than between-subjects

designs. That is, this makes within-subjects designs more able to detect an effect of the independent variable than are between-subjects designs.

Within-subjects designs are often called "repeated-measures" designs since repeated measurements are taken for each subject. Similarly, a within-subject variable can be called a repeated-measures factor.

Complex Designs

Designs can contain combinations of between-subject and within-subject variables. For example, the "Weapons and Aggression" case study has one between-subject variable (gender) and two within-subject variables (the type of priming word and the type of word to be responded to).

This page titled 16.5: Experimental Designs is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **6.5: Experimental Designs** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

16.6: Causation

Learning Objectives

- Explain how experimentation allows causal inferences
- Explain the role of unmeasured variables
- Explain the "third-variable" problem
- Explain how causation can be inferred in non-experimental designs

The concept of causation is a complex one in the philosophy of science. Since a full coverage of this topic is well beyond the scope of this text, we focus on two specific topics:

1. the establishment of causation in experiments
2. the establishment of causation in non-experimental designs

Stanford's Encyclopedia of Philosophy: Causation Topics

Establishing Causation in Experiments

Consider a simple experiment in which subjects are sampled randomly from a population and then assigned randomly to either the experimental group or the control group. Assume the condition means on the dependent variable differed. Does this mean the treatment caused the difference?

To make this discussion more concrete, assume that the experimental group received a drug for insomnia, the control group received a placebo, and the dependent variable was the number of minutes the subject slept that night. An obvious obstacle to inferring causality is that there are many unmeasured variables that affect how many hours someone sleeps. Among them are how much stress the person is under, physiological and genetic factors, how much caffeine they consumed, how much sleep they got the night before, etc. Perhaps differences between the groups on these factors are responsible for the difference in the number of minutes slept.

At first blush it might seem that the random assignment eliminates differences in unmeasured variables. However, this is not the case. Random assignment ensures that differences on unmeasured variables are chance differences. It does not ensure that there are no differences. Perhaps, by chance, many subjects in the control group were under high stress and this stress made it more difficult to fall asleep. The fact that the greater stress in the control group was due to chance does not mean it could not be responsible for the difference between the control and the experimental groups. In other words, the observed difference in "minutes slept" could have been due to a chance difference between the control group and the experimental group rather than due to the drug's effect.

This problem seems intractable since, by definition, it is impossible to measure an "unmeasured variable" just as it is impossible to measure and control all variables that affect the dependent variable. However, although it is impossible to assess the effect of any single unmeasured variable, it is possible to assess the combined effects of all unmeasured variables. Since everyone in a given condition is treated the same in the experiment, differences in their scores on the dependent variable must be due to the unmeasured variables. Therefore, a measure of the differences among the subjects within a condition is a measure of the sum total of the effects of the unmeasured variables. The most common measure of differences is the variance. By using the within-condition variance to assess the effects of unmeasured variables, statistical methods determine the probability that these unmeasured variables could produce a difference between conditions as large or larger than the difference obtained in the experiment. If that probability is low, then it is inferred (that's why they call it inferential statistics) that the treatment had an effect and that the differences are not entirely due to chance. Of course, there is always some nonzero probability that the difference occurred by chance so total certainty is not a possibility.

Causation in Non-Experimental Designs

It is almost a cliché that correlation does not mean causation. The main fallacy in inferring causation from correlation is called the "third variable problem" and means that a third variable is responsible for the correlation between two other variables. An excellent example used by Li (1975) to illustrate this point is the positive correlation in Taiwan in the $1970's$ between the use of contraception

and the number of electric appliances in one's house. Of course, using contraception does not induce you to buy electrical appliances or vice versa. Instead, the third variable of education level affects both.

Does the possibility of a third-variable problem make it impossible to draw causal inferences without doing an experiment? One approach is to simply assume that you do not have a third-variable problem. This approach, although common, is not very satisfactory. However, be aware that the assumption of no third-variable problem may be hidden behind a complex causal model that contains sophisticated and elegant mathematics.

A better though, admittedly more difficult approach, is to find converging evidence. This was the approach taken to conclude that smoking causes cancer. The analysis included converging evidence from retrospective studies, prospective studies, lab studies with animals, and theoretical understandings of cancer causes.

A second problem is determining the direction of causality. A correlation between two variables does not indicate which variable is causing which. For example, Reinhart and Rogoff (**2010**) found a strong correlation between public debt and GDP growth. Although some have argued that public debt slows growth, most evidence supports the alternative that slow growth increases public debt.

Excellent Video on Causality Featuring Evidence that Smoking Causes Cancer(See Chapter 11)

1. Li, C. (1975) Path *analysis: A primer*. Boxwood Press, Pacific Grove. CA .
2. Reinhart, C. M. and Rogoff, K. S. (2010). Growth in a Time of Debt. Working Paper 15639, National Bureau of Economic Research, http://www.nber.org/papers/w15639

This page titled 16.6: Causation is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **6.6: Causation** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

16.7: Statistical Literacy

Learning Objectives

- Design a statistical study involving niacin, HDL and heart disease

Low HDL and Niacin

A low level of High-density lipoproteins (HDL) have long been known to be a risk factor for heart disease. Taking niacin has been shown to increase HDL levels and has been recommended for patients with low levels of HDL. The assumption of this recommendation is that niacin causes HDL to increase thus causing a lower risk for heart disease.

Example 16.7.1: What do you think?

What experimental design involving niacin would test whether the relationship between HDL and heart disease is causal?

Solution

You could randomly assign patients with low levels of HDL to a condition in which they received niacin or to one in which they did not. A finding that niacin increased HDL without decreasing heart disease would cast doubt on the causal relationship. This is exactly what was found in a study conducted by the NIH. See the description of the results here.

This page titled 16.7: Statistical Literacy is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **6.7: Statistical Literacy** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

16.E: Research Design (Exercises)

General Questions

Q1

To be a scientific theory, the theory must be potentially _____.

Q2

What is the difference between a faith-based explanation and a scientific explanation?

Q3

What does it mean for a theory to be parsimonious?

Q4

Define reliability in terms of parallel forms.

Q5

Define true score.

Q6

What is the reliability if the true score variance is **80** and the test score variance is **100**?

Q7

What statistic relates to how close a score on one test will be to a score on a parallel form?

Q8

What is the effect of test length on the reliability of a test?

Q9

Distinguish between predictive validity and construct validity.

Q10

What is the theoretical maximum correlation of a test with a criterion if the test has a reliability of **0.81**?

Q11

An experiment solicits subjects to participate in a highly stressful experiment. What type of sampling bias is likely to occur?

Q12

Give an example of survivorship bias not presented in this text.

Q13

Distinguish "between-subject" variables from "within-subjects" variables.

Q14

Of the variables "gender" and "trials," which is likely to be a between-subjects variable and which a within-subjects variable?

Q15

Define interaction.

Q16

What is counterbalancing used for?

Q17

How does randomization deal with the problem of pre-existing differences between groups?

Q18

Give an example of the "third variable problem" other than those in this text.

This page titled 16.E: Research Design (Exercises) is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **6.E: Research Design (Exercises)** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

CHAPTER OVERVIEW

17: Preparing Datasets and Other Pragmatic Matters

> *The garden of life never seems to confine itself to the plots philosophers have laid out for its convenience. Maybe a few more tractors would do the trick.*
>
> *–Roger Zelazny*[103]

This is a somewhat strange chapter, even by my standards. My goal in this chapter is to talk a bit more honestly about the realities of working with data than you'll see anywhere else in the book. The problem with real world data sets is that they are *messy*. Very often the data file that you start out with doesn't have the variables stored in the right format for the analysis you want to do. Sometimes might be a lot of missing values in your data set. Sometimes you only want to analyse a subset of the data. Et cetera. In other words, there's a lot of ***data manipulation*** that you need to do, just to get all your data set into the format that you need it. The purpose of this chapter is to provide a basic introduction to all these pragmatic topics. Although the chapter is motivated by the kinds of practical issues that arise when manipulating real data, I'll stick with the practice that I've adopted through most of the book and rely on very small, toy data sets that illustrate the underlying issue. Because this chapter is essentially a collection of "tricks" and doesn't tell a single coherent story, it may be useful to start with a list of topics:

- Section 7.1. Tabulating data.
- Section 7.2. Transforming or recoding a variable.
- Section 7.3. Some useful mathematical functions.
- Section 7.4. Extracting a subset of a vector.
- Section 7.5. Extracting a subset of a data frame.
- Section 7.6. Sorting, flipping or merging data sets.
- Section 7.7. Reshaping a data frame.
- Section 7.8. Manipulating text.
- Section 7.9. Opening data from different file types.
- Section 7.10. Coercing data from one type to another.
- Section 7.11. Other important data types.
- Section 7.12. Miscellaneous topics.

As you can see, the list of topics that the chapter covers is pretty broad, and there's a *lot* of content there. Even though this is one of the longest and hardest chapters in the book, I'm really only scratching the surface of several fairly different and important topics. My advice, as usual, is to read through the chapter once and try to follow as much of it as you can. Don't worry too much if you can't grasp it all at once, especially the later sections. The rest of the book is only lightly reliant on this chapter, so you can get away with just understanding the basics. However, what you'll probably find is that later on you'll need to flick back to this chapter in order to understand some of the concepts that I refer to here.

17.1: Tabulating and Cross-tabulating Data
17.2: Transforming and Recoding a Variable
17.3: A few More Mathematical Functions and Operations
17.4: Extracting a Subset of a Vector
17.5: Extracting a Subset of a Data Frame
17.6: Sorting, Flipping and Merging Data
17.7: Reshaping a Data Frame
17.8: Working with Text
17.9: Reading Unusual Data Files
17.10: Coercing Data from One Class to Another
17.11: Other Useful Data Structures

17.12: Miscellaneous Topics
17.13: Summary

This page titled 17: Preparing Datasets and Other Pragmatic Matters is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

17.1: Tabulating and Cross-tabulating Data

A very common task when analysing data is the construction of frequency tables, or cross-tabulation of one variable against another. There are several functions that you can use in R for that purpose. In this section I'll illustrate the use of three functions – `table()`, `xtabs()` and `tabulate()` – though there are other options (e.g., `ftable()`) available.

17.1.1 Creating tables from vectors

Let's start with a simple example. As the father of a small child, I naturally spend a lot of time watching TV shows like *In the Night Garden*. In the `nightgarden.Rdata` file, I've transcribed a short section of the dialogue. The file contains two variables, `speaker` and `utterance`, and when we take a look at the data, it becomes very clear what happened to my sanity.

```
library(lsr)
load("./rbook-master/data/nightgarden.Rdata" )
who()
```

```
##    -- Name --    -- Class --    -- Size --
##    speaker       character      10
##    utterance     character      10
```

```
print( speaker )
```

```
##  [1] "upsy-daisy"  "upsy-daisy"  "upsy-daisy"  "upsy-daisy"  "tombliboo"
##  [6] "tombliboo"   "makka-pakka" "makka-pakka" "makka-pakka" "makka-pakka"
```

```
print( utterance )
```

```
##  [1] "pip" "pip" "onk" "onk" "ee"  "oo"  "pip" "pip" "onk" "onk"
```

With these as my data, one task I might find myself needing to do is construct a frequency count of the number of words each character speaks during the show. The `table()` function provides a simple way do to this. The basic usage of the `table()` function is as follows:

```
table(speaker)
```

```
## speaker
## makka-pakka   tombliboo  upsy-daisy
##           4           2           4
```

The output here tells us on the first line that what we're looking at is a tabulation of the `speaker` variable. On the second line it lists all the different speakers that exist in the data, and on the third line it tells you how many times that speaker appears in the data. In other words, it's a frequency table[104] Notice that in the command above I didn't name the argument, since `table()` is another function that makes use of unnamed arguments. You just type in a list of the variables that you want R to tabulate, and it tabulates them. For instance, if I type in the name of two variables, what I get as the output is a cross-tabulation:

```
table(speaker, utterance)
```

```
##             utterance
## speaker       ee onk oo pip
##   makka-pakka  0   2  0   2
##   tombliboo    1   0  1   0
##   upsy-daisy   0   2  0   2
```

When interpreting this table, remember that these are counts: so the fact that the first row and second column corresponds to a value of 2 indicates that Makka-Pakka (row 1) says "onk" (column 2) twice in this data set. As you'd expect, you can produce three way or higher order cross tabulations just by adding more objects to the list of inputs. However, I won't discuss that in this section.

17.1.2 Creating tables from data frames

Most of the time your data are stored in a data frame, not kept as separate variables in the workspace. Let's create one:

```
itng <- data.frame( speaker, utterance )
itng
```

```
##        speaker utterance
## 1   upsy-daisy       pip
## 2   upsy-daisy       pip
## 3   upsy-daisy       onk
## 4   upsy-daisy       onk
## 5    tombliboo        ee
## 6    tombliboo        oo
## 7  makka-pakka       pip
## 8  makka-pakka       pip
## 9  makka-pakka       onk
## 10 makka-pakka       onk
```

There's a couple of options under these circumstances. Firstly, if you just want to cross-tabulate all of the variables in the data frame, then it's really easy:

```
table(itng)
```

```
##             utterance
## speaker       ee onk oo pip
##   makka-pakka  0   2  0   2
##   tombliboo    1   0  1   0
##   upsy-daisy   0   2  0   2
```

However, it's often the case that you want to select particular variables from the data frame to tabulate. This is where the `xtabs()` function is useful. In this function, you input a one sided `formula` in order to list all the variables you want to cross-tabulate, and the name of the `data` frame that stores the data:

```
xtabs( formula = ~ speaker + utterance, data = itng )
```

```
##             utterance
## speaker       ee onk oo pip
##   makka-pakka  0   2  0   2
##   tombliboo    1   0  1   0
##   upsy-daisy   0   2  0   2
```

Clearly, this is a totally unnecessary command in the context of the `itng` data frame, but in most situations when you're analysing real data this is actually extremely useful, since your data set will almost certainly contain lots of variables and you'll only want to tabulate a few of them at a time.

17.1.3 Converting a table of counts to a table of proportions

The tabulation commands discussed so far all construct a table of raw frequencies: that is, a count of the total number of cases that satisfy certain conditions. However, often you want your data to be organised in terms of proportions rather than counts. This is where the `prop.table()` function comes in handy. It has two arguments:

- `x` . The frequency table that you want to convert.
- `margin` . Which "dimension" do you want to calculate proportions for. By default, R assumes you want the proportion to be expressed as a fraction of all possible events. See examples for details.

To see how this works:

```
itng.table <- table(itng)  # create the table, and assign it to a variable
itng.table                 # display the table again, as a reminder
```

```
##             utterance
## speaker       ee onk oo pip
##   makka-pakka  0   2  0   2
##   tombliboo    1   0  1   0
##   upsy-daisy   0   2  0   2
```

```
prop.table( x = itng.table ) # express as proportion:
```

```
##             utterance
## speaker       ee onk  oo pip
##   makka-pakka 0.0 0.2 0.0 0.2
##   tombliboo   0.1 0.0 0.1 0.0
##   upsy-daisy  0.0 0.2 0.0 0.2
```

Notice that there were 10 observations in our original data set, so all that R has done here is divide all our raw frequencies by 10. That's a sensible default, but more often you actually want to calculate the proportions separately by row (`margin = 1`) or by column (`margin = 2`). Again, this is most clearly seen by looking at examples:

```
prop.table( x = itng.table, margin = 1)
```

```
##             utterance
## speaker       ee onk  oo pip
##   makka-pakka 0.0 0.5 0.0 0.5
##   tombliboo   0.5 0.0 0.5 0.0
##   upsy-daisy  0.0 0.5 0.0 0.5
```

Notice that each row now sums to 1, but that's not true for each column. What we're looking at here is the proportions of utterances made by each character. In other words, 50% of Makka-Pakka's utterances are "pip", and the other 50% are "onk". Let's contrast this with the following command:

```
prop.table( x = itng.table, margin = 2)
```

```
##             utterance
## speaker        ee onk  oo pip
##   makka-pakka 0.0 0.5 0.0 0.5
##   tombliboo   1.0 0.0 1.0 0.0
##   upsy-daisy  0.0 0.5 0.0 0.5
```

Now the columns all sum to 1 but the rows don't. In this version, what we're seeing is the proportion of characters associated with each utterance. For instance, whenever the utterance "ee" is made (in this data set), 100% of the time it's a Tombliboo saying it.

17.1.4 level tabulation

One final function I want to mention is the `tabulate()` function, since this is actually the low-level function that does most of the hard work. It takes a numeric vector as input, and outputs frequencies as outputs:

```
some.data <- c(1,2,3,1,1,3,1,1,2,8,3,1,2,4,2,3,5,2)
tabulate(some.data)
```

```
## [1] 6 5 4 1 1 0 0 1
```

This page titled 17.1: Tabulating and Cross-tabulating Data is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **7.1: Tabulating and Cross-tabulating Data** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

17.2: Transforming and Recoding a Variable

It's not uncommon in real world data analysis to find that one of your variables isn't quite equivalent to the variable that you really want. For instance, it's often convenient to take a continuous-valued variable (e.g., age) and break it up into a smallish number of categories (e.g., younger, middle, older). At other times, you may need to convert a numeric variable into a different numeric variable (e.g., you may want to analyse at the absolute value of the original variable). In this section I'll describe a few key tricks that you can make use of to do this.

17.2.1 Creating a transformed variable

The first trick to discuss is the idea of ***transforming*** a variable. Taken literally, *anything* you do to a variable is a transformation, but in practice what it usually means is that you apply a relatively simple mathematical function to the original variable, in order to create new variable that either (a) provides a better way of describing the thing you're actually interested in or (b) is more closely in agreement with the assumptions of the statistical tests you want to do. Since – at this stage – I haven't talked about statistical tests or their assumptions, I'll show you an example based on the first case.

To keep the explanation simple, the variable we'll try to transform (`likert.raw`) isn't inside a data frame, though in real life it almost certainly would be. However, I think it's useful to start with an example that doesn't use data frames because it illustrates the fact that you already know how to do variable transformations. To see this, let's go through an example. Suppose I've run a short study in which I ask 10 people a single question:

On a scale of 1 (strongly disagree) to 7 (strongly agree), to what extent do you agree with the proposition that "Dinosaurs are awesome"?

Now let's load and look at the data. The data file `likert.Rdata` contains a single variable that contains the raw Likert-scale responses:

```
load("./rbook-master/data/likert.Rdata")
likert.raw
```

```
##  [1] 1 7 3 4 4 4 2 6 5 5
```

However, if you think about it, this isn't the best way to represent these responses. Because of the fairly symmetric way that we set up the response scale, there's a sense in which the midpoint of the scale should have been coded as 0 (no opinion), and the two endpoints should be +3 (strong agree) and −3 (strong disagree). By recoding the data in this way, it's a bit more reflective of how we really think about the responses. The recoding here is trivially easy: we just subtract 4 from the raw scores:

```
likert.centred <- likert.raw - 4
likert.centred
```

```
##  [1] -3  3 -1  0  0  0 -2  2  1  1
```

One reason why it might be useful to have the data in this format is that there are a lot of situations where you might prefer to analyse the *strength* of the opinion separately from the *direction* of the opinion. We can do two different transformations on this `likert.centred` variable in order to distinguish between these two different concepts. Firstly, to compute an `opinion.strength` variable, we want to take the absolute value of the centred data (using the `abs()` function that we've seen previously), like so:

```
opinion.strength <- abs( likert.centred )
opinion.strength
```

```
##  [1] 3 3 1 0 0 0 2 2 1 1
```

Secondly, to compute a variable that contains only the direction of the opinion and ignores the strength, we can use the `sign()` function to do this. If you type `?sign` you'll see that this function is really simple: all negative numbers are converted to −1, all positive numbers are converted to 1 and zero stays as 0. So, when we apply the `sign()` function we obtain the following:

```
opinion.dir <- sign( likert.centred )
opinion.dir
```

```
##  [1] -1  1 -1  0  0  0 -1  1  1  1
```

And we're done. We now have three shiny new variables, all of which are useful transformations of the original `likert.raw` data. All of this should seem pretty familiar to you. The tools that you use to do regular calculations in R (e.g., Chapters 3 and 4) are very much the same ones that you use to transform your variables! To that end, in Section 7.3 I'll revisit the topic of doing calculations in R because there's a lot of other functions and operations that are worth knowing about.

Before moving on, you might be curious to see what these calculations look like if the data had started out in a data frame. To that end, it may help to note that the following example does all of the calculations using variables inside a data frame, and stores the variables created inside it:

```
df <- data.frame( likert.raw )                  # create data frame
df$likert.centred <- df$likert.raw - 4          # create centred data
df$opinion.strength <- abs( df$likert.centred ) # create strength variable
df$opinion.dir <- sign( df$likert.centred )     # create direction variable
df
```

```
##    likert.raw likert.centred opinion.strength opinion.dir
## 1           1             -3                3          -1
## 2           7              3                3           1
## 3           3             -1                1          -1
## 4           4              0                0           0
## 5           4              0                0           0
## 6           4              0                0           0
## 7           2             -2                2          -1
## 8           6              2                2           1
## 9           5              1                1           1
## 10          5              1                1           1
```

In other words, the commands you use are basically ones as before: it's just that every time you want to read a variable from the data frame or write to the data frame, you use the `$` operator. That's the easiest way to do it, though I should make note of the fact that people sometimes make use of the `within()` function to do the same thing. However, since (a) I don't use the `within()` function anywhere else in this book, and (b) the `$` operator works just fine, I won't discuss it any further.

17.2.2 Cutting a numeric variable into categories

One pragmatic task that arises more often than you'd think is the problem of cutting a numeric variable up into discrete categories. For instance, suppose I'm interested in looking at the age distribution of people at a social gathering:

```
age <- c( 60,58,24,26,34,42,31,30,33,2,9 )
```

In some situations it can be quite helpful to group these into a smallish number of categories. For example, we could group the data into three broad categories: young (0-20), adult (21-40) and older (41-60). This is a quite coarse-grained classification, and the labels that I've attached only make sense in the context of this data set (e.g., viewed more generally, a 42 year old wouldn't consider themselves as "older"). We can slice this variable up quite easily using the `cut()` function.[105] To make things a little cleaner, I'll start by creating a variable that defines the boundaries for the categories:

```
age.breaks <- seq( from = 0, to = 60, by = 20 )
age.breaks
```

```
## [1]  0 20 40 60
```

and another one for the labels:

```
age.labels <- c( "young", "adult", "older" )
age.labels
```

```
## [1] "young" "adult" "older"
```

Note that there are four numbers in the `age.breaks` variable, but only three labels in the `age.labels` variable; I've done this because the `cut()` function requires that you specify the *edges* of the categories rather than the mid-points. In any case, now that we've done this, we can use the `cut()` function to assign each observation to one of these three categories. There are several arguments to the `cut()` function, but the three that we need to care about are:

- `x` . The variable that needs to be categorised.
- `breaks` . This is either a vector containing the locations of the breaks separating the categories, or a number indicating how many categories you want.
- `labels` . The labels attached to the categories. This is optional: if you don't specify this R will attach a boring label showing the range associated with each category.

Since we've already created variables corresponding to the breaks and the labels, the command we need is just:

```
age.group <- cut( x = age,               # the variable to be categorised
                  breaks = age.breaks,   # the edges of the categories
                  labels = age.labels )  # the labels for the categories
```

Note that the output variable here is a factor. In order to see what this command has actually done, we could just print out the `age.group` variable, but I think it's actually more helpful to create a data frame that includes both the original variable and the categorised one, so that you can see the two side by side:

```
data.frame(age, age.group)
```

```
##    age age.group
## 1   60     older
## 2   58     older
## 3   24     adult
## 4   26     adult
## 5   34     adult
## 6   42     older
## 7   31     adult
## 8   30     adult
## 9   33     adult
## 10   2     young
## 11   9     young
```

It can also be useful to tabulate the output, just to see if you've got a nice even division of the sample:

```
table( age.group )
```

```
## age.group
## young adult older
##     2     6     3
```

In the example above, I made all the decisions myself. Much like the `hist()` function that we saw in Chapter 6, if you want to you can delegate a lot of the choices to R. For instance, if you want you can specify the *number* of categories you want, rather than giving explicit ranges for them, and you can allow R to come up with some labels for the categories. To give you a sense of how this works, have a look at the following example:

```
age.group2 <- cut( x = age, breaks = 3 )
```

With this command, I've asked for three categories, but let R make the choices for where the boundaries should be. I won't bother to print out the `age.group2` variable, because it's not terribly pretty or very interesting. Instead, all of the important information can be extracted by looking at the tabulated data:

```
table( age.group2 )
```

```
## age.group2
## (1.94,21.3] (21.3,40.7] (40.7,60.1]
##           2           6           3
```

This output takes a little bit of interpretation, but it's not complicated. What R has done is determined that the lowest age category should run from 1.94 years up to 21.3 years, the second category should run from 21.3 years to 40.7 years, and so on. The formatting on those labels might look a bit funny to those of you who haven't studied a lot of maths, but it's pretty simple. When R describes the first category as corresponding to the range (1.94,21.3] what it's saying is that the range consists of those numbers that are larger than 1.94 but less than *or equal to* 21.3. In other words, the weird asymmetric brackets is R s way of telling you that if there happens to be a value that is exactly equal to 21.3, then it belongs to the first category, not the second one. Obviously, this isn't actually possible since I've only specified the ages to the nearest whole number, but R doesn't know this and so it's trying to be precise just in case. This notation is actually pretty standard, but I suspect not everyone reading the book will have seen it before. In any case, those labels are pretty ugly, so it's usually a good idea to specify your own, meaningful labels to the categories.

Before moving on, I should take a moment to talk a little about the mechanics of the `cut()` function. Notice that R has tried to divide the `age` variable into three roughly equal sized bins. Unless you specify the particular breaks you want, that's what it will do. But suppose you want to divide the `age` variable into three categories of different size, but with approximately identical numbers of people. How would you do that? Well, if that's the case, then what you want to do is have the breaks correspond to the 0th, 33rd, 66th and 100th percentiles of the data. One way to do this would be to calculate those values using the `quantiles()` function and then use those quantiles as input to the `cut()` function. That's pretty easy to do, but it does take a couple of lines to type. So instead, the `lsr` package has a function called `quantileCut()` that does exactly this:

```
age.group3 <- quantileCut( x = age, n = 3 )
table( age.group3 )
```

```
## age.group3
## (1.94,27.3] (27.3,33.7] (33.7,60.1]
##           4           3           4
```

Notice the difference in the boundaries that the `quantileCut()` function selects. The first and third categories now span an age range of about 25 years each, whereas the middle category has shrunk to a span of only 6 years. There are some situations where this is genuinely what you want (that's why I wrote the function!), but in general you should be careful. Usually the numeric variable that you're trying to cut into categories is already expressed in meaningful units (i.e., it's interval scale), but if you cut it into unequal bin sizes then it's often very difficult to attach meaningful interpretations to the resulting categories.

More generally, regardless of whether you're using the original `cut()` function or the `quantileCut()` version, it's important to take the time to figure out whether or not the resulting categories make any sense at all in terms of your research project. If they don't make any sense to you as meaningful categories, then any data analysis that uses those categories is likely to be just as meaningless. More generally, in practice I've noticed that people have a very strong desire to carve their (continuous and messy) data into a few (discrete and simple) categories; and then run analysis using the categorised data instead of the original one.[106] I wouldn't go so far as to say that this is an inherently bad idea, but it does have some fairly serious drawbacks at times, so I would advise some caution if you are thinking about doing it.

This page titled 17.2: Transforming and Recoding a Variable is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **7.2: Transforming and Recoding a Variable** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

17.3: A few More Mathematical Functions and Operations

In Section 7.2 I discussed the ideas behind variable transformations, and showed that a lot of the transformations that you might want to apply to your data are based on fairly simple mathematical functions and operations, of the kind that we discussed in Chapter 3. In this section I want to return to that discussion, and mention several other mathematical functions and arithmetic operations that I didn't bother to mention when introducing you to R, but are actually quite useful for a lot of real world data analysis. Table 7.1 gives a brief overview of the various mathematical functions I want to talk about (and some that I already have talked about). Obviously this doesn't even come close to cataloging the range of possibilities available in R, but it does cover a very wide range of functions that are used in day to day data analysis.

Table 7.1: Some of the mathematical functions available in R.

mathematical.function	R.function	example.input	answer
square root	sqrt()	sqrt(25)	5
absolute value	abs()	abs(-23)	23
logarithm (base 10)	log10()	log10(1000)	3
logarithm (base e)	log()	log(1000)	6.908
exponentiation	exp()	exp(6.908)	1000.245
rounding to nearest	round()	round(1.32)	1
rounding down	floor()	floor(1.32)	1
rounding up	ceiling()	ceiling(1.32)	2

17.3.1 Rounding a number

One very simple transformation that crops up surprisingly often is the need to round a number to the nearest whole number, or to a certain number of significant digits. To start with, let's assume that we want to round to a whole number. To that end, there are three useful functions in R you want to know about: `round()`, `floor()` and `ceiling()`. The `round()` function just rounds to the *nearest* whole number. So if you round the number 4.3, it "rounds down" to `4`, like so:

```
round( x = 4.3 )
```

```
## [1] 4
```

In contrast, if we want to round the number 4.7, we would round upwards to 5. In everyday life, when someone talks about "rounding", they usually mean "round to nearest", so this is the function we use most of the time. However sometimes you have reasons to want to always round up or always round down. If you want to always round down, use the `floor()` function instead; and if you want to force R to round up, then use `ceiling()`. That's the only difference between the three functions. What if you want to round to a certain number of digits? Let's suppose you want to round to a fixed number of decimal places, say 2 decimal places. If so, what you need to do is specify the `digits` argument to the `round()` function. That's pretty straightforward:

```
round( x = 0.0123, digits = 2 )
```

```
## [1] 0.01
```

The only subtlety that you need to keep in mind is that sometimes what you want to do is round to 2 ***significant digits*** and not to two decimal places. The difference is that, when determining the number of significant digits, zeros don't count. To see this, let's apply the `signif()` function instead of the `round()` function:

```
signif( x = 0.0123, digits = 2 )
```

```
## [1] 0.012
```

This time around, we get an answer of 0.012 because the zeros don't count as significant digits. Quite often scientific journals will ask you to report numbers to two or three significant digits, so it's useful to remember the distinction.

17.3.2 Modulus and integer division

Table 7.2: Two more arithmetic operations that sometimes come in handy

operation	operator	example.input	answer
integer division	%/%	42 %/% 10	4
modulus	%%	42 %% 10	2

Since we're on the topic of simple calculations, there are two other arithmetic operations that I should mention, since they can come in handy when working with real data. These operations are calculating a modulus and doing integer division. They don't come up anywhere else in this book, but they are worth knowing about. First, let's consider ***integer division***. Suppose I have $42 in my wallet, and want to buy some sandwiches, which are selling for $10 each. How many sandwiches can I afford[107] to buy? The answer is of course 4. Note that it's not 4.2, since no shop will sell me one-fifth of a sandwich. That's integer division. In R we perform integer division by using the `%/%` operator:

```
42 %/% 10
```

```
## [1] 4
```

Okay, that's easy enough. What about the ***modulus***? Basically, a modulus is the remainder after integer division, and it's calculated using the `%%` operator. For the sake of argument, let's suppose I buy four overpriced $10 sandwiches. If I started out with $42, how much money do I have left? The answer, as both R and common sense tells us, is $2:

```
42 %% 10
```

```
## [1] 2
```

So that's also pretty easy. There is, however, one subtlety that I need to mention, and this relates to how negative numbers are handled. Firstly, what would happen if I tried to do integer division with a negative number? Let's have a look:

```
-42 %/% 10
```

```
## [1] -5
```

This might strike you as counterintuitive: why does `42 %/% 10` produce an answer of `4`, but `-42 %/% 10` gives us an answer of `-5`? Intuitively you might think that the answer to the second one should be `-4`. The way to think about it is like this. Suppose I *owe* the sandwich shop $42, but I don't have any money. How many sandwiches would *I* have to give *them* in order

to stop them from calling security? The answer[108] here is 5, not 4. If I handed them 4 sandwiches, I'd still owe them $2, right? So I actually have to give them 5 sandwiches. And since it's *me* giving them the sandwiches, the answer to `-42 %/% 10` is `-5` . As you might expect, the behaviour of the modulus operator has a similar pattern. If I've handed 5 sandwiches over to the shop in order to pay off my debt of $42, then *they* now owe me $8. So the modulus is now:

```
-42 %% 10
```

```
## [1] 8
```

17.3.3 Logarithms and exponentials

As I've mentioned earlier, R has an incredible range of mathematical functions built into it, and there really wouldn't be much point in trying to describe or even list all of them. For the most part, I've focused only on those functions that are strictly necessary for this book. However I do want to make an exception for logarithms and exponentials. Although they aren't needed anywhere else in this book, they are *everywhere* in statistics more broadly, and not only that, there are a *lot* of situations in which it is convenient to analyse the logarithm of a variable (i.e., to take a "log-transform" of the variable). I suspect that many (maybe most) readers of this book will have encountered logarithms and exponentials before, but from past experience I know that there's a substantial proportion of students who take a social science statistics class who haven't touched logarithms since high school, and would appreciate a bit of a refresher.

In order to understand logarithms and exponentials, the easiest thing to do is to actually calculate them and see how they relate to other simple calculations. There are three R functions in particular that I want to talk about, namely `log()` , `log10()` and `exp()` . To start with, let's consider `log10()` , which is known as the "logarithm in base 10". The trick to understanding a ***logarithm*** is to understand that it's basically the "opposite" of taking a power. Specifically, the logarithm in base 10 is closely related to the powers of 10. So let's start by noting that 10-cubed is 1000. Mathematically, we would write this:

$$10^3=1000$$

and in R we'd calculate it by using the command `10^3` . The trick to understanding a logarithm is to recognise that the statement that "10 to the power of 3 is equal to 1000" is equivalent to the statement that "the logarithm (in base 10) of 1000 is equal to 3". Mathematically, we write this as follows,

$$\log_{10}(1000)=3$$

and if we wanted to do the calculation in R we would type this:

```
log10( 1000 )
```

```
## [1] 3
```

Obviously, since you already know that $10^3=1000$ there's really no point in getting R to tell you that the base-10 logarithm of 1000 is 3. However, most of the time you probably don't know what right answer is. For instance, I can honestly say that I didn't know that $10^{2.69897}=500$, so it's rather convenient for me that I can use R to calculate the base-10 logarithm of 500:

```
log10( 500 )
```

```
## [1] 2.69897
```

Or at least it would be convenient if I had a pressing need to know the base-10 logarithm of 500.

Okay, since the `log10()` function is related to the powers of 10, you might expect that there are other logarithms (in bases other than 10) that are related to other powers too. And of course that's true: there's not really anything mathematically special about the number 10. You and I happen to find it useful because decimal numbers are built around the number 10, but the big bad

world of mathematics scoffs at our decimal numbers. Sadly, the universe doesn't actually care how we write down numbers. Anyway, the consequence of this cosmic indifference is that there's nothing particularly special about calculating logarithms in base 10. You could, for instance, calculate your logarithms in base 2, and in fact R does provide a function for doing that, which is (not surprisingly) called `log2()`. Since we know that $23=2\times2\times2=8$, it's not surprise to see that

```
log2( 8 )
```

```
## [1] 3
```

Alternatively, a third type of logarithm – and one we see a lot more of in statistics than either base 10 or base 2 – is called the ***natural logarithm***, and corresponds to the logarithm in base e. Since you might one day run into it, I'd better explain what e is. The number e, known as ***Euler's number***, is one of those annoying "irrational" numbers whose decimal expansion is infinitely long, and is considered one of the most important numbers in mathematics. The first few digits of e are:

$$e=2.718282$$

There are quite a few situation in statistics that require us to calculate powers of e, though none of them appear in this book. Raising e to the power x is called the ***exponential*** of x, and so it's very common to see e^x written as exp(x). And so it's no surprise that R has a function that calculate exponentials, called `exp()`. For instance, suppose I wanted to calculate e^3. I could try typing in the value of e manually, like this:

```
2.718282 ^ 3
```

```
## [1] 20.08554
```

but it's much easier to do the same thing using the `exp()` function:

```
exp( 3 )
```

```
## [1] 20.08554
```

Anyway, because the number e crops up so often in statistics, the natural logarithm (i.e., logarithm in base e) also tends to turn up. Mathematicians often write it as loge(x) or ln(x), or sometimes even just log(x). In fact, R works the same way: the `log()` function corresponds to the natural logarithm[109] Anyway, as a quick check, let's calculate the natural logarithm of 20.08554 using R:

```
log( 20.08554 )
```

```
## [1] 3
```

And with that, I think we've had quite enough exponentials and logarithms for this book!

This page titled 17.3: A few More Mathematical Functions and Operations is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **7.3: A few More Mathematical Functions and Operations** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

17.4: Extracting a Subset of a Vector

One very important kind of data handling is being able to extract a particular subset of the data. For instance, you might be interested only in analysing the data from one experimental condition, or you may want to look closely at the data from people over 50 years in age. To do this, the first step is getting R to extract the subset of the data corresponding to the observations that you're interested in. In this section I'll talk about subsetting as it applies to vectors, extending the discussion from Chapters 3 and 4. In Section 7.5 I'll go on to talk about how this discussion extends to data frames.

17.4.1 Refresher

This section returns to the `nightgarden.Rdata` data set. If you're reading this whole chapter in one sitting, then you should already have this data set loaded. If not, don't forget to use the `load("nightgarden.Rdata")` command. For this section, let's ignore the `itng` data frame that we created earlier, and focus instead on the two vectors `speaker` and `utterance` (see Section 7.1 if you've forgotten what those vectors look like). Suppose that what I want to do is pull out only those utterances that were made by Makka-Pakka. To that end, I could first use the equality operator to have R tell me which cases correspond to Makka-Pakka speaking:

```
is.MP.speaking <- speaker == "makka-pakka"
is.MP.speaking
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
```

and then use logical indexing to get R to print out those elements of `utterance` for which `is.MP.speaking` is true, like so:

```
utterance[ is.MP.speaking ]
```

```
## [1] "pip" "pip" "onk" "onk"
```

Or, since I'm lazy, I could collapse it to a single command like so:

```
utterance[ speaker == "makka-pakka" ]
```

```
## [1] "pip" "pip" "onk" "onk"
```

17.4.2 Using `%in%` match multiple cases

A second useful trick to be aware of is the `%in%` operator[110]. It's actually very similar to the `==` operator, except that you can supply a collection of acceptable values. For instance, suppose I wanted to keep only those cases when the utterance is either "pip" or "oo". One simple way do to this is:

```
utterance %in% c("pip","oo")
```

```
##  [1]  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE  TRUE FALSE FALSE
```

What this does if return `TRUE` for those elements of `utterance` that are either `"pip"` or `"oo"` and returns `FALSE` for all the others. What that means is that if I want a list of all those instances of characters speaking either of these two words, I could do this:

```
speaker[ utterance %in% c("pip","oo") ]
```

```
## [1] "upsy-daisy"  "upsy-daisy"  "tombliboo"   "makka-pakka" "makka-pakka"
```

17.4.3 Using negative indices to drop elements

Before moving onto data frames, there's a couple of other tricks worth mentioning. The first of these is to use negative values as indices. Recall from Section 3.10 that we can use a vector of numbers to extract a set of elements that we would like to keep. For instance, suppose I want to keep only elements 2 and 3 from `utterance`. I could do so like this:

```
utterance[2:3]
```

```
## [1] "pip" "onk"
```

But suppose, on the other hand, that I have discovered that observations 2 and 3 are untrustworthy, and I want to keep everything *except* those two elements. To that end, R lets you use negative numbers to remove specific values, like so:

```
utterance [ -(2:3) ]
```

```
## [1] "pip" "onk" "ee"  "oo"  "pip" "pip" "onk" "onk"
```

The output here corresponds to element 1 of the original vector, followed by elements 4, 5, and so on. When all you want to do is remove a few cases, this is a very handy convention.

17.4.4 Splitting a vector by group

One particular example of subsetting that is especially common is the problem of splitting one one variable up into several different variables, one corresponding to each group. For instance, in our *In the Night Garden* example, I might want to create subsets of the `utterance` variable for every character. One way to do this would be to just repeat the exercise that I went through earlier separately for each character, but that quickly gets annoying. A faster way do it is to use the `split()` function. The arguments are:

- `x`. The variable that needs to be split into groups.
- `f`. The grouping variable.

What this function does is output a list (Section 4.9), containing one variable for each group. For instance, I could split up the `utterance` variable by `speaker` using the following command:

```
speech.by.char <- split( x = utterance, f = speaker )
speech.by.char
```

```
## $`makka-pakka`
## [1] "pip" "pip" "onk" "onk"
##
## $tombliboo
## [1] "ee" "oo"
##
## $`upsy-daisy`
## [1] "pip" "pip" "onk" "onk"
```

Once you're starting to become comfortable working with lists and data frames, this output is all you need, since you can work with this list in much the same way that you would work with a data frame. For instance, if you want the first utterance made by Makka-Pakka, all you need to do is type this:

```
speech.by.char$`makka-pakka`[1]
```

```
## [1] "pip"
```

Just remember that R does need you to add the quoting characters (i.e. `` ` ``). Otherwise, there's nothing particularly new or difficult here.

However, sometimes – especially when you're just starting out – it can be convenient to pull these variables out of the list, and into the workspace. This isn't too difficult to do, though it can be a little daunting to novices. To that end, I've included a function called `importList()` in the `lsr` package that does this.[111] First, here's what you'd have if you had wiped the workspace before the start of this section:

```
who()
```

```
##    -- Name --          -- Class --    -- Size --
##    age                 numeric        11
##    age.breaks          numeric        4
##    age.group           factor         11
##    age.group2          factor         11
##    age.group3          factor         11
##    age.labels          character      3
##    df                  data.frame     10 x 4
##    is.MP.speaking      logical        10
##    itng                data.frame     10 x 2
##    itng.table          table          3 x 4
##    likert.centred      numeric        10
##    likert.raw          numeric        10
##    opinion.dir         numeric        10
##    opinion.strength    numeric        10
##    some.data           numeric        18
##    speaker             character      10
##    speech.by.char      list           3
##    utterance           character      10
```

Now we use the `importList()` function to copy all of the variables within the `speech.by.char` list:

```
importList( speech.by.char, ask = FALSE)
```

Because the `importList()` function is attempting to create new variables based on the names of the elements of the list, it pauses to check that you're okay with the variable names. The reason it does this is that, if one of the to-be-created variables has the same name as a variable that you already have in your workspace, that variable will end up being overwritten, so it's a good idea to check. Assuming that you type `y`, it will go on to create the variables. Nothing *appears* to have happened, but if we look at our workspace now:

```
who()
```

```
##    -- Name --          -- Class --    -- Size --
##    age                numeric        11
##    age.breaks         numeric        4
##    age.group          factor         11
##    age.group2         factor         11
##    age.group3         factor         11
##    age.labels         character      3
##    df                 data.frame     10 x 4
##    is.MP.speaking     logical        10
##    itng               data.frame     10 x 2
##    itng.table         table          3 x 4
##    likert.centred     numeric        10
##    likert.raw         numeric        10
##    makka.pakka        character      4
##    opinion.dir        numeric        10
##    opinion.strength   numeric        10
##    some.data          numeric        18
##    speaker            character      10
##    speech.by.char     list           3
##    tombliboo          character      2
##    upsy.daisy         character      4
##    utterance          character      10
```

we see that there are three new variables, called `makka.pakka`, `tombliboo` and `upsy.daisy`. Notice that the `importList()` function has converted the original character strings into valid R variable names, so the variable corresponding to `"makka-pakka"` is actually `makka.pakka`.[112] Nevertheless, even though the names can change, note that each of these variables contains the exact same information as the original elements of the list did. For example:

```
> makka.pakka
[1] "pip" "pip" "onk" "onk"
```

This page titled 17.4: Extracting a Subset of a Vector is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **7.4: Extracting a Subset of a Vector** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

17.5: Extracting a Subset of a Data Frame

In this section we turn to the question of how to subset a data frame rather than a vector. To that end, the first thing I should point out is that, if all you want to do is subset *one* of the variables inside the data frame, then as usual the `$` operator is your friend. For instance, suppose I'm working with the `itng` data frame, and what I want to do is create the `speech.by.char` list. I can use the exact same tricks that I used last time, since what I really want to do is `split()` the `itng$utterance` vector, using the `itng$speaker` vector as the grouping variable. However, most of the time what you actually want to do is select several different variables within the data frame (i.e., keep only some of the columns), or maybe a subset of cases (i.e., keep only some of the rows). In order to understand how this works, we need to talk more specifically about data frames and how to subset them.

Using the `subset()` function

There are several different ways to subset a data frame in R, some easier than others. I'll start by discussing the `subset()` function, which is probably the conceptually simplest way do it. For our purposes there are three different arguments that you'll be most interested in:

- `x` . The data frame that you want to subset.
- `subset` . A vector of logical values indicating which cases (rows) of the data frame you want to keep. By default, all cases will be retained.
- `select` . This argument indicates which variables (columns) in the data frame you want to keep. This can either be a list of variable names, or a logical vector indicating which ones to keep, or even just a numeric vector containing the relevant column numbers. By default, all variables will be retained.

Let's start with an example in which I use all three of these arguments. Suppose that I want to subset the `itng` data frame, keeping only the utterances made by Makka-Pakka. What that means is that I need to use the `select` argument to pick out the `utterance` variable, and I also need to use the `subset` variable, to pick out the cases when Makka-Pakka is speaking (i.e., `speaker == "makka-pakka"`). Therefore, the command I need to use is this:

```
df <- subset( x = itng,                          # data frame is itng
              subset = speaker == "makka-pakka",  # keep only Makka-Pakkas speech
              select = utterance )                # keep only the utterance variable
print( df )
```

```
##    utterance
## 7        pip
## 8        pip
## 9        onk
## 10       onk
```

The variable `df` here is still a data frame, but it only contains one variable (called `utterance`) and four cases. Notice that the row numbers are actually the same ones from the original data frame. It's worth taking a moment to briefly explain this. The reason that this happens is that these "row numbers' are actually row *names*. When you create a new data frame from scratch R will assign each row a fairly boring row name, which is identical to the row number. However, when you subset the data frame, each row keeps its original row name. This can be quite useful, since – as in the current example – it provides you a visual reminder of what each row in the new data frame corresponds to in the original data frame. However, if it annoys you, you can change the row names using the `rownames()` function.[113]

In any case, let's return to the `subset()` function, and look at what happens when we don't use all three of the arguments. Firstly, suppose that I didn't bother to specify the `select` argument. Let's see what happens:

```
subset( x = itng,
        subset = speaker == "makka-pakka" )
```

```
##        speaker utterance
## 7  makka-pakka       pip
## 8  makka-pakka       pip
## 9  makka-pakka       onk
## 10 makka-pakka       onk
```

Not surprisingly, R has kept the same cases from the original data set (i.e., rows 7 through 10), but this time it has kept all of the variables from the data frame. Equally unsurprisingly, if I don't specify the `subset` argument, what we find is that R keeps all of the cases:

```
subset( x = itng,
         select = utterance )
```

```
##    utterance
## 1        pip
## 2        pip
## 3        onk
## 4        onk
## 5         ee
## 6         oo
## 7        pip
## 8        pip
## 9        onk
## 10       onk
```

Again, it's important to note that this output is still a data frame: it's just a data frame with only a single variable.

Using square brackets: I. Rows and columns

Throughout the book so far, whenever I've been subsetting a vector I've tended use the square brackets `[]` to do so. But in the previous section when I started talking about subsetting a data frame I used the `subset()` function. As a consequence, you might be wondering whether it is possible to use the square brackets to subset a data frame. The answer, of course, is yes. Not only *can* you use square brackets for this purpose, as you become more familiar with R you'll find that this is actually much more convenient than using `subset()`. Unfortunately, the use of square brackets for this purpose is somewhat complicated, and can be very confusing to novices. So be warned: this section is more complicated than it feels like it "should" be. With that warning in place, I'll try to walk you through it slowly. For this section, I'll use a slightly different data set, namely the `garden` data frame that is stored in the `"nightgarden2.Rdata"` file.

```
load("./rbook-master/data/nightgarden2.Rdata" )
garden
```

```
##            speaker utterance line
## case.1  upsy-daisy       pip    1
## case.2  upsy-daisy       pip    2
## case.3   tombliboo        ee    5
## case.4 makka-pakka       pip    7
## case.5 makka-pakka       onk    9
```

As you can see, the `garden` data frame contains 3 variables and 5 cases, and this time around I've used the `rownames()` function to attach slightly verbose labels to each of the cases. Moreover, let's assume that what we want to do is to pick out rows 4 and 5 (the two cases when Makka-Pakka is speaking), and columns 1 and 2 (variables `speaker` and `utterance`).

How shall we do this? As usual, there's more than one way. The first way is based on the observation that, since a data frame is basically a table, every element in the data frame has a row number and a column number. So, if we want to pick out a single element, we have to specify the row number *and* a column number within the square brackets. By convention, the row number comes first. So, for the data frame above, which has 5 rows and 3 columns, the numerical indexing scheme looks like this:

```
knitr::kable(data.frame(stringsAsFactors=FALSE, row = c("1","2","3", "4", "5"), col1
"[5,1]"), col2 = c("[1,2]", "[2,2]", "[3,2]", "[4,2]", "[5,2]"), col3 = c("[1,3]", "[
```

row	col1	col2	col3
1	[1,1]	[1,2]	[1,3]
2	[2,1]	[2,2]	[2,3]
3	[3,1]	[3,2]	[3,3]
4	[4,1]	[4,2]	[4,3]
5	[5,1]	[5,2]	[5,3]

If I want the 3rd case of the 2nd variable, what I would type is `garden[3,2]`, and R would print out some output showing that, this element corresponds to the utterance `"ee"`. However, let's hold off from actually doing that for a moment, because there's something slightly counterintuitive about the specifics of what R does under those circumstances (see Section 7.5.4). Instead, let's aim to solve our original problem, which is to pull out two rows (4 and 5) and two columns (1 and 2). This is fairly simple to do, since R allows us to specify multiple rows and multiple columns. So let's try that:

```
garden[ 4:5, 1:2 ]
```

```
##            speaker utterance
## case.4 makka-pakka       pip
## case.5 makka-pakka       onk
```

Clearly, that's exactly what we asked for: the output here is a data frame containing two variables and two cases. Note that I could have gotten the same answer if I'd used the `c()` function to produce my vectors rather than the `:` operator. That is, the following command is equivalent to the last one:

```
garden[ c(4,5), c(1,2) ]
```

```
##            speaker utterance
## case.4 makka-pakka       pip
## case.5 makka-pakka       onk
```

It's just not as pretty. However, if the columns and rows that you want to keep don't happen to be next to each other in the original data frame, then you might find that you have to resort to using commands like `garden[ c(2,4,5), c(1,3) ]` to extract them.

A second way to do the same thing is to use the names of the rows and columns. That is, instead of using the row numbers and column numbers, you use the character strings that are used as the labels for the rows and columns. To apply this idea to our `garden` data frame, we would use a command like this:

```
garden[ c("case.4", "case.5"), c("speaker", "utterance") ]
```

```
##            speaker utterance
## case.4 makka-pakka       pip
## case.5 makka-pakka       onk
```

Once again, this produces exactly the same output, so I haven't bothered to show it. Note that, although this version is more annoying to *type* than the previous version, it's a bit easier to *read*, because it's often more meaningful to refer to the elements by their names rather than their numbers. Also note that you don't have to use the same convention for the rows and columns. For instance, I often find that the variable names are meaningful and so I sometimes refer to them by name, whereas the row names are pretty arbitrary so it's easier to refer to them by number. In fact, that's more or less exactly what's happening with the `garden` data frame, so it probably makes more sense to use this as the command:

```
garden[ 4:5, c("speaker", "utterance") ]
```

```
##            speaker utterance
## case.4 makka-pakka       pip
## case.5 makka-pakka       onk
```

Again, the output is identical.

Finally, both the rows and columns can be indexed using logicals vectors as well. For example, although I *claimed* earlier that my goal was to extract cases 4 and 5, it's pretty obvious that what I really wanted to do was select the cases where Makka-Pakka is speaking. So what I could have done is create a logical vector that indicates which cases correspond to Makka-Pakka speaking:

```
is.MP.speaking <- garden$speaker == "makka-pakka"
is.MP.speaking
```

```
## [1] FALSE FALSE FALSE  TRUE  TRUE
```

As you can see, the 4th and 5th elements of this vector are `TRUE` while the others are `FALSE`. Now that I've constructed this "indicator" variable, what I can do is use this vector to select the rows that I want to keep:

```
garden[ is.MP.speaking, c("speaker", "utterance") ]
```

```
##            speaker utterance
## case.4 makka-pakka       pip
## case.5 makka-pakka       onk
```

And of course the output is, yet again, the same.

Using square brackets: II. Some elaborations

There are two fairly useful elaborations on this "rows and columns" approach that I should point out. Firstly, what if you want to keep all of the rows, or all of the columns? To do this, all we have to do is leave the corresponding entry blank, but it is crucial to remember to keep the comma*! For instance, suppose I want to keep all the rows in the `garden` data, but I only want to retain the first two columns. The easiest way do this is to use a command like this:

```
garden[ , 1:2 ]
```

```
##            speaker utterance
## case.1  upsy-daisy       pip
## case.2  upsy-daisy       pip
## case.3   tombliboo        ee
## case.4 makka-pakka       pip
## case.5 makka-pakka       onk
```

Alternatively, if I want to keep all the columns but only want the last two rows, I use the same trick, but this time I leave the second index blank. So my command becomes:

```
garden[ 4:5, ]
```

```
##            speaker utterance line
## case.4 makka-pakka       pip    7
## case.5 makka-pakka       onk    9
```

The second elaboration I should note is that it's still okay to use negative indexes as a way of telling R to delete certain rows or columns. For instance, if I want to delete the 3rd column, then I use this command:

```
garden[ , -3 ]
```

```
##            speaker utterance
## case.1  upsy-daisy       pip
## case.2  upsy-daisy       pip
## case.3   tombliboo        ee
## case.4 makka-pakka       pip
## case.5 makka-pakka       onk
```

whereas if I want to delete the 3rd row, then I'd use this one:

```
garden[ -3,  ]
```

```
##            speaker utterance line
## case.1  upsy-daisy       pip    1
## case.2  upsy-daisy       pip    2
## case.4 makka-pakka       pip    7
## case.5 makka-pakka       onk    9
```

So that's nice.

Using square brackets: III. Understanding "dropping"

At this point some of you might be wondering why I've been so terribly careful to choose my examples in such a way as to ensure that the output always has are multiple rows and multiple columns. The reason for this is that I've been trying to hide the somewhat curious "dropping" behaviour that R produces when the output only has a single column. I'll start by showing you what happens, and then I'll try to explain it. Firstly, let's have a look at what happens when the output contains only a single *row*:

```
garden[ 5, ]
```

```
##            speaker utterance line
## case.5 makka-pakka       onk    9
```

This is exactly what you'd expect to see: a data frame containing three variables, and only one case per variable. Okay, no problems so far. What happens when you ask for a single *column*? Suppose, for instance, I try this as a command:

```
garden[ , 3 ]
```

Based on everything that I've shown you so far, you would be well within your rights to expect to see R produce a data frame containing a single variable (i.e., `line`) and five cases. After all, that *is* what the `subset()` command does in this situation, and it's pretty consistent with everything else that I've shown you so far about how square brackets work. In other words, you should expect to see this:

```
       line
case.1    1
case.2    2
case.3    5
case.4    7
case.5    9
```

However, that is emphatically not what happens. What you actually get is this:

```
garden[ , 3 ]
```

```
## [1] 1 2 5 7 9
```

That output is *not a data frame* at all! That's just an ordinary numeric vector containing 5 elements. What's going on here is that R has "noticed" that the output that we've asked for doesn't really "need" to be wrapped up in a data frame at all, because it only corresponds to a single variable. So what it does is "drop" the output from a data frame *containing* a single variable, "down" to a simpler output that corresponds to that variable. This behaviour is actually very convenient for day to day usage once you've become familiar with it – and I suppose that's the real reason why R does this – but there's no escaping the fact that it is *deeply* confusing to novices. It's especially confusing because the behaviour appears only for a very specific case: (a) it only works for

columns and not for rows, because the columns correspond to variables and the rows do not, and (b) it only applies to the "rows and columns" version of the square brackets, and not to the `subset()` function,[114] or to the "just columns" use of the square brackets (next section). As I say, it's very confusing when you're just starting out. For what it's worth, you can suppress this behaviour if you want, by setting `drop = FALSE` when you construct your bracketed expression. That is, you could do something like this:

```
garden[ , 3, drop = FALSE ]
```

```
##        line
## case.1    1
## case.2    2
## case.3    5
## case.4    7
## case.5    9
```

I suppose that helps a little bit, in that it gives you some control over the dropping behaviour, but I'm not sure it helps to make things any easier to understand. Anyway, that's the "dropping" special case. Fun, isn't it?

Using square brackets: IV. Columns only

As if the weird "dropping" behaviour wasn't annoying enough, R actually provides a completely different way of using square brackets to index a data frame. Specifically, if you *only* give a single index, R will assume you want the corresponding columns, not the rows. Do not be fooled by the fact that this second method also uses square brackets: it behaves differently to the "rows and columns" method that I've discussed in the last few sections. Again, what I'll do is show you *what* happens first, and then I'll try to explain *why* it happens afterwards. To that end, let's start with the following command:

```
garden[ 1:2 ]
```

```
##           speaker utterance
## case.1  upsy-daisy       pip
## case.2  upsy-daisy       pip
## case.3   tombliboo        ee
## case.4 makka-pakka       pip
## case.5 makka-pakka       onk
```

As you can see, the output gives me the first two columns, much as if I'd typed `garden[,1:2]` . It doesn't give me the first two rows, which is what I'd have gotten if I'd used a command like `garden[1:2,]` . Not only that, if I ask for a *single* column, R does not drop the output:

```
garden[3]
```

```
##        line
## case.1    1
## case.2    2
## case.3    5
## case.4    7
## case.5    9
```

As I said earlier, the *only* case where dropping occurs by default is when you use the "row and columns" version of the square brackets, and the output happens to correspond to a single column. However, if you really want to force R to drop the output, you can do so using the "double brackets" notation:

```
garden[[3]]
```

```
## [1] 1 2 5 7 9
```

Note that R will only allow you to ask for one column at a time using the double brackets. If you try to ask for multiple columns in this way, you get completely different behaviour,[115] which may or may not produce an error, but definitely won't give you the output you're expecting. The only reason I'm mentioning it at all is that you might run into double brackets when doing further reading, and a lot of books don't explicitly point out the difference between `[` and `[[`. However, I promise that I won't be using `[[` anywhere else in this book.

Okay, for those few readers that have persevered with this section long enough to get here without having set fire to the book, I should explain *why* R has these two different systems for subsetting a data frame (i.e., "row and column" versus "just columns"), and why they behave so differently to each other. I'm not 100% sure about this since I'm still reading through some of the old references that describe the early development of R, but I think the answer relates to the fact that data frames are actually a very strange hybrid of two different kinds of thing. At a low level, a data frame is a list (Section 4.9). I can demonstrate this to you by overriding the normal `print()` function[116] and forcing R to print out the `garden` data frame using the default print method rather than the special one that is defined only for data frames. Here's what we get:

```
print.default( garden )
```

```
## $speaker
## [1] upsy-daisy  upsy-daisy  tombliboo   makka-pakka makka-pakka
## Levels: makka-pakka tombliboo upsy-daisy
##
## $utterance
## [1] pip pip ee  pip onk
## Levels: ee onk oo pip
##
## $line
## [1] 1 2 5 7 9
##
## attr(,"class")
## [1] "data.frame"
```

Apart from the weird part of the output right at the bottom, this is *identical* to the print out that you get when you print out a list (see Section 4.9). In other words, a data frame is a list. View from this "list based" perspective, it's clear what `garden[1]` is: it's the first variable stored in the list, namely `speaker`. In other words, when you use the "just columns" way of indexing a data frame, using only a single index, R assumes that you're thinking about the data frame as if it were a *list of variables*. In fact, when you use the `$` operator you're taking advantage of the fact that the data frame is secretly a list.

However, a data frame is more than just a list. It's a very special kind of list where all the variables are of the same length, and the first element in each variable happens to correspond to the first "case" in the data set. That's why no-one ever wants to see a data frame printed out in the default "list-like" way that I've shown in the extract above. In terms of the deeper *meaning* behind what a data frame is used for, a data frame really does have this rectangular shape to it:

```
print( garden )
```

```
##            speaker utterance line
## case.1  upsy-daisy       pip    1
## case.2  upsy-daisy       pip    2
## case.3   tombliboo        ee    5
## case.4 makka-pakka       pip    7
## case.5 makka-pakka       onk    9
```

Because of the fact that a data frame is basically a table of data, R provides a second "row and column" method for interacting with the data frame (see Section 7.11.1 for a related example). This method makes much more sense in terms of the high-level *table of data* interpretation of what a data frame is, and so for the most part it's this method that people tend to prefer. In fact, throughout the rest of the book I will be sticking to the "row and column" approach (though I will use `$` a lot), and never again referring to the "just columns" approach. However, it does get used a lot in practice, so I think it's important that this book explain what's going on.

And now let us never speak of this again.

This page titled 17.5: Extracting a Subset of a Data Frame is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **7.5: Extracting a Subset of a Data Frame** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

17.6: Sorting, Flipping and Merging Data

In this section I discuss a few useful operations that I feel are loosely related to one another: sorting a vector, sorting a data frame, binding two or more vectors together into a data frame (or matrix), and flipping a data frame (or matrix) on its side. They're all fairly straightforward tasks, at least in comparison to some of the more obnoxious data handling problems that turn up in real life.

17.6.1 Sorting a numeric or character vector

One thing that you often want to do is sort a variable. If it's a numeric variable you might want to sort in increasing or decreasing order. If it's a character vector you might want to sort alphabetically, etc. The `sort()` function provides this capability.

```
numbers <- c(2,4,3)
sort( x = numbers )
```

```
## [1] 2 3 4
```

You can ask for R to sort in decreasing order rather than increasing:

```
sort( x = numbers, decreasing = TRUE )
```

```
## [1] 4 3 2
```

And you can ask it to sort text data in alphabetical order:

```
text <- c("aardvark", "zebra", "swing")
sort( text )
```

```
## [1] "aardvark" "swing"    "zebra"
```

That's pretty straightforward. That being said, it's important to note that I'm glossing over something here. When you apply `sort()` to a character vector it doesn't strictly sort into alphabetical order. R actually has a slightly different notion of how characters are ordered (see Section 7.8.5 and Table 7.3), which is more closely related to how computers store text data than to how letters are ordered in the alphabet. However, that's a topic we'll discuss later. For now, the only thing I should note is that the `sort()` function doesn't alter the original variable. Rather, it creates a new, sorted variable as the output. So if I inspect my original `text` variable:

```
text
```

```
## [1] "aardvark" "zebra"    "swing"
```

I can see that it has remained unchanged.

17.6.2 Sorting a factor

You can also sort factors, but the story here is slightly more subtle because there's two different ways you can sort a factor: alphabetically (by label) or by factor level. The `sort()` function uses the latter. To illustrate, let's look at the two different examples. First, let's create a factor in the usual way:

```
fac <- factor( text )
fac
```

```
## [1] aardvark zebra    swing
## Levels: aardvark swing zebra
```

Now let's sort it:

```
sort(fac)
```

```
## [1] aardvark swing    zebra
## Levels: aardvark swing zebra
```

This *looks* like it's sorted things into alphabetical order, but that's only because the factor levels themselves happen to be alphabetically ordered. Suppose I deliberately define the factor levels in a non-alphabetical order:

```
fac <- factor( text, levels = c("zebra","swing","aardvark") )
fac
```

```
## [1] aardvark zebra    swing
## Levels: zebra swing aardvark
```

Now what happens when we try to sort `fac` this time? The answer:

```
sort(fac)
```

```
## [1] zebra    swing    aardvark
## Levels: zebra swing aardvark
```

It sorts the data into the numerical order implied by the factor levels, not the alphabetical order implied by the labels attached to those levels. Normally you never notice the distinction, because by default the factor levels are assigned in alphabetical order, but it's important to know the difference:

17.6.3 Sorting a data frame

The `sort()` function doesn't work properly with data frames. If you want to sort a data frame the standard advice that you'll find online is to use the `order()` function (not described in this book) to determine what order the rows should be sorted, and then use square brackets to do the shuffling. There's nothing inherently wrong with this advice, I just find it tedious. To that end, the `lsr` package includes a function called `sortFrame()` that you can use to do the sorting. The first argument to the function is named (`x`), and should correspond to the data frame that you want sorted. After that, all you do is type a list of the names of the variables that you want to use to do the sorting. For instance, if I type this:

```
sortFrame( garden, speaker, line)
```

```
##           speaker utterance line
## case.4 makka-pakka       pip    7
## case.5 makka-pakka       onk    9
## case.3   tombliboo        ee    5
## case.1  upsy-daisy       pip    1
## case.2  upsy-daisy       pip    2
```

what R does is first sort by `speaker` (factor level order). Any ties (i.e., data from the same speaker) are then sorted in order of `line` (increasing numerical order). You can use the minus sign to indicate that numerical variables should be sorted in reverse order:

```
sortFrame( garden, speaker, -line)
```

```
##           speaker utterance line
## case.5 makka-pakka       onk    9
## case.4 makka-pakka       pip    7
## case.3   tombliboo        ee    5
## case.2  upsy-daisy       pip    2
## case.1  upsy-daisy       pip    1
```

As of the current writing, the `sortFrame()` function is under development. I've started introducing functionality to allow you to use the `-` sign to non-numeric variables or to make a distinction between sorting factors alphabetically or by factor level. The idea is that you should be able to type in something like this:

```
sortFrame( garden, -speaker)
```

and have the output correspond to a sort of the `garden` data frame in *reverse* alphabetical order (or reverse factor level order) of `speaker`. As things stand right now, this will actually work, and it will produce sensible output:

```
sortFrame( garden, -speaker)
```

```
##           speaker utterance line
## case.1  upsy-daisy       pip    1
## case.2  upsy-daisy       pip    2
## case.3   tombliboo        ee    5
## case.4 makka-pakka       pip    7
## case.5 makka-pakka       onk    9
```

However, I'm not completely convinced that I've set this up in the ideal fashion, so this may change a little bit in the future.

17.6.4 Binding vectors together

A not-uncommon task that you might find yourself needing to undertake is to combine several vectors. For instance, let's suppose we have the following two numeric vectors:

```
cake.1 <- c(100, 80, 0, 0, 0)
cake.2 <- c(100, 100, 90, 30, 10)
```

The numbers here might represent the amount of each of the two cakes that are left at five different time points. Apparently the first cake is tastier, since that one gets devoured faster. We've already seen one method for combining these vectors: we could use the `data.frame()` function to convert them into a data frame with two variables, like so:

```
cake.df <- data.frame( cake.1, cake.2 )
cake.df
```

```
##   cake.1 cake.2
## 1    100    100
## 2     80    100
## 3      0     90
## 4      0     30
## 5      0     10
```

Two other methods that I want to briefly refer to are the `rbind()` and `cbind()` functions, which will convert the vectors into a matrix. I'll discuss matrices properly in Section 7.11.1 but the details don't matter too much for our current purposes. The `cbind()` function ("column bind") produces a very similar looking output to the data frame example:

```
cake.mat1 <- cbind( cake.1, cake.2 )
cake.mat1
```

```
##      cake.1 cake.2
## [1,]    100    100
## [2,]     80    100
## [3,]      0     90
## [4,]      0     30
## [5,]      0     10
```

but nevertheless it's important to keep in mind that `cake.mat1` is a matrix rather than a data frame, and so has a few differences from the `cake.df` variable. The `rbind()` function ("row bind") produces a somewhat different output: it binds the vectors together row-wise rather than column-wise, so the output now looks like this:

```
cake.mat2 <- rbind( cake.1, cake.2 )
cake.mat2
```

```
##        [,1] [,2] [,3] [,4] [,5]
## cake.1  100   80    0    0    0
## cake.2  100  100   90   30   10
```

You can add names to a matrix by using the `rownames()` and `colnames()` functions, and I should also point out that there's a fancier function in R called `merge()` that supports more complicated "database like" merging of vectors and data frames, but I won't go into details here.

17.6.5 Binding multiple copies of the same vector together

It is sometimes very useful to bind together multiple copies of the same vector. You could do this using the `rbind` and `cbind` functions, using comands like this one

```
fibonacci <- c( 1,1,2,3,5,8 )
rbind( fibonacci, fibonacci, fibonacci )
```

```
##           [,1] [,2] [,3] [,4] [,5] [,6]
## fibonacci    1    1    2    3    5    8
## fibonacci    1    1    2    3    5    8
## fibonacci    1    1    2    3    5    8
```

but that can be pretty annoying, especially if you needs lots of copies. To make this a little easier, the `lsr` package has two additional functions `rowCopy` and `colCopy` that do the same job, but all you have to do is specify the number of copies that you want, instead of typing the name in over and over again. The two arguments you need to specify are `x`, the vector to be copied, and `times`, indicating how many copies should be created:[117]

```
rowCopy( x = fibonacci, times = 3 )
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    1    1    2    3    5    8
## [2,]    1    1    2    3    5    8
## [3,]    1    1    2    3    5    8
```

Of course, in practice you don't need to name the arguments all the time. For instance, here's an example using the `colCopy()` function with the argument names omitted:

```
colCopy( fibonacci, 3 )
```

```
##      [,1] [,2] [,3]
## [1,]    1    1    1
## [2,]    1    1    1
## [3,]    2    2    2
## [4,]    3    3    3
## [5,]    5    5    5
## [6,]    8    8    8
```

17.6.6 Transposing a matrix or data frame

```
load("./rbook-master/data/cakes.Rdata" )
cakes
```

```
##        time.1 time.2 time.3 time.4 time.5
## cake.1    100     80      0      0      0
## cake.2    100    100     90     30     10
## cake.3    100     20     20     20     20
## cake.4    100    100    100    100    100
```

And just to make sure you believe me that this is actually a matrix:

```
class( cakes )
```

```
## [1] "matrix"
```

Okay, now let's transpose the matrix:

```
cakes.flipped <- t( cakes )
cakes.flipped
```

```
##        cake.1 cake.2 cake.3 cake.4
## time.1    100    100    100    100
## time.2     80    100     20    100
## time.3      0     90     20    100
## time.4      0     30     20    100
## time.5      0     10     20    100
```

The output here is still a matrix:

```
class( cakes.flipped )
```

```
## [1] "matrix"
```

At this point you should have two questions: (1) how do we do the same thing for data frames? and (2) why should we care about this? Let's start with the how question. First, I should note that you can transpose a data frame just fine using the `t()` function, but that has the slightly awkward consequence of converting the output from a data frame to a matrix, which isn't usually what you want. It's quite easy to convert the output back again, of course,[118] but I hate typing two commands when I can do it with one. To that end, the `lsr` package has a simple "convenience" function called `tFrame()` which does exactly the same thing as `t()` but converts the output to a data frame for you. To illustrate this, let's transpose the `itng` data frame that we used earlier. Here's the original data frame:

```
itng
```

```
##        speaker utterance
## 1   upsy-daisy       pip
## 2   upsy-daisy       pip
## 3   upsy-daisy       onk
## 4   upsy-daisy       onk
## 5    tombliboo        ee
## 6    tombliboo        oo
## 7  makka-pakka       pip
## 8  makka-pakka       pip
## 9  makka-pakka       onk
## 10 makka-pakka       onk
```

and here's what happens when you transpose it using `tFrame()`:

```
tFrame( itng )
```

```
##                  V1         V2         V3         V4         V5         V6
## speaker   upsy-daisy upsy-daisy upsy-daisy upsy-daisy tombliboo tombliboo
## utterance        pip        pip        onk        onk         ee         oo
##                   V7          V8          V9         V10
## speaker   makka-pakka makka-pakka makka-pakka makka-pakka
## utterance         pip         pip         onk         onk
```

An important point to recognise is that transposing a data frame is not always a sensible thing to do: in fact, I'd go so far as to argue that it's usually *not* sensible. It depends a lot on whether the "cases" from your original data frame would make sense as variables,

and to think of each of your original "variables" as cases. I think that's emphatically *not* true for our `itng` data frame, so I wouldn't advise doing it in this situation.

That being said, sometimes it really is true. For instance, had we originally stored our `cakes` variable as a data frame instead of a matrix, then it would absolutely be sensible to flip the data frame![119] There are some situations where it is useful to flip your data frame, so it's nice to know that you can do it. Indeed, that's the main reason why I have spent so much time talking about this topic. A lot of statistical tools make the assumption that the rows of your data frame (or matrix) correspond to observations, and the columns correspond to the variables. That's not unreasonable, of course, since that is a pretty standard convention. However, think about our `cakes` example here. This is a situation where you might want do an analysis of the different cakes (i.e. cakes as variables, time points as cases), but equally you might want to do an analysis where you think of the times as being the things of interest (i.e., times as variables, cakes as cases). If so, then it's useful to know how to flip a matrix or data frame around.

This page titled 17.6: Sorting, Flipping and Merging Data is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **7.6: Sorting, Flipping and Merging Data** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

17.7: Reshaping a Data Frame

One of the most annoying tasks that you need to undertake on a regular basis is that of reshaping a data frame. Framed in the most general way, reshaping the data means taking the data in whatever format it's given to you, and converting it to the format you need it. Of course, if we're going to characterise the problem that broadly, then about half of this chapter can probably be thought of as a kind of reshaping. So we're going to have to narrow things down a little bit. To that end, I'll talk about a few different tools that you can use for a few different tasks. In particular, I'll discuss a couple of easy to use (but limited) functions that I've included in the `lsr` package. In future versions of the book I plan to expand this discussion to include some of the more powerful tools that are available in R, but I haven't had the time to do so yet.

17.7.1 Long form and wide form data

The most common format in which you might obtain data is as a "case by variable" layout, commonly known as the ***wide form*** of the data.

```
load("./rbook-master/data/repeated.Rdata")
who()
```

```
##    -- Name --          -- Class --    -- Size --
##    age                 numeric        11
##    age.breaks          numeric        4
##    age.group           factor         11
##    age.group2          factor         11
##    age.group3          factor         11
##    age.labels          character      3
##    cake.1              numeric        5
##    cake.2              numeric        5
##    cake.df             data.frame     5 x 2
##    cake.mat1           matrix         5 x 2
##    cake.mat2           matrix         2 x 5
##    cakes               matrix         4 x 5
##    cakes.flipped       matrix         5 x 4
##    choice              data.frame     4 x 10
##    df                  data.frame     4 x 1
##    drugs               data.frame     10 x 8
##    fac                 factor         3
##    fibonacci           numeric        6
##    garden              data.frame     5 x 3
##    is.MP.speaking      logical        5
##    itng                data.frame     10 x 2
##    itng.table          table          3 x 4
##    likert.centred      numeric        10
##    likert.raw          numeric        10
##    makka.pakka         character      4
##    numbers             numeric        3
##    opinion.dir         numeric        10
##    opinion.strength    numeric        10
##    some.data           numeric        18
##    speaker             character      10
##    speech.by.char      list           3
##    text                character      3
##    tombliboo           character      2
##    upsy.daisy          character      4
##    utterance           character      10
```

To get a sense of what I'm talking about, consider an experiment in which we are interested in the different effects that alcohol and and caffeine have on people's working memory capacity (WMC) and reaction times (RT). We recruit 10 participants, and measure their WMC and RT under three different conditions: a "no drug" condition, in which they are not under the influence of either caffeine or alcohol, a "caffeine" condition, in which they are under the inflence of caffeine, and an "alcohol" condition, in which… well, you can probably guess. Ideally, I suppose, there would be a fourth condition in which both drugs are administered, but for the sake of simplicity let's ignore that. The `drugs` data frame gives you a sense of what kind of data you might observe in an experiment like this:

```
drugs
```

```
##    id gender WMC_alcohol WMC_caffeine WMC_no.drug RT_alcohol RT_caffeine
## 1   1 female         3.7          3.7         3.9        488         236
## 2   2 female         6.4          7.3         7.9        607         376
## 3   3 female         4.6          7.4         7.3        643         226
## 4   4   male         6.4          7.8         8.2        684         206
## 5   5 female         4.9          5.2         7.0        593         262
## 6   6   male         5.4          6.6         7.2        492         230
## 7   7   male         7.9          7.9         8.9        690         259
## 8   8   male         4.1          5.9         4.5        486         230
## 9   9 female         5.2          6.2         7.2        686         273
## 10 10 female         6.2          7.4         7.8        645         240
##    RT_no.drug
## 1         371
## 2         349
## 3         412
## 4         252
## 5         439
## 6         464
## 7         327
## 8         305
## 9         327
## 10        498
```

This is a data set in "wide form", in which each participant corresponds to a single row. We have two variables that are characteristics of the subject (i.e., their `id` number and their `gender`) and six variables that refer to one of the two measured variables (WMC or RT) in one of the three testing conditions (alcohol, caffeine or no drug). Because all of the testing conditions (i.e., the three drug types) are applied to all participants, drug type is an example of a ***within-subject factor***.

17.7.2 Reshaping data using `wideToLong()`

The "wide form" of this data set is useful for some situations: it is often very useful to have each row correspond to a single subject. However, it is not the only way in which you might want to organise this data. For instance, you might want to have a separate row for each "testing occasion". That is, "participant 1 under the influence of alcohol" would be one row, and "participant 1 under the influence of caffeine" would be another row. This way of organising the data is generally referred to as the ***long form*** of the data. It's not too difficult to switch between wide and long form, and I'll explain how it works in a moment; for now, let's just have a look at what the long form of this data set looks like:

```
drugs.2 <- wideToLong( data = drugs, within = "drug" )
head(drugs.2)
```

```
##   id gender    drug WMC  RT
## 1  1 female alcohol 3.7 488
## 2  2 female alcohol 6.4 607
## 3  3 female alcohol 4.6 643
## 4  4   male alcohol 6.4 684
## 5  5 female alcohol 4.9 593
## 6  6   male alcohol 5.4 492
```

The `drugs.2` data frame that we just created has 30 rows: each of the 10 participants appears in three separate rows, one corresponding to each of the three testing conditions. And instead of having a variable like `WMC_caffeine` that indicates that

we were measuring "WMC" in the "caffeine" condition, this information is now recorded in two separate variables, one called `drug` and another called `WMC`. Obviously, the long and wide forms of the data contain the same information, but they represent quite different ways of organising that information. Sometimes you find yourself needing to analyse data in wide form, and sometimes you find that you need long form. So it's really useful to know how to switch between the two.

In the example I gave above, I used a function called `wideToLong()` to do the transformation. The `wideToLong()` function is part of the `lsr` package. The key to understanding this function is that it relies on the *variable names* to do all the work. Notice that the variable names in the `drugs` data frame follow a very clear scheme. Whenever you have a variable with a name like `WMC_caffeine` you know that the variable being measured is "WMC", and that the specific condition in which it is being measured is the "caffeine" condition. Similarly, you know that `RT_no.drug` refers to the "RT" variable measured in the "no drug" condition. The measured variable comes first (e.g., `WMC`), followed by a separator character (in this case the separator is an underscore, `_`), and then the name of the condition in which it is being measured (e.g., `caffeine`). There are two different prefixes (i.e, the strings before the separator, `WMC`, `RT`) which means that there are two separate variables being measured. There are three different suffixes (i.e., the strings after the separtator, `caffeine`, `alcohol`, `no.drug`) meaning that there are three different levels of the within-subject factor. Finally, notice that the separator string (i.e., `_`) does not appear anywhere in two of the variables (`id`, `gender`), indicating that these are ***between-subject*** variables, namely variables that do not vary within participant (e.g., a person's `gender` is the same regardless of whether they're under the influence of alcohol, caffeine etc).

Because of the fact that the variable naming scheme here is so informative, it's quite possible to reshape the data frame without any additional input from the user. For example, in this particular case, you could just type the following:

```
wideToLong( drugs )
```

```
##    id gender   within WMC  RT
## 1   1 female  alcohol 3.7 488
## 2   2 female  alcohol 6.4 607
## 3   3 female  alcohol 4.6 643
## 4   4   male  alcohol 6.4 684
## 5   5 female  alcohol 4.9 593
## 6   6   male  alcohol 5.4 492
## 7   7   male  alcohol 7.9 690
## 8   8   male  alcohol 4.1 486
## 9   9 female  alcohol 5.2 686
## 10 10 female  alcohol 6.2 645
## 11  1 female caffeine 3.7 236
## 12  2 female caffeine 7.3 376
## 13  3 female caffeine 7.4 226
## 14  4   male caffeine 7.8 206
## 15  5 female caffeine 5.2 262
## 16  6   male caffeine 6.6 230
## 17  7   male caffeine 7.9 259
## 18  8   male caffeine 5.9 230
## 19  9 female caffeine 6.2 273
## 20 10 female caffeine 7.4 240
## 21  1 female  no.drug 3.9 371
## 22  2 female  no.drug 7.9 349
## 23  3 female  no.drug 7.3 412
## 24  4   male  no.drug 8.2 252
## 25  5 female  no.drug 7.0 439
## 26  6   male  no.drug 7.2 464
## 27  7   male  no.drug 8.9 327
## 28  8   male  no.drug 4.5 305
## 29  9 female  no.drug 7.2 327
## 30 10 female  no.drug 7.8 498
```

This is pretty good, actually. The only think it has gotten wrong here is that it doesn't know what name to assign to the within-subject factor, so instaed of calling it something sensible like `drug`, it has use the unimaginative name `within`. If you want to ensure that the `wideToLong()` function applies a sensible name, you have to specify the `within` argument, which is just a character string that specifies the name of the within-subject factor. So when I used this command earlier,

```
drugs.2 <- wideToLong( data = drugs, within = "drug" )
```

all I was doing was telling R to use `drug` as the name of the within subject factor.

Now, as I was hinting earlier, the `wideToLong()` function is very inflexible. It *requires* that the variable names all follow this naming scheme that I outlined earlier. If you don't follow this naming scheme it won't work.[120] The only flexibility that I've included here is that you can change the separator character by specifying the `sep` argument. For instance, if you were using variable names of the form `WMC/caffeine`, for instance, you could specify that `sep="/"`, using a command like this

```
drugs.2 <- wideToLong( data = drugs, within = "drug", sep = "/" )
```

and it would still work.

17.7.3 Reshaping data using `longToWide()`

To convert data from long form to wide form, the `lsr` package also includes a function called `longToWide()`. Recall from earlier that the long form of the data (i.e., the `drugs.2` data frame) contains variables named `id`, `gender`, `drug`, `WMC` and `RT`. In order to convert from long form to wide form, all you need to do is indicate which of these variables are measured separately for each condition (i.e., `WMC` and `RT`), and which variable is the within-subject factor that specifies the condition (i.e., `drug`). You do this via a two-sided formula, in which the measured variables are on the left hand side, and the within-subject factor is on the ritght hand side. In this case, the formula would be `WMC + RT ~ drug`. So the command that we would use might look like this:

```
longToWide( data=drugs.2, formula= WMC+RT ~ drug )
```

```
##    id gender WMC_alcohol RT_alcohol WMC_caffeine RT_caffeine WMC_no.drug
## 1   1 female         3.7        488          3.7         236         3.9
## 2   2 female         6.4        607          7.3         376         7.9
## 3   3 female         4.6        643          7.4         226         7.3
## 4   4   male         6.4        684          7.8         206         8.2
## 5   5 female         4.9        593          5.2         262         7.0
## 6   6   male         5.4        492          6.6         230         7.2
## 7   7   male         7.9        690          7.9         259         8.9
## 8   8   male         4.1        486          5.9         230         4.5
## 9   9 female         5.2        686          6.2         273         7.2
## 10 10 female         6.2        645          7.4         240         7.8
##    RT_no.drug
## 1         371
## 2         349
## 3         412
## 4         252
## 5         439
## 6         464
## 7         327
## 8         305
## 9         327
## 10        498
```

or, if we chose to omit argument names, we could simplify it to this:

```
longToWide( drugs.2, WMC+RT ~ drug )
```

```
##    id gender WMC_alcohol RT_alcohol WMC_caffeine RT_caffeine WMC_no.drug
## 1   1 female         3.7        488          3.7         236         3.9
## 2   2 female         6.4        607          7.3         376         7.9
## 3   3 female         4.6        643          7.4         226         7.3
## 4   4   male         6.4        684          7.8         206         8.2
## 5   5 female         4.9        593          5.2         262         7.0
## 6   6   male         5.4        492          6.6         230         7.2
## 7   7   male         7.9        690          7.9         259         8.9
## 8   8   male         4.1        486          5.9         230         4.5
## 9   9 female         5.2        686          6.2         273         7.2
## 10 10 female         6.2        645          7.4         240         7.8
##    RT_no.drug
## 1         371
## 2         349
## 3         412
## 4         252
## 5         439
## 6         464
## 7         327
## 8         305
## 9         327
## 10        498
```

Note that, just like the `wideToLong()` function, the `longToWide()` function allows you to override the default separator character. For instance, if the command I used had been

```
longToWide( drugs.2, WMC+RT ~ drug, sep="/" )
```

```
##    id gender WMC/alcohol RT/alcohol WMC/caffeine RT/caffeine WMC/no.drug
## 1   1 female         3.7        488          3.7         236         3.9
## 2   2 female         6.4        607          7.3         376         7.9
## 3   3 female         4.6        643          7.4         226         7.3
## 4   4   male         6.4        684          7.8         206         8.2
## 5   5 female         4.9        593          5.2         262         7.0
## 6   6   male         5.4        492          6.6         230         7.2
## 7   7   male         7.9        690          7.9         259         8.9
## 8   8   male         4.1        486          5.9         230         4.5
## 9   9 female         5.2        686          6.2         273         7.2
## 10 10 female         6.2        645          7.4         240         7.8
##    RT/no.drug
## 1         371
## 2         349
## 3         412
## 4         252
## 5         439
## 6         464
## 7         327
## 8         305
## 9         327
## 10        498
```

the output would contain variables with names like `RT/alcohol` instead of `RT_alcohol` .

17.7.4 Reshaping with multiple within-subject factors

As I mentioned above, the `wideToLong()` and `longToWide()` functions are quite limited in terms of what they can do. However, they do handle a broader range of situations than the one outlined above. Consider the following, fairly simple psychological experiment. I'm interested in the effects of practice on some simple decision making problem. It doesn't really matter what the problem is, other than to note that I'm interested in two distinct outcome variables. Firstly, I care about people's accuracy, measured by the proportion of decisions that people make correctly, denoted PC. Secondly, I care about people's speed, measured by the mean response time taken to make those decisions, denoted MRT. That's standard in psychological experiments: the speed-accuracy trade-off is pretty ubiquitous, so we generally need to care about both variables.

To look at the effects of practice over the long term, I test each participant on two days, `day1` and `day2` , where for the sake of argument I'll assume that `day1` and `day2` are about a week apart. To look at the effects of practice over the short term, the testing during each day is broken into two "blocks", `block1` and `block2` , which are about 20 minutes apart. This isn't the world's most complicated experiment, but it's still a fair bit more complicated than the last one. This time around we have two within-subject factors (i.e., `day` and `block`) and we have two measured variables for each condition (i.e., `PC` and `MRT`). The `choice` data frame shows what the wide form of this kind of data might look like:

```
choice
```

```
##   id gender MRT/block1/day1 MRT/block1/day2 MRT/block2/day1
## 1  1   male             415             400             455
## 2  2   male             500             490             532
## 3  3 female             478             468             499
## 4  4 female             550             502             602
##   MRT/block2/day2 PC/block1/day1 PC/block1/day2 PC/block2/day1
## 1             450             79             88             82
## 2             518             83             92             86
## 3             474             91             98             90
## 4             588             75             89             78
##   PC/block2/day2
## 1             93
## 2             97
## 3            100
## 4             95
```

Notice that this time around we have variable names of the form `MRT/block1/day2`. As before, the first part of the name refers to the measured variable (response time), but there are now two suffixes, one indicating that the testing took place in block 1, and the other indicating that it took place on day 2. And just to complicate matters, it uses `/` as the separator character rather than `_`. Even so, reshaping this data set is pretty easy. The command to do it is,

```
choice.2 <- wideToLong( choice, within=c("block","day"), sep="/" )
```

which is pretty much the exact same command we used last time. The only difference here is that, because there are two within-subject factors, the `within` argument is a vector that contains two names. When we look at the long form data frame that this creates, we get this:

```
choice.2
```

```
##    id gender MRT  PC  block  day
## 1   1   male 415  79 block1 day1
## 2   2   male 500  83 block1 day1
## 3   3 female 478  91 block1 day1
## 4   4 female 550  75 block1 day1
## 5   1   male 400  88 block1 day2
## 6   2   male 490  92 block1 day2
## 7   3 female 468  98 block1 day2
## 8   4 female 502  89 block1 day2
## 9   1   male 455  82 block2 day1
## 10  2   male 532  86 block2 day1
## 11  3 female 499  90 block2 day1
## 12  4 female 602  78 block2 day1
## 13  1   male 450  93 block2 day2
## 14  2   male 518  97 block2 day2
## 15  3 female 474 100 block2 day2
## 16  4 female 588  95 block2 day2
```

In this long form data frame we have two between-subject variables (`id` and `gender`), two variables that define our within-subject manipulations (`block` and `day`), and two more contain the measurements we took (`MRT` and `PC`).

To convert this back to wide form is equally straightforward. We use the `longToWide()` function, but this time around we need to alter the formula in order to tell it that we have two within-subject factors. The command is now

```
longToWide( choice.2, MRT+PC ~ block+day, sep="/" )
```

```
##   id gender MRT/block1/day1 PC/block1/day1 MRT/block1/day2 PC/block1/day2
## 1  1   male             415             79             400             88
## 2  2   male             500             83             490             92
## 3  3 female             478             91             468             98
## 4  4 female             550             75             502             89
##   MRT/block2/day1 PC/block2/day1 MRT/block2/day2 PC/block2/day2
## 1             455             82             450             93
## 2             532             86             518             97
## 3             499             90             474            100
## 4             602             78             588             95
```

and this produces a wide form data set containing the same variables as the original `choice` data frame.

17.7.5 What other options are there?

The advantage to the approach described in the previous section is that it solves a quite specific problem (but a commonly encountered one) with a minimum of fuss. The disadvantage is that the tools are quite limited in scope. They allow you to switch your data back and forth between two different formats that are very common in everyday data analysis. However, there a number of other tools that you can use if need be. Just within the core packages distributed with R there is the `reshape()` function, as well as the `stack()` and `unstack()` functions, all of which can be useful under certain circumstances. And there are of course thousands of packages on CRAN that you can use to help you with different tasks. One popular package for this purpose is the `reshape` package, written by Hadley Wickham (**???** for details see Wickham2007). There are two key functions in this package, called `melt()` and `cast()` that are pretty useful for solving a lot of reshaping problems. In a future version of this book I intend to discuss `melt()` and `cast()` in a fair amount of detail.

This page titled 17.7: Reshaping a Data Frame is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **7.7: Reshaping a Data Frame** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

17.8: Working with Text

Sometimes your data set is quite text heavy. This can be for a lot of different reasons. Maybe the raw data are actually taken from text sources (e.g., newspaper articles), or maybe your data set contains a lot of free responses to survey questions, in which people can write whatever text they like in response to some query. Or maybe you just need to rejig some of the text used to describe nominal scale variables. Regardless of what the reason is, you'll probably want to know a little bit about how to handle text in R. Some things you already know how to do: I've discussed the use of `nchar()` to calculate the number of characters in a string (Section 3.8.1), and a lot of the general purpose tools that I've discussed elsewhere (e.g., the `==` operator) have been applied to text data as well as to numeric data. However, because text data is quite rich, and generally not as well structured as numeric data, R provides a lot of additional tools that are quite specific to text. In this section I discuss only those tools that come as part of the base packages, but there are other possibilities out there: the `stringr` package provides a powerful alternative that is a lot more coherent than the basic tools, and is well worth looking into.

17.8.1 Shortening a string

The first task I want to talk about is how to shorten a character string. For example, suppose that I have a vector that contains the names of several different animals:

```
animals <- c( "cat", "dog", "kangaroo", "whale" )
```

It might be useful in some contexts to extract the first three letters of each word. This is often useful when annotating figures, or when creating variable labels: it's often very inconvenient to use the full name, so you want to shorten it to a short code for space reasons. The `strtrim()` function can be used for this purpose. It has two arguments: `x` is a vector containing the text to be shortened and `width` specifies the number of characters to keep. When applied to the `animals` data, here's what we get:

```
strtrim( x = animals, width = 3 )
```

```
## [1] "cat" "dog" "kan" "wha"
```

Note that the only thing that `strtrim()` does is chop off excess characters at the end of a string. It doesn't insert any whitespace characters to fill them out if the original string is shorter than the `width` argument. For example, if I trim the `animals` data to 4 characters, here's what I get:

```
strtrim( x = animals, width = 4 )
```

```
## [1] "cat"  "dog"  "kang" "whal"
```

The `"cat"` and `"dog"` strings still only use 3 characters. Okay, but what if you don't want to start from the first letter? Suppose, for instance, I only wanted to keep the second and third letter of each word. That doesn't happen quite as often, but there are some situations where you need to do something like that. If that does happen, then the function you need is `substr()`, in which you specify a `start` point and a `stop` point instead of specifying the width. For instance, to keep only the 2nd and 3rd letters of the various `animals`, I can do the following:

```
substr( x = animals, start = 2, stop = 3 )
```

```
## [1] "at" "og" "an" "ha"
```

17.8.2 Pasting strings together

Much more commonly, you will need either to glue several character strings together or to pull them apart. To glue several strings together, the `paste()` function is very useful. There are three arguments to the `paste()` function:

- `...` As usual, the dots "match" up against any number of inputs. In this case, the inputs should be the various different strings you want to paste together.
- `sep` . This argument should be a string, indicating what characters R should use as separators, in order to keep each of the original strings separate from each other in the pasted output. By default the value is a single space, `sep = " "` . This is made a little clearer when we look at the examples.
- `collapse` . This is an argument indicating whether the `paste()` function should interpret vector inputs as things to be collapsed, or whether a vector of inputs should be converted into a vector of outputs. The default value is `collapse = NULL` which is interpreted as meaning that vectors should not be collapsed. If you want to collapse vectors into as single string, then you should specify a value for `collapse` . Specifically, the value of `collapse` should correspond to the separator character that you want to use for the collapsed inputs. Again, see the examples below for more details.

That probably doesn't make much sense yet, so let's start with a simple example. First, let's try to paste two words together, like this:

```
paste( "hello", "world" )
```

```
## [1] "hello world"
```

Notice that R has inserted a space between the `"hello"` and `"world"` . Suppose that's not what I wanted. Instead, I might want to use `.` as the separator character, or to use no separator at all. To do either of those, I would need to specify `sep = "."` or `sep = ""` .[121] For instance:

```
paste( "hello", "world", sep = "." )
```

```
## [1] "hello.world"
```

Now let's consider a slightly more complicated example. Suppose I have two vectors that I want to `paste()` together. Let's say something like this:

```
hw <- c( "hello", "world" )
ng <- c( "nasty", "government" )
```

And suppose I want to paste these together. However, if you think about it, this statement is kind of ambiguous. It could mean that I want to do an "element wise" paste, in which all of the first elements get pasted together (`"hello nasty"`) and all the second elements get pasted together (`"world government"`). Or, alternatively, I might intend to collapse everything into one big string (`"hello nasty world government"`). By default, the `paste()` function assumes that you want to do an element-wise paste:

```
paste( hw, ng )
```

```
## [1] "hello nasty"      "world government"
```

However, there's nothing stopping you from overriding this default. All you have to do is specify a value for the `collapse` argument, and R will chuck everything into one dirty big string. To give you a sense of exactly how this works, what I'll do in this

next example is specify *different* values for `sep` and `collapse` :

```
paste( hw, ng, sep = ".", collapse = ":::")
```

```
## [1] "hello.nasty:::world.government"
```

17.8.3 Splitting strings

At other times you have the opposite problem to the one in the last section: you have a whole lot of text bundled together into a single string that needs to be pulled apart and stored as several different variables. For instance, the data set that you get sent might include a single variable containing someone's full name, and you need to separate it into first names and last names. To do this in R you can use the `strsplit()` function, and for the sake of argument, let's assume that the string you want to split up is the following string:

```
monkey <- "It was the best of times. It was the blurst of times."
```

To use the `strsplit()` function to break this apart, there are three arguments that you need to pay particular attention to:

- `x` . A vector of character strings containing the data that you want to split.
- `split` . Depending on the value of the `fixed` argument, this is either a fixed string that specifies a delimiter, or a regular expression that matches against one or more possible delimiters. If you don't know what regular expressions are (probably most readers of this book), don't use this option. Just specify a separator string, just like you would for the `paste()` function.
- `fixed` . Set `fixed = TRUE` if you want to use a fixed delimiter. As noted above, unless you understand regular expressions this is definitely what you want. However, the default value is `fixed = FALSE` , so you have to set it explicitly.

Let's look at a simple example:

```
monkey.1 <- strsplit( x = monkey, split = " ", fixed = TRUE )
monkey.1
```

```
## [[1]]
##  [1] "It"     "was"    "the"    "best"   "of"     "times." "It"
##  [8] "was"    "the"    "blurst" "of"     "times."
```

One thing to note in passing is that the output here is a list (you can tell from the part of the output), whose first and only element is a character vector. This is useful in a lot of ways, since it means that you can input a character vector for `x` and then then have the `strsplit()` function split all of them, but it's kind of annoying when you only have a single input. To that end, it's useful to know that you can `unlist()` the output:

```
unlist( monkey.1 )
```

```
##  [1] "It"     "was"    "the"    "best"   "of"     "times." "It"
##  [8] "was"    "the"    "blurst" "of"     "times."
```

To understand why it's important to remember to use the `fixed = TRUE` argument, suppose we wanted to split this into two separate sentences. That is, we want to use `split = "."` as our delimiter string. As long as we tell R to remember to treat this as a *fixed* separator character, then we get the right answer:

```
strsplit( x = monkey, split = ".", fixed = TRUE )
```

```
## [[1]]
## [1] "It was the best of times"    " It was the blurst of times"
```

However, if we don't do this, then R will assume that when you typed `split = "."` you were trying to construct a "regular expression", and as it happens the character `.` has a special meaning within a regular expression. As a consequence, if you forget to include the `fixed = TRUE` part, you won't get the answers you're looking for.

17.8.4 Making simple conversions

A slightly different task that comes up quite often is making transformations to text. A simple example of this would be converting text to lower case or upper case, which you can do using the `toupper()` and `tolower()` functions. Both of these functions have a single argument `x` which contains the text that needs to be converted. An example of this is shown below:

```
text <- c( "lIfe", "Impact" )
tolower( x = text )
```

```
## [1] "life"   "impact"
```

A slightly more powerful way of doing text transformations is to use the `chartr()` function, which allows you to specify a "character by character" substitution. This function contains three arguments, `old`, `new` and `x`. As usual `x` specifies the text that needs to be transformed. The `old` and `new` arguments are strings of the same length, and they specify how `x` is to be converted. Every instance of the first character in `old` is converted to the first character in `new` and so on. For instance, suppose I wanted to convert `"albino"` to `"libido"`. To do this, I need to convert all of the `"a"` characters (all 1 of them) in `"albino"` into `"l"` characters (i.e., `a` → `l`). Additionally, I need to make the substitutions `l` → `i` and `n` → `d`. To do so, I would use the following command:

```
old.text <- "albino"
chartr( old = "aln", new = "lid", x = old.text )
```

```
## [1] "libido"
```

17.8.5 Applying logical operations to text

In Section 3.9.5 we discussed a very basic text processing tool, namely the ability to use the equality operator `==` to test to see if two strings are identical to each other. However, you can also use other logical operators too. For instance R also allows you to use the `<` and `>` operators to determine which of two strings comes first, alphabetically speaking. Sort of. Actually, it's a bit more complicated than that, but let's start with a simple example:

```
"cat" < "dog"
```

```
## [1] TRUE
```

In this case, we see that `"cat"` does does come before `"dog"` alphabetically, so R judges the statement to be true. However, if we ask R to tell us if `"cat"` comes before `"anteater"`,

```
"cat" < "anteater"
```

```
## [1] FALSE
```

It tell us that the statement is false. So far, so good. But text data is a bit more complicated than the dictionary suggests. What about `"cat"` and `"CAT"` ? Which of these comes first? Let's try it and find out:

```
"CAT" < "cat"
```

```
## [1] FALSE
```

In other words, R assumes that uppercase letters come before lowercase ones. Fair enough. No-one is likely to be surprised by that. What you might find surprising is that R assumes that *all* uppercase letters come before *all* lowercase ones. That is, while `"anteater" < "zebra"` is a true statement, and the uppercase equivalent `"ANTEATER" < "ZEBRA"` is also true, it is *not* true to say that `"anteater" < "ZEBRA"` , as the following extract illustrates:

```
"anteater" < "ZEBRA"
```

```
## [1] TRUE
```

This may seem slightly counterintuitive. With that in mind, it may help to have a quick look Table 7.3, which lists various text characters in the order that R uses.

Table 7.3: The ordering of various text characters used by the < and > operators, as well as by the sort() function. Not shown is the "space" character, which actually comes rst on the list.

Characters
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ ] ^ _ ‘ a b c d e f g h i j k l m n o p q r s t u v w x y z } | {

One function that I want to make a point of talking about, even though it's not quite on topic, is the `cat()` function. The `cat()` function is a of mixture of `paste()` and `print()` . That is, what it does is concatenate strings and then print them out. In your own work you can probably survive without it, since `print()` and `paste()` will actually do what you need, but the `cat()` function is so widely used that I think it's a good idea to talk about it here. The basic idea behind `cat()` is straightforward. Like `paste()` , it takes several arguments as inputs, which it converts to strings, collapses (using a separator character specified using the `sep` argument), and prints on screen. If you want, you can use the `file` argument to tell R to print the output into a file rather than on screen (I won't do that here). However, it's important to note that the `cat()` function collapses vectors first, and *then* concatenates them. That is, notice that when I use `cat()` to combine `hw` and `ng` , I get a different result than if I'd used `paste()`

```
cat( hw, ng )
```

```
## hello world nasty government
```

```
paste( hw, ng, collapse = " " )
```

```
## [1] "hello nasty world government"
```

Notice the difference in the ordering of words. There's a few additional details that I need to mention about `cat()` . Firstly, `cat()` really is a function for *printing*, and not for creating text strings to store for later. You can't assign the output to a variable, as the following example illustrates:

```
x <- cat( hw, ng )
```

```
## hello world nasty government
```

```
x
```

```
## NULL
```

Despite my attempt to store the output as a variable, `cat()` printed the results on screen anyway, and it turns out that the variable I created doesn't contain anything at all.[122] Secondly, the `cat()` function makes use of a number of "special" characters. I'll talk more about these in the next section, but I'll illustrate the basic point now, using the example of `"\n"` which is interpreted as a "new line" character. For instance, compare the behaviour of `print()` and `cat()` when asked to print the string `"hello\nworld"` :

```
print( "hello\nworld" )  # print literally:
```

```
## [1] "hello\nworld"
```

```
cat( "hello\nworld" )  # interpret as newline
```

```
## hello
## world
```

In fact, this behaviour is important enough that it deserves a section of its very own…

17.8.6 Using escape characters in text

The previous section brings us quite naturally to a fairly fundamental issue when dealing with strings, namely the issue of delimiters and escape characters. Reduced to its most basic form, the problem we have is that R commands are written using text characters, and our strings also consist of text characters. So, suppose I want to type in the word "hello", and have R encode it as a string. If I were to just type `hello` , R will think that I'm referring to a variable or a function called `hello` rather than interpret it as a string. The solution that R adopts is to require you to enclose your string by ***delimiter*** characters, which can be either double quotes or single quotes. So, when I type `"hello"` or `'hello'` then R knows that it should treat the text in between the quote marks as a character string. However, this isn't a complete solution to the problem: after all, `"` and `'` are themselves perfectly legitimate text characters, and so we might want to include those in our string as well. For instance, suppose I wanted to encode the name "O'Rourke" as a string. It's *not* legitimate for me to type `'O'rourke'` because R is too stupid to realise that "O'Rourke" is a real word. So it will interpret the `'O'` part as a complete string, and then will get confused when it reaches the `Rourke'` part. As a consequence, what you get is an error message:

```
'O'Rourke'
Error: unexpected symbol in "'O'Rourke"
```

To some extent, R offers us a cheap fix to the problem because of the fact that it allows us to use either `"` or `'` as the delimiter character. Although `'O'rourke'` will make R cry, it is perfectly happy with `"O'Rourke"` :

```
"O'Rourke"
```

```
## [1] "O'Rourke"
```

This is a real advantage to having two different delimiter characters. Unfortunately, anyone with even the slightest bit of deviousness to them can see the problem with this. Suppose I'm reading a book that contains the following passage,

P.J. O'Rourke says, "Yay, money!". It's a joke, but no-one laughs.

and I want to enter this as a string. Neither the `'` or `"` delimiters will solve the problem here, since this string contains both a single quote character and a double quote character. To encode strings like this one, we have to do something a little bit clever.

Table 7.4: Standard escape characters that are evaluated by some text processing commands, including `cat()`. This convention dates back to the development of the C programming language in the 1970s, and as a consequence a lot of these characters make most sense if you pretend that R is actually a typewriter, as explained in the main text. Type `?Quotes` for the corresponding R help file.

Escape.sequence	Interpretation
`\n`	Newline
`\t`	Horizontal Tab
`\v`	Vertical Tab
`\b`	Backspace
`\r`	Carriage Return
`\f`	Form feed
`\a`	Alert sound
`\\`	Backslash
`\'`	Single quote
`\"`	Double quote

The solution to the problem is to designate an ***escape character***, which in this case is `\`, the humble backslash. The escape character is a bit of a sacrificial lamb: if you include a backslash character in your string, R will *not* treat it as a literal character at all. It's actually used as a way of inserting "special" characters into your string. For instance, if you want to force R to insert actual quote marks into the string, then what you actually type is `\'` or `\"` (these are called ***escape sequences***). So, in order to encode the string discussed earlier, here's a command I could use:

```
PJ <- "P.J. O\'Rourke says, \"Yay, money!\". It\'s a joke, but no-one laughs."
```

Notice that I've included the backslashes for both the single quotes and double quotes. That's actually overkill: since I've used `"` as my delimiter, I only needed to do this for the double quotes. Nevertheless, the command has worked, since I didn't get an error message. Now let's see what happens when I print it out:

```
print( PJ )
```

```
## [1] "P.J. O'Rourke says, \"Yay, money!\". It's a joke, but no-one laughs."
```

Hm. Why has R printed out the string using `\"`? For the exact same reason that *I* needed to insert the backslash in the first place. That is, when R prints out the `PJ` string, it has enclosed it with delimiter characters, and it wants to unambiguously show us which of the double quotes are delimiters and which ones are actually part of the string. Fortunately, if this bugs you, you can make

it go away by using the `print.noquote()` function, which will just print out the literal string that you encoded in the first place:

```
print.noquote( PJ )
```

Typing `cat(PJ)` will produce a similar output.

Introducing the escape character solves a lot of problems, since it provides a mechanism by which we can insert all sorts of characters that aren't on the keyboard. For instance, as far as a computer is concerned, "new line" is actually a text character. It's the character that is printed whenever you hit the "return" key on your keyboard. If you want to insert a new line character into your string, you can actually do this by including the escape sequence `\n`. Or, if you want to insert a backslash character, then you can use `\\`. A list of the standard escape sequences recognised by R is shown in Table 7.4. A lot of these actually date back to the days of the typewriter (e.g., carriage return), so they might seem a bit counterintuitive to people who've never used one. In order to get a sense for what the various escape sequences do, we'll have to use the `cat()` function, because it's the only function "dumb" enough to literally print them out:

```
cat( "xxxx\boo" )  # \b is a backspace, so it deletes the preceding x
cat( "xxxx\too" )  # \t is a tab, so it inserts a tab space
cat( "xxxx\noo" )  # \n is a newline character
cat( "xxxx\roo" )  # \r returns you to the beginning of the line
```

And that's pretty much it. There are a few other escape sequence that R recognises, which you can use to insert arbitrary ASCII or Unicode characters into your string (type `?Quotes` for more details) but I won't go into details here.

17.8.7 Matching and substituting text

Another task that we often want to solve is find all strings that match a certain criterion, and possibly even to make alterations to the text on that basis. There are several functions in R that allow you to do this, three of which I'll talk about briefly here: `grep()`, `gsub()` and `sub()`. Much like the `substr()` function that I talked about earlier, all three of these functions are intended to be used in conjunction with regular expressions (see Section 7.8.9 but you can also use them in a simpler fashion, since they all allow you to set `fixed = TRUE`, which means we can ignore all this regular expression rubbish and just use simple text matching.

So, how do these functions work? Let's start with the `grep()` function. The purpose of this function is to input a vector of character strings `x`, and to extract all those strings that fit a certain pattern. In our examples, I'll assume that the `pattern` in question is a literal sequence of characters that the string must contain (that's what `fixed = TRUE` does). To illustrate this, let's start with a simple data set, a vector that contains the names of three `beers`. Something like this:

```
beers <- c( "little creatures", "sierra nevada", "coopers pale" )
```

Next, let's use `grep()` to find out which of these strings contains the substring `"er"`. That is, the `pattern` that we need to match is the fixed string `"er"`, so the command we need to use is:

```
grep( pattern = "er", x = beers, fixed = TRUE )
```

```
## [1] 2 3
```

What the output here is telling us is that the second and third elements of `beers` both contain the substring `"er"`. Alternatively, however, we might prefer it if `grep()` returned the actual strings themselves. We can do this by specifying `value = TRUE` in our function call. That is, we'd use a command like this:

```
grep( pattern = "er", x = beers, fixed = TRUE, value = TRUE )
```

```
## [1] "sierra nevada" "coopers pale"
```

The other two functions that I wanted to mention in this section are `gsub()` and `sub()`. These are both similar in spirit to `grep()` insofar as what they do is search through the input strings (`x`) and find all of the strings that match a `pattern`. However, what these two functions do is *replace* the pattern with a `replacement` string. The `gsub()` function will replace *all* instances of the pattern, whereas the `sub()` function just replaces the first instance of it in each string. To illustrate how this works, suppose I want to replace all instances of the letter `"a"` with the string `"BLAH"`. I can do this to the `beers` data using the `gsub()` function:

```
gsub( pattern = "a", replacement = "BLAH", x = beers, fixed = TRUE )
```

```
## [1] "little creBLAHtures"    "sierrBLAH nevBLAHdBLAH"
## [3] "coopers pBLAHle"
```

Notice that all three of the `"a"` s in `"sierra nevada"` have been replaced. In contrast, let's see what happens when we use the exact same command, but this time using the `sub()` function instead:

```
sub( pattern = "a", replacement = "BLAH", x = beers, fixed = TRUE )
```

```
## [1] "little creBLAHtures" "sierrBLAH nevada"    "coopers pBLAHle"
```

Only the first `"a"` is changed.

17.8.8 Regular expressions (not really)

There's one last thing I want to talk about regarding text manipulation, and that's the concept of a ***regular expression***. Throughout this section we've often needed to specify `fixed = TRUE` in order to force R to treat some of our strings as actual strings, rather than as regular expressions. So, before moving on, I want to very briefly explain what regular expressions are. I'm *not* going to talk at all about how they work or how you specify them, because they're genuinely complicated and not at all relevant to this book. However, they are extremely powerful tools and they're quite widely used by people who have to work with lots of text data (e.g., people who work with natural language data), and so it's handy to at least have a vague idea about what they are. The basic idea is quite simple. Suppose I want to extract all strings in my `beers` vector that contain a vowel followed immediately by the letter `"s"`. That is, I want to finds the beer names that contain either `"as"`, `"es"`, `"is"`, `"os"` or `"us"`. One possibility would be to manually specify all of these possibilities and then match against these as fixed strings one at a time, but that's tedious. The alternative is to try to write out a single "regular" expression that matches all of these. The regular expression that does this[123] is `"[aeiou]s"`, and you can kind of see what the syntax is doing here. The bracketed expression means "any of the things in the middle", so the expression as a whole means "any of the things in the middle" (i.e. vowels) followed by the letter `"s"`. When applied to our beer names we get this:

```
grep( pattern = "[aeiou]s", x = beers, value = TRUE )
```

```
## [1] "little creatures"
```

So it turns out that only `"little creatures"` contains a vowel followed by the letter `"s"`. But of course, had the data contained a beer like `"fosters"`, that would have matched as well because it contains the string `"os"`. However, I deliberately chose not to include it because Fosters is not – in my opinion – a proper beer.[124] As you can tell from this example, regular expressions are a neat tool for specifying *patterns* in text: in this case, "vowel then s". So they are definitely things worth

knowing about if you ever find yourself needing to work with a large body of text. However, since they are fairly complex and not necessary for any of the applications discussed in this book, I won't talk about them any further.

This page titled 17.8: Working with Text is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **7.8: Working with Text** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

17.9: Reading Unusual Data Files

In this section I'm going to switch topics (again!) and turn to the question of how you can load data from a range of different sources. Throughout this book I've assumed that your data are stored as an `.Rdata` file or as a "properly" formatted CSV file. And if so, then the basic tools that I discussed in Section 4.5 should be quite sufficient. However, in real life that's not a terribly plausible assumption to make, so I'd better talk about some of the other possibilities that you might run into.

17.9.1 Loading data from text files

The first thing I should point out is that if your data are saved as a text file but aren't *quite* in the proper CSV format, then there's still a pretty good chance that the `read.csv()` function (or equivalently, `read.table()`) will be able to open it. You just need to specify a few more of the optional arguments to the function. If you type `?read.csv` you'll see that the `read.csv()` function actually has several arguments that you can specify. Obviously you need to specify the `file` that you want it to load, but the others all have sensible default values. Nevertheless, you will sometimes need to change them. The ones that I've often found myself needing to change are:

- `header`. A lot of the time when you're storing data as a CSV file, the first row actually contains the column names and not data. If that's not true, you need to set `header = FALSE`.
- `sep`. As the name "comma separated value" indicates, the values in a row of a CSV file are usually separated by commas. This isn't universal, however. In Europe the decimal point is typically written as `,` instead of `.` and as a consequence it would be somewhat awkward to use `,` as the separator. Therefore it is not unusual to use `;` over there. At other times, I've seen a TAB character used. To handle these cases, we'd need to set `sep = ";"` or `sep = "\t"`.
- `quote`. It's conventional in CSV files to include a quoting character for textual data. As you can see by looking at the `booksales.csv}` file, this is usually a double quote character, `"`. But sometimes there is no quoting character at all, or you might see a single quote mark `'` used instead. In those cases you'd need to specify `quote = ""` or `quote = "'"`.
- `skip`. It's actually very common to receive CSV files in which the first few rows have nothing to do with the actual data. Instead, they provide a human readable summary of where the data came from, or maybe they include some technical info that doesn't relate to the data. To tell R to ignore the first (say) three lines, you'd need to set `skip = 3`
- `na.strings`. Often you'll get given data with missing values. For one reason or another, some entries in the table are missing. The data file needs to include a "special" string to indicate that the entry is missing. By default R assumes that this string is `NA`, since that's what *it* would do, but there's no universal agreement on what to use in this situation. If the file uses `???` instead, then you'll need to set `na.strings = "???"`.

It's kind of nice to be able to have all these options that you can tinker with. For instance, have a look at the data file shown pictured in Figure 7.1. This file contains almost the same data as the last file (except it doesn't have a header), and it uses a bunch of wacky features that you don't normally see in CSV files. In fact, it just so happens that I'm going to have to change all five of those arguments listed above in order to load this file. Here's how I would do it:

```
data <- read.csv( file = "./rbook-master/data/booksales2.csv",  # specify the name of
                  header = FALSE,           # variable names in the file?
                  skip = 8,                 # ignore the first 8 lines
                  quote = "*",              # what indicates text data?
                  sep = "\t",               # what separates different entries?
                  na.strings = "NFI" )      # what is the code for missing data?
```

If I now have a look at the data I've loaded, I see that this is what I've got:

```
head( data )
```

```
##          V1 V2  V3   V4
## 1   January 31   0 high
## 2 February  28 100 high
## 3    March  31 200  low
## 4    April  30  50  out
## 5    May    31  NA  out
## 6    June   30   0 high
```

Because I told R to expect `*` to be used as the quoting character instead of `"` ; to look for tabs (which we write like this: `\t`) instead of commas, and to skip the first 8 lines of the file, it's basically loaded the right data. However, since `booksales2.csv` doesn't contain the column names, R has made them up. Showing the kind of imagination I expect from insentient software, R decided to call them `V1` , `V2` , `V3` and `V4` . Finally, because I told it that the file uses "NFI" to denote missing data, R correctly figures out that the sales data for May are actually missing.

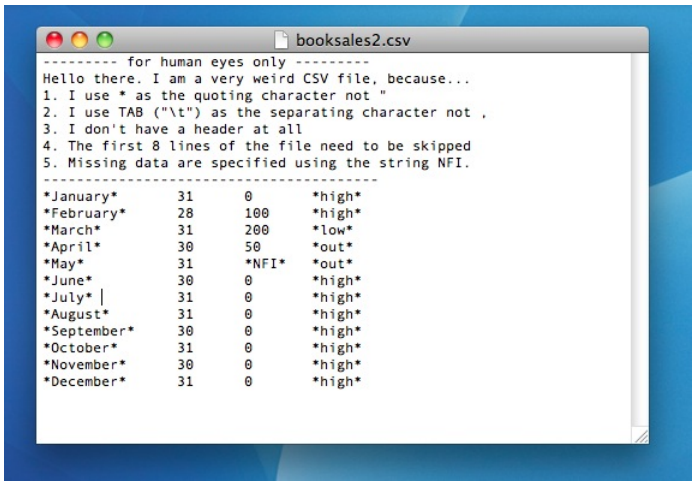


Figure 7.1: The `booksales2.csv` data file. It contains more or less the same data as the original `booksales.csv` data file, but has a lot of very quirky features.

In real life you'll rarely see data this stupidly formatted.[125]

17.9.2 Loading data from SPSS (and other statistics packages)

The commands listed above are the main ones we'll need for data files in this book. But in real life we have many more possibilities. For example, you might want to read data files in from other statistics programs. Since SPSS is probably the most widely used statistics package in psychology, it's worth briefly showing how to open SPSS data files (file extension `.sav`). It's surprisingly easy. The extract below should illustrate how to do so:

```
library( foreign )                # load the package
X <- read.spss( "./rbook-master/data/datafile.sav" )   # create a list containing the
X <- as.data.frame( X )           # convert to data frame
```

If you wanted to import from an SPSS file to a data frame directly, instead of importing a list and then converting the list to a data frame, you can do that too:

```
X <- read.spss( file = "datafile.sav", to.data.frame = TRUE )
```

And that's pretty much it, at least as far as SPSS goes. As far as other statistical software goes, the `foreign` package provides a wealth of possibilities. To open SAS files, check out the `read.ssd()` and `read.xport()` functions. To open data from Minitab, the `read.mtp()` function is what you're looking for. For Stata, the `read.dta()` function is what you want. For Systat, the `read.systat()` function is what you're after.

17.9.3 Loading Excel files

A different problem is posed by Excel files. Despite years of yelling at people for sending data to me encoded in a proprietary data format, I get sent a lot of Excel files. In general R does a pretty good job of opening them, but it's bit finicky because Microsoft don't seem to be terribly fond of people using non-Microsoft products, and go to some lengths to make it tricky. If you get an Excel file, my suggestion would be to open it up in Excel (or better yet, OpenOffice, since that's free software) and then save the spreadsheet as a CSV file. Once you've got the data in that format, you can open it using `read.csv()`. However, if for some reason you're desperate to open the `.xls` or `.xlsx` file directly, then you can use the `read.xls()` function in the `gdata` package:

```
library( gdata )                    # load the package
X <- read.xls( "datafile.xlsx" )    # create a data frame
```

This usually works. And if it doesn't, you're probably justified in "suggesting" to the person that sent you the file that they should send you a nice clean CSV file instead.

17.9.4 Loading Matlab (& Octave) files

A lot of scientific labs use Matlab as their default platform for scientific computing; or Octave as a free alternative. Opening Matlab data files (file extension `.mat`) slightly more complicated, and if it wasn't for the fact that Matlab is so very widespread and is an extremely good platform, I wouldn't mention it. However, since Matlab is so widely used, I think it's worth discussing briefly how to get Matlab and R to play nicely together. The way to do this is to install the `R.matlab` package (don't forget to install the dependencies too). Once you've installed and loaded the package, you have access to the `readMat()` function. As any Matlab user will know, the `.mat` files that Matlab produces are workspace files, very much like the `.Rdata` files that R produces. So you can't import a `.mat` file as a data frame. However, you can import it as a list. So, when we do this:

```
library( R.matlab )                    # load the package
data <- readMat( "matlabfile.mat" )    # read the data file to a list
```

The `data` object that gets created will be a list, containing one variable for every variable stored in the Matlab file. It's fairly straightforward, though there are some subtleties that I'm ignoring. In particular, note that if you don't have the `Rcompression` package, you can't open Matlab files above the version 6 format. So, if like me you've got a recent version of Matlab, and don't have the `Rcompression` package, you'll need to save your files using the `-v6` flag otherwise R can't open them.

Oh, and Octave users? The `foreign` package contains a `read.octave()` command. Just this once, the world makes life easier for you folks than it does for all those cashed-up swanky Matlab bastards.

17.9.5 Saving other kinds of data

Given that I talked extensively about how to load data from non-R files, it might be worth briefly mentioning that R is also pretty good at writing data into other file formats besides it's own native ones. I won't discuss them in this book, but the `write.csv()` function can write CSV files, and the `write.foreign()` function (in the `foreign` package) can write SPSS, Stata and SAS files. There are also a lot of low level commands that you can use to write very specific information to a file, so if you really, really needed to you could create your own `write.obscurefiletype()` function, but that's also a long way beyond the scope of this book. For now, all that I want you to recognise is that this capability is there if you need it.

17.9.6 done yet?

Of course not. If I've learned nothing else about R it's that you're *never bloody done*. This listing doesn't even come close to exhausting the possibilities. Databases are supported by the `RODBC`, `DBI`, and `RMySQL` packages among others. You can open webpages using the `RCurl` package. Reading and writing JSON objects is supported through the `rjson` package. And so on. In a sense, the right question is not so much "can R do this?" so much as "whereabouts in the wilds of CRAN *is* the damn package that does it?"

This page titled 17.9: Reading Unusual Data Files is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **7.9: Reading Unusual Data Files** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

17.10: Coercing Data from One Class to Another

Sometimes you want to change the variable class. This can happen for all sorts of reasons. Sometimes when you import data from files, it can come to you in the wrong format: numbers sometimes get imported as text, dates usually get imported as text, and many other possibilities besides. Regardless of how you've ended up in this situation, there's a very good chance that sometimes you'll want to convert a variable from one class into another one. Or, to use the correct term, you want to ***coerce*** the variable from one class into another. Coercion is a little tricky, and so I'll only discuss the very basics here, using a few simple examples.

Firstly, let's suppose we have a variable `x` that is *supposed* to be representing a number, but the data file that you've been given has encoded it as text. Let's imagine that the variable is something like this:

```
x <- "100"  # the variable
class(x)    # what class is it?
```

```
## [1] "character"
```

Obviously, if I want to do calculations using `x` in its current state, R is going to get very annoyed at me. It thinks that `x` is text, so it's not going to allow me to try to do mathematics using it! Obviously, we need to coerce `x` from character to numeric. We can do that in a straightforward way by using the `as.numeric()` function:

```
x <- as.numeric(x)  # coerce the variable
class(x)            # what class is it?
```

```
## [1] "numeric"
```

```
x + 1               # hey, addition works!
```

```
## [1] 101
```

Not surprisingly, we can also convert it back again if we need to. The function that we use to do this is the `as.character()` function:

```
x <- as.character(x)   # coerce back to text
class(x)               # check the class:
```

```
## [1] "character"
```

However, there's some fairly obvious limitations: you can't coerce the string `"hello world"` into a number because, well, there's isn't a number that corresponds to it. Or, at least, you can't do anything useful:

```
as.numeric( "hello world" )  # this isn't going to work.
```

```
## Warning: NAs introduced by coercion
```

```
## [1] NA
```

In this case R doesn't give you an error message; it just gives you a warning, and then says that the data is missing (see Section 4.6.1 for the interpretation of `NA`).

That gives you a feel for how to change between numeric and character data. What about logical data? To cover this briefly, coercing text to logical data is pretty intuitive: you use the `as.logical()` function, and the character strings `"T"`, `"TRUE"`, `"True"` and `"true"` all convert to the logical value of `TRUE`. Similarly `"F"`, `"FALSE"`, `"False"`, and `"false"` all become `FALSE`. All other strings convert to `NA`. When you go back the other way using `as.character()`, `TRUE` converts to `"TRUE"` and `FALSE` converts to `"FALSE"`. Converting numbers to logicals – again using `as.logical()` – is straightforward. Following the convention in the study of logic, the number `0` converts to `FALSE`. Everything else is `TRUE`. Going back using `as.numeric()`, `FALSE` converts to `0` and `TRUE` converts to `1`.

This page titled 17.10: Coercing Data from One Class to Another is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **7.10: Coercing Data from One Class to Another** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

17.11: Other Useful Data Structures

Up to this point we have encountered several different kinds of variables. At the simplest level, we've seen numeric data, logical data and character data. However, we've also encountered some more complicated kinds of variables, namely factors, formulas, data frames and lists. We'll see a few more specialised data structures later on in this book, but there's a few more generic ones that I want to talk about in passing. None of them are central to the rest of the book (and in fact, the only one we'll even see anywhere else is the matrix), but they do crop up a fair bit in real life.

17.11.1 Matrices

In various different places in this chapter I've made reference to an R data structure called a ***matrix***, and mentioned that I'd talk a bit more about matrices later on. That time has come. Much like a data frame, a matrix is basically a big rectangular table of data, and in fact there are quite a few similarities between the two. However, there are also some key differences, so it's important to talk about matrices in a little detail. Let's start by using `rbind()` to create a small matrix:[126]

```
row.1 <- c( 2,3,1 )          # create data for row 1
row.2 <- c( 5,6,7 )          # create data for row 2
M <- rbind( row.1, row.2 )   # row bind them into a matrix
print( M )                   # and print it out...
```

```
##       [,1] [,2] [,3]
## row.1    2    3    1
## row.2    5    6    7
```

The variable `M` is a matrix, which we can confirm by using the `class()` function. Notice that, when we bound the two vectors together, R retained the names of the original variables as row names. We could delete these if we wanted by typing `rownames(M)<-NULL`, but I generally prefer having meaningful names attached to my variables, so I'll keep them. In fact, let's also add some highly unimaginative column names as well:

```
colnames(M) <- c( "col.1", "col.2", "col.3" )
print(M)
```

```
##       col.1 col.2 col.3
## row.1     2     3     1
## row.2     5     6     7
```

You can use square brackets to subset a matrix in much the same way that you can for data frames, again specifying a row index and then a column index. For instance, `M[2,3]` pulls out the entry in the 2nd row and 3rd column of the matrix (i.e., `7`), whereas `M[2,]` pulls out the entire 2nd row, and `M[,3]` pulls out the entire 3rd column. However, it's worth noting that when you pull out a column, R will print the results horizontally, not vertically. The reason for this relates to how matrices (and arrays generally) are implemented. The original matrix `M` is treated as a two-dimensional objects, containing 2 rows and 3 columns. However, whenever you pull out a single row or a single column, the result is considered to be one-dimensional. As far as R is concerned there's no real reason to distinguish between a one-dimensional object printed vertically (a column) and a one-dimensional object printed horizontally (a row), and it prints them all out horizontally.[127] There is also a way of using only a single index, but due to the internal structure to how R defines a matrix, it works very differently to what we saw previously with data frames.

The single-index approach is illustrated in Table 7.5 but I don't really want to focus on it since we'll never really need it for this book, and matrices don't play anywhere near as large a role in this book as data frames do. The reason for these differences is that for this is that, for both data frames and matrices, the "row and column" version exists to allow the human user to interact with the object in the psychologically meaningful way: since both data frames and matrices are basically just tables of data, it's the same in

each case. However, the single-index version is really a method for you to interact with the object in terms of its internal structure, and the internals for data frames and matrices are quite different.

Table 7.5: The row and column version, which is identical to the corresponding indexing scheme for a data frame of the same size.

Row	Col.1	Col.2	Col.3
Row 1	[1,1]	[1,2]	[1,3]
Row 2	[2,1]	[2,2]	[2,3]

Table 7.5: The single-index version, which is quite different to what we would get with a data frame.

Row	Col.1	Col.2	Col.3
Row 1	1	3	5
Row 2	2	4	6

The critical difference between a data frame and a matrix is that, in a data frame, we have this notion that each of the columns corresponds to a different variable: as a consequence, the columns in a data frame can be of different data types. The first column could be numeric, and the second column could contain character strings, and the third column could be logical data. In that sense, there is a fundamental asymmetry build into a data frame, because of the fact that columns represent variables (which can be qualitatively different to each other) and rows represent cases (which cannot). Matrices are intended to be thought of in a different way. At a fundamental level, a matrix really is just *one* variable: it just happens that this one variable is formatted into rows and columns. If you want a matrix of numeric data, every single element in the matrix *must* be a number. If you want a matrix of character strings, every single element in the matrix *must* be a character string. If you try to mix data of different types together, then R will either spit out an error, or quietly coerce the underlying data into a list. If you want to find out what class R secretly thinks the data within the matrix is, you need to do something like this:

```
class( M[1] )
```

```
## [1] "numeric"
```

You can't type `class(M)`, because all that will happen is R will tell you that `M` is a matrix: we're not interested in the class of the matrix itself, we want to know what class the underlying data is assumed to be. Anyway, to give you a sense of how R enforces this, let's try to change one of the elements of our numeric matrix into a character string:

```
M[1,2] <- "text"
M
```

```
##       col.1 col.2  col.3
## row.1 "2"   "text" "1"
## row.2 "5"   "6"    "7"
```

It looks as if R has coerced all of the data in our matrix into character strings. And in fact, if we now typed in `class(M[1])` we'd see that this is exactly what has happened. If you alter the contents of one element in a matrix, R will change the underlying data type as necessary.

There's only one more thing I want to talk about regarding matrices. The concept behind a matrix is very much a mathematical one, and in mathematics a matrix is a most definitely a two-dimensional object. However, when doing data analysis, we often have reasons to want to use higher dimensional tables (e.g., sometimes you need to cross-tabulate three variables against each other). You can't do this with matrices, but you can do it with ***arrays***. An array is just like a matrix, except it can have more than two

dimensions if you need it to. In fact, as far as R is concerned a matrix is just a special kind of array, in much the same way that a data frame is a special kind of list. I don't want to talk about arrays too much, but I will very briefly show you an example of what a 3D array looks like. To that end, let's cross tabulate the `speaker` and `utterance` variables from the `nightgarden.Rdata` data file, but we'll add a third variable to the cross-tabs this time, a logical variable which indicates whether or not I was still awake at this point in the show:

```
dan.awake <- c( TRUE,TRUE,TRUE,TRUE,TRUE,FALSE,FALSE,FALSE,FALSE,FALSE )
```

Now that we've got all three variables in the workspace (assuming you loaded the `nightgarden.Rdata` data earlier in the chapter) we can construct our three way cross-tabulation, using the `table()` function.

```
xtab.3d <- table( speaker, utterance, dan.awake )
xtab.3d
```

```
## , , dan.awake = FALSE
##
##              utterance
## speaker       ee onk oo pip
##   makka-pakka  0   2  0   2
##   tombliboo    0   0  1   0
##   upsy-daisy   0   0  0   0
##
## , , dan.awake = TRUE
##
##              utterance
## speaker       ee onk oo pip
##   makka-pakka  0   0  0   0
##   tombliboo    1   0  0   0
##   upsy-daisy   0   2  0   2
```

Hopefully this output is fairly straightforward: because R can't print out text in three dimensions, what it does is show a sequence of 2D slices through the 3D table. That is, the `, , dan.awake = FALSE` part indicates that the 2D table that follows below shows the 2D cross-tabulation of `speaker` against `utterance` only for the `dan.awake = FALSE` instances, and so on.[128]

17.11.2 Ordered factors

One topic that I neglected to mention when discussing factors previously (Section 4.7 is that there are actually two different types of factor in R, unordered factors and ordered factors. An unordered factor corresponds to a nominal scale variable, and all of the factors we've discussed so far in this book have been unordered (as will all the factors used anywhere else except in this section). However, it's often very useful to explicitly tell R that your variable is *ordinal scale*, and if so you need to declare it to be an ***ordered factor***. For instance, earlier in this chapter we made use of a variable consisting of Likert scale data, which we represented as the `likert.raw` variable:

```
likert.raw
```

```
##  [1] 1 7 3 4 4 4 2 6 5 5
```

We can declare this to be an ordered factor in by using the `factor()` function, and setting `ordered = TRUE` . To illustrate how this works, let's create an ordered factor called `likert.ordinal` and have a look at it:

```
likert.ordinal <- factor( x = likert.raw,        # the raw data
                          levels = seq(7,1,-1),  # strongest agreement is 1, weakest
                          ordered = TRUE )       # and it's ordered
print( likert.ordinal )
```

```
##  [1] 1 7 3 4 4 4 2 6 5 5
## Levels: 7 < 6 < 5 < 4 < 3 < 2 < 1
```

Notice that when we print out the ordered factor, R explicitly tells us what order the levels come in. Because I wanted to order my levels in terms of *increasing* strength of agreement, and because a response of 1 corresponded to the strongest agreement and 7 to the strongest disagreement, it was important that I tell R to encode 7 as the lowest value and 1 as the largest. Always check this when creating an ordered factor: it's very easy to accidentally encode your data "upside down" if you're not paying attention. In any case, note that we can (and should) attach meaningful names to these factor levels by using the `levels()` function, like this:

```
levels( likert.ordinal ) <- c( "strong.disagree", "disagree", "weak.disagree",
                               "neutral", "weak.agree", "agree", "strong.agree" )
print( likert.ordinal )
```

```
##  [1] strong.agree    strong.disagree weak.agree      neutral
##  [5] neutral         neutral         agree           disagree
##  [9] weak.disagree   weak.disagree
## 7 Levels: strong.disagree < disagree < weak.disagree < ... < strong.agree
```

One nice thing about using ordered factors is that there are a lot of analyses for which R automatically treats ordered factors differently from unordered factors, and generally in a way that is more appropriate for ordinal data. However, since I don't discuss that in this book, I won't go into details. Like so many things in this chapter, my main goal here is to make you aware that R has this capability built into it; so if you ever need to start thinking about ordinal scale variables in more detail, you have at least some idea where to start looking!

17.11.3 Dates and times

Times and dates are very annoying types of data. To a first approximation we can say that there are 365 days in a year, 24 hours in a day, 60 minutes in an hour and 60 seconds in a minute, but that's not quite correct. The length of the solar day is not exactly 24 hours, and the length of solar year is not exactly 365 days, so we have a complicated system of corrections that have to be made to keep the time and date system working. On top of that, the measurement of time is usually taken relative to a local time zone, and most (but not all) time zones have both a standard time and a daylight savings time, though the date at which the switch occurs is not at all standardised. So, as a form of data, times and dates *suck*. Unfortunately, they're also important. Sometimes it's possible to avoid having to use any complicated system for dealing with times and dates. Often you just want to know what year something happened in, so you can just use numeric data: in quite a lot of situations something as simple as `this.year <- 2011` works just fine. If you can get away with that for your application, this is probably the best thing to do. However, sometimes you really do need to know the actual date. Or, even worse, the actual time. In this section, I'll very briefly introduce you to the basics of how R deals with date and time data. As with a lot of things in this chapter, I won't go into details because I don't use this kind of data anywhere else in the book. The goal here is to show you the basics of what you need to do if you ever encounter this kind of data in real life. And then we'll all agree never to speak of it again.

To start with, let's talk about the date. As it happens, modern operating systems are very good at keeping track of the time and date, and can even handle all those annoying timezone issues and daylight savings pretty well. So R takes the quite sensible view that it can just ask the operating system what the date is. We can pull the date using the `Sys.Date()` function:

```
today <- Sys.Date()  # ask the operating system for the date
print(today)         # display the date
```

```
## [1] "2018-12-30"
```

Okay, that seems straightforward. But, it does rather look like `today` is just a character string, doesn't it? That would be a problem, because dates really do have a numeric character to them, and it would be nice to be able to do basic addition and subtraction to them. Well, fear not. If you type in `class(today)`, R will tell you that the class of the `today` variable is `"Date"`. What this means is that, hidden underneath this text string that prints out an actual date, R actually has a numeric representation.[129] What that means is that you actually can add and subtract days. For instance, if we add 1 to `today`, R will print out the date for tomorrow:

```
today + 1
```

```
## [1] "2018-12-31"
```

Let's see what happens when we add 365 days:

```
today + 365
```

```
## [1] "2019-12-30"
```

This is particularly handy if you forget that a year is a leap year since in that case you'd probably get it wrong is doing this in your head. R provides a number of functions for working with dates, but I don't want to talk about them in any detail. I will, however, make passing mention of the `weekdays()` function which will tell you what day of the week a particular date corresponded to, which is extremely convenient in some situations:

```
weekdays( today )
```

```
## [1] "Sunday"
```

I'll also point out that you can use the `as.Date()` to convert various different kinds of data into dates. If the data happen to be strings formatted exactly according to the international standard notation (i.e., `yyyy-mm-dd`) then the conversion is straightforward, because that's the format that R expects to see by default. You can convert dates from other formats too, but it's slightly trickier, and beyond the scope of this book.

What about times? Well, times are even more annoying, so much so that I don't intend to talk about them at all in this book, other than to point you in the direction of some vaguely useful things. R itself does provide you with some tools for handling time data, and in fact there are two separate classes of data that are used to represent times, known by the odd names `POSIXct` and `POSIXlt`. You can use these to work with times if you want to, but for most applications you would probably be better off downloading the `chron` package, which provides some much more user friendly tools for working with times and dates.

This page titled 17.11: Other Useful Data Structures is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **7.11: Other Useful Data Structures** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

17.12: Miscellaneous Topics

To finish this chapter, I have a few topics to discuss that don't really fit in with any of the other things in this chapter. They're all kind of useful things to know about, but they are really just "odd topics" that don't fit with the other examples. Here goes:

17.12.1 problems with floating point arithmetic

If I've learned nothing else about transfinite arithmetic (and I haven't) it's that infinity is a tedious and inconvenient concept. Not only is it annoying and counterintuitive at times, but it has nasty practical consequences. As we were all taught in high school, there are some numbers that *cannot* be represented as a decimal number of finite length, nor can they be represented as any kind of fraction between two whole numbers; $\sqrt{2}$, π and e, for instance. In everyday life we mostly don't care about this. I'm perfectly happy to approximate π as 3.14, quite frankly. Sure, this does produce some rounding errors from time to time, and if I'd used a more detailed approximation like 3.1415926535 I'd be less likely to run into those issues, but in all honesty I've never needed my calculations to be *that* precise. In other words, although our pencil and paper calculations cannot represent the number π exactly as a decimal number, we humans are smart enough to realise that we don't care. Computers, unfortunately, are dumb … and you don't have to dig too deep in order to run into some very weird issues that arise because they can't represent numbers perfectly. Here is my favourite example:

```
0.1 + 0.2 == 0.3
```

```
## [1] FALSE
```

Obviously, R has made a mistake here, because this is definitely the wrong answer. Your first thought might be that R is broken, and you might be considering switching to some other language. But you can reproduce the same error in dozens of different programming languages, so the issue isn't specific to R. Your next thought might be that it's something in the hardware, but you can get the same mistake on any machine. It's something deeper than that.

The fundamental issue at hand is ***floating point arithmetic***, which is a fancy way of saying that computers will *always* round a number to fixed number of significant digits. The exact number of significant digits that the computer stores isn't important to us:[130] what matters is that whenever the number that the computer is trying to store is very long, you get rounding errors. That's actually what's happening with our example above. There are teeny tiny rounding errors that have appeared in the computer's storage of the numbers, and these rounding errors have in turn caused the internal storage of `0.1 + 0.2` to be a tiny bit different from the internal storage of `0.3`. How big are these differences? Let's ask R:

```
0.1 + 0.2 - 0.3
```

```
## [1] 5.551115e-17
```

Very tiny indeed. No sane person would care about differences that small. But R is not a sane person, and the equality operator `==` is very literal minded. It returns a value of `TRUE` only when the two values that it is given are absolutely identical to each other. And in this case they are not. However, this only answers half of the question. The other half of the question is, why are we getting these rounding errors when we're only using nice simple numbers like 0.1, 0.2 and 0.3? This seems a little counterintuitive. The answer is that, like most programming languages, R doesn't store numbers using their *decimal* expansion (i.e., base 10: using digits 0, 1, 2 …, 9). We humans like to write our numbers in base 10 because we have 10 fingers. But computers don't have fingers, they have transistors; and transistors are built to store 2 numbers not 10. So you can see where this is going: the internal storage of a number in R is based on its *binary* expansion (i.e., base 2: using digits 0 and 1). And unfortunately, here's what the binary expansion of 0.1 looks like:

$$.1(\text{decimal}) = .0001100110011001100110011001100110011001100110011001101\ldots(\text{binary})$$

and the pattern continues forever. In other words, from the perspective of your computer, which likes to encode numbers in binary,[131] 0.1 is not a simple number at all. To a computer, 0.1 is actually an infinitely long binary number! As a consequence, the

computer can make minor errors when doing calculations here.

With any luck you now understand the problem, which ultimately comes down to the twin fact that (1) we usually think in decimal numbers and computers usually compute with binary numbers, and (2) computers are finite machines and can't store infinitely long numbers. The only questions that remain are when you should care and what you should do about it. Thankfully, you don't have to care very often: because the rounding errors are small, the only practical situation that I've seen this issue arise for is when you want to test whether an arithmetic fact holds exactly numbers are identical (e.g., is someone's response time equal to *exactly* 2×0.33 seconds?) This is pretty rare in real world data analysis, but just in case it does occur, it's better to use a test that allows for a small *tolerance*. That is, if the difference between the two numbers is below a certain threshold value, we deem them to be equal for all practical purposes. For instance, you could do something like this, which asks whether the difference between the two numbers is less than a tolerance of 10−10

```
abs( 0.1 + 0.2 - 0.3 ) < 10^-10
```

```
## [1] TRUE
```

To deal with this problem, there is a function called `all.equal()` that lets you test for equality but allows a small tolerance for rounding errors:

```
all.equal( 0.1 + 0.2, 0.3 )
```

```
## [1] TRUE
```

17.12.2 recycling rule

There's one thing that I haven't mentioned about how vector arithmetic works in R, and that's the ***recycling rule***. The easiest way to explain it is to give a simple example. Suppose I have two vectors of different length, `x` and `y`, and I want to add them together. It's not obvious what that actually means, so let's have a look at what R does:

```
x <- c( 1,1,1,1,1,1 )  # x is length 6
y <- c( 0,1 )          # y is length 2
x + y                  # now add them:
```

```
## [1] 1 2 1 2 1 2
```

As you can see from looking at this output, what R has done is "recycle" the value of the shorter vector (in this case `y`) several times. That is, the first element of `x` is added to the first element of `y`, and the second element of `x` is added to the second element of `y`. However, when R reaches the third element of `x` there isn't any corresponding element in `y`, so it returns to the beginning: thus, the third element of `x` is added to the *first* element of `y`. This process continues until R reaches the last element of `x`. And that's all there is to it really. The same recycling rule also applies for subtraction, multiplication and division. The only other thing I should note is that, if the length of the longer vector isn't an exact multiple of the length of the shorter one, R still does it, but also gives you a warning message:

```
x <- c( 1,1,1,1,1 )    # x is length 5
y <- c( 0,1 )          # y is length 2
x + y                  # now add them:
```

```
## Warning in x + y: longer object length is not a multiple of shorter object
## length
```

```
## [1] 1 2 1 2 1
```

17.12.3 introduction to environments

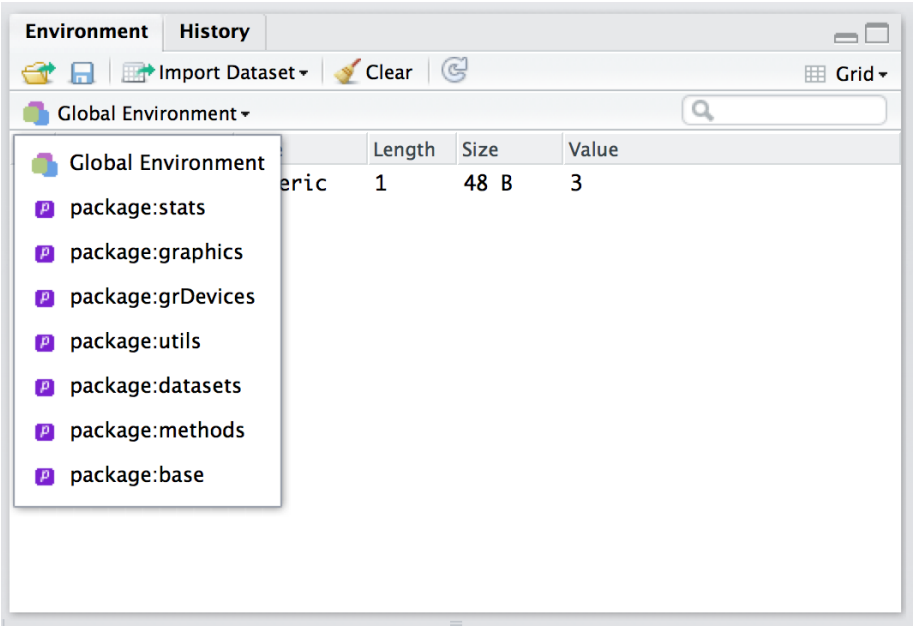


Figure 7.2: The environment panel in Rstudio can actually show you the contents of any loaded package: each package defines a separate environment, so you can select the one you want to look at in this panel.

In this section I want to ask a slightly different question: what *is* the workspace exactly? This question seems simple, but there's a fair bit to it. This section can be skipped if you're not really interested in the technical details. In the description I gave earlier, I talked about the workspace as an abstract location in which R variables are stored. That's basically true, but it hides a couple of key details. For example, any time you have R open, it has to store *lots* of things in the computer's memory, not just your variables. For example, the `who()` function that I wrote has to be stored in memory somewhere, right? If it weren't I wouldn't be able to use it. That's pretty obvious. But equally obviously it's not in the workspace either, otherwise you should have seen it! Here's what's happening. R needs to keep track of a lot of different things, so what it does is organise them into ***environments***, each of which can contain lots of different variables and functions. Your workspace is one such environment. Every package that you have loaded is another environment. And every time you call a function, R briefly creates a temporary environment in which the function itself can work, which is then deleted after the calculations are complete. So, when I type in `search()` at the command line

```
search()
```

```
##  [1] ".GlobalEnv"        "package:lsr"       "package:stats"
##  [4] "package:graphics"  "package:grDevices" "package:utils"
##  [7] "package:datasets"  "package:methods"   "Autoloads"
## [10] "package:base"
```

what I'm actually looking at is a *sequence of environments*. The first one, `".GlobalEnv"` is the technically-correct name for your workspace. No-one really calls it that: it's either called the workspace or the global environment. And so when you type in `objects()` or `who()` what you're really doing is listing the contents of `".GlobalEnv"` . But there's no reason why we can't look up the contents of these other environments using the `objects()` function (currently `who()` doesn't support this). You just have to be a bit more explicit in your command. If I wanted to find out what is in the `package:stats` environment (i.e., the environment into which the contents of the `stats` package have been loaded), here's what I'd get

```
head(objects("package:stats"))
```

```
## [1] "acf"        "acf2AR"     "add.scope"  "add1"       "addmargins"
## [6] "aggregate"
```

where this time I've used head() to hide a lot of output because the stats package contains about 500 functions. In fact, you can actually use the environment panel in Rstudio to browse any of your loaded packages (just click on the text that says "Global Environment" and you'll see a dropdown menu like the one shown in Figure 7.2). The key thing to understand then, is that you can access any of the R variables and functions that are stored in one of these environments, precisely because those are the environments that you have loaded![132]

17.12.4 Attaching a data frame

The last thing I want to mention in this section is the `attach()` function, which you often see referred to in introductory R books. Whenever it is introduced, the author of the book usually mentions that the `attach()` function can be used to "attach" the data frame to the search path, so you don't have to use the `$` operator. That is, if I use the command `attach(df)` to attach my data frame, I no longer need to type `df$variable`, and instead I can just type `variable`. This is true as far as it goes, but it's very misleading and novice users often get led astray by this description, because it hides a lot of critical details.

Here is the very abridged description: when you use the `attach()` function, what R does is create an entirely new *environment* in the search path, just like when you load a package. Then, what it does is *copy* all of the variables in your data frame into this new environment. When you do this, however, you end up with two completely different versions of all your variables: one in the original data frame, and one in the new environment. Whenever you make a statement like `df$variable` you're working with the variable inside the data frame; but when you just type `variable` you're working with the copy in the new environment. And here's the part that really upsets new users: *changes to one version are not reflected in the other version*. As a consequence, it's really easy for R to end up with different value stored in the two different locations, and you end up really confused as a result.

To be fair to the writers of the `attach()` function, the help documentation does actually state all this quite explicitly, and they even give some examples of how this can cause confusion at the bottom of the help page. And I can actually see how it can be very useful to create copies of your data in a separate location (e.g., it lets you make all kinds of modifications and deletions to the data without having to touch the original data frame). However, I don't think it's helpful for new users, since it means you have to be very careful to keep track of which copy you're talking about. As a consequence of all this, for the purpose of this book I've decided not to use the `attach()` function. It's something that you can investigate yourself once you're feeling a little more confident with R, but I won't do it here.

This page titled 17.12: Miscellaneous Topics is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **7.12: Miscellaneous Topics** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

17.13: Summary

Obviously, there's no real coherence to this chapter. It's just a grab bag of topics and tricks that can be handy to know about, so the best wrap up I can give here is just to repeat this list:

- Section 7.1. Tabulating data.
- Section 7.2. Transforming or recoding a variable.
- Section 7.3. Some useful mathematical functions.
- Section 7.4. Extracting a subset of a vector.
- Section 7.5. Extracting a subset of a data frame.
- Section 7.6. Sorting, flipping or merging data sets.
- Section 7.7. Reshaping a data frame.
- Section 7.8. Manipulating text.
- Section 7.9. Opening data from different file types.
- Section 7.10. Coercing data from one type to another.
- Section 7.11. Other important data types.
- Section 7.12. Miscellaneous topics.

There are a number of books out there that extend this discussion. A couple of my favourites are Spector (2008) "Data Manipulation with R" and Teetor (2011) "R Cookbook".

References

Spector, P. 2008. *Data Manipulation with R*. New York, NY: Springer.

Teetor, P. 2011. *R Cookbook*. Sebastopol, CA: O'Reilly.

103. The quote comes from *Home is the Hangman*, published in 1975.
104. As usual, you can assign this output to a variable. If you type `speaker.freq <- table(speaker)` at the command prompt R will store the table as a variable. If you then type `class(speaker.freq)` you'll see that the output is actually of class `table` . The key thing to note about a table object is that it's basically a matrix (see Section 7.11.1.
105. It's worth noting that there's also a more powerful function called `recode()` function in the `car` package that I won't discuss in this book but is worth looking into if you're looking for a bit more flexibility.
106. If you've read further into the book, and are re-reading this section, then a good example of this would be someone choosing to do an ANOVA using `age.group3` as the grouping variable, instead of running a regression using `age` as a predictor. There are sometimes good reasons for do this: for instance, if the relationship between `age` and your outcome variable is highly non-linear, and you aren't comfortable with trying to run non-linear regression! However, unless you really do have a good rationale for doing this, it's best not to. It tends to introduce all sorts of other problems (e.g., the data will probably violate the normality assumption), and you can lose a lot of power.
107. The real answer is 0: $10 for a sandwich is a total ripoff so I should go next door and buy noodles.
108. Again, I doubt that's the right "real world" answer. I suspect that most sandwich shops won't allow you to pay off your debts to them in sandwiches. But you get the idea.
109. Actually, that's a bit of a lie: the `log()` function is more flexible than that, and can be used to calculate logarithms in *any* base. The `log()` function has a `base` argument that you can specify, which has a default value of e. Thus `log10(1000)` is actually equivalent to `log(x = 1000, base = 10)` .
110. It's also worth checking out the `match()` function
111. It also works on data frames if you ever feel the need to import all of your variables from the data frame into the workspace. This can be useful at times, though it's not a good idea if you have large data sets or if you're working with multiple data sets at once. In particular, if you do this, never forget that you now have *two* copies of all your variables, one in the workspace and another in the data frame.
112. You can do this yourself using the `make.names()` function. In fact, this is itself a handy thing to know about. For example, if you want to convert the names of the variables in the `speech.by.char` list into valid R variable names, you could use a

command like this: `names(speech.by.char) <- make.names(names(speech.by.char))` . However, I won't go into details here.

113. Conveniently, if you type `rownames(df) <- NULL` R will renumber all the rows from scratch. For the `df` data frame, the labels that currently run from 7 to 10 will be changed to go from 1 to 4.
114. Actually, you can make the `subset()` function behave this way by using the optional `drop` argument, but by default `subset()` does not drop, which is probably more sensible and more intuitive to novice users.
115. Specifically, recursive indexing, a handy tool in some contexts but not something that I want to discuss here.
116. Remember, `print()` is generic: see Section 4.11.
117. Note for advanced users: both of these functions are just wrappers to the `matrix()` function, which is pretty flexible in terms of the ability to convert vectors into matrices. Also, while I'm on this topic, I'll briefly mention the fact that if you're a Matlab user and looking for an equivalent of Matlab's `repmat()` function, I'd suggest checking out the `matlab` package which contains R versions of a lot of handy Matlab functions.
118. The function you need for that is called `as.data.frame()` .
119. In truth, I suspect that most of the cases when you can sensibly flip a data frame occur when all of the original variables are measurements of the same type (e.g., all variables are response times), and if so you could easily have chosen to encode your data as a matrix instead of as a data frame. But since people do sometimes prefer to work with data frames, I've written the `tFrame()` function for the sake of convenience. I don't really think it's something that is needed very often.
120. This limitation is deliberate, by the way: if you're getting to the point where you want to do something more complicated, you should probably start learning how to use `reshape()` , `cast()` and `melt()` or some of other the more advanced tools. The `wideToLong()` and `longToWide()` functions are included only to help you out when you're first starting to use R.
121. To be honest, it does bother me a little that the default value of `sep` is a space. Normally when I want to paste strings together I don't want any separator character, so I'd prefer it if the default were `sep=""` . To that end, it's worth noting that there's also a `paste0()` function, which is identical to `paste()` except that it always assumes that `sep=""` . Type `?paste` for more information about this.
122. Note that you can capture the output from `cat()` if you want to, but you have to be sneaky and use the `capture.output()` function. For example, the command `x <- capture.output(cat(hw,ng))` would work just fine.
123. Sigh. For advanced users: R actually supports two different ways of specifying regular expressions. One is the POSIX standard, the other is to use Perl-style regular expressions. The default is generally POSIX. If you understand regular expressions, that probably made sense to you. If not, don't worry. It's not important.
124. I thank Amy Perfors for this example.
125. If you're lucky.
126. You can also use the `matrix()` command itself, but I think the "binding" approach is a little more intuitive.
127. This has some interesting implications for how matrix algebra is implemented in R (which I'll admit I initially found odd), but that's a little beyond the scope of this book. However, since there will be a small proportion of readers that do care, I'll quickly outline the basic thing you need to get used to: when multiplying a matrix by a vector (or one-dimensional array) using the `\%*\%` operator R will attempt to interpret the vector (or 1D array) as either a row-vector or column-vector, depending on whichever one makes the multiplication work. That is, suppose M is the 2×3 matrix, and v is a 1×3 row vector. It is impossible to multiply Mv, since the dimensions don't conform, but you *can* multiply by the corresponding column vector, Mvt. So, if I set `v <- M[2,]` and then try to calculate `M \%*\% v` , which you'd think would fail, it actually works because R treats the one dimensional array as if it were a column vector for the purposes of matrix multiplication. Note that if both objects are one dimensional arrays/vectors, this leads to ambiguity since vvt (inner product) and vtv (outer product) yield different answers. In this situation, the `\%*\%` operator returns the inner product not the outer product. To understand all the details, check out the help documentation.
128. I should note that if you type `class(xtab.3d)` you'll discover that this is a `"table"` object rather than an `"array"` object. However, this labelling is only skin deep. The underlying data structure here is actually an array. Advanced users may wish to check this using the command `class(unclass(xtab.3d))` , but it's not important for our purposes. All I really want to do in this section is show you what the output looks like when you encounter a 3D array.
129. Date objects are coded as the number of days that have passed since January 1, 1970.

130. For advanced users: type `?double` for more information.
131. Or at least, that's the default. If all your numbers are integers (whole numbers), then you can explicitly tell R to store them as integers by adding an `L` suffix at the end of the number. That is, an assignment like `x <- 2L` tells R to assign `x` a value of 2, and to store it as an integer rather than as a binary expansion. Type `?integer` for more details.
132. For advanced users: that's a little over simplistic in two respects. First, it's a terribly imprecise way of talking about scoping. Second, it might give you the impression that all the variables in question are actually loaded into memory. That's not quite true, since that would be very wasteful of memory. Instead R has a "lazy loading" mechanism, in which what R actually does is create a "promise" to load those objects if they're actually needed. For details, check out the `delayedAssign()` function.

This page titled 17.13: Summary is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **7.13: Summary** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

CHAPTER OVERVIEW

18: Basic Programming

> *Machine dreams hold a special vertigo.*
>
> *–William Gibson*[133]

Up to this point in the book I've tried hard to avoid using the word "programming" too much because – at least in my experience – it's a word that can cause a lot of fear. For one reason or another, programming (like mathematics and statistics) is often perceived by people on the "outside" as a black art, a magical skill that can be learned only by some kind of super-nerd. I think this is a shame. It's certainly true that advanced programming is a very specialised skill: several different skills actually, since there's quite a lot of different kinds of programming out there. However, the *basics* of programming aren't all that hard, and you can accomplish a lot of very impressive things just using those basics.

With that in mind, the goal of this chapter is to discuss a few basic programming concepts and how to apply them in R. However, before I do, I want to make one further attempt to point out just how non-magical programming really is, via one very simple observation: *you already know how to do it*. Stripped to its essentials, programming is nothing more (and nothing less) than the process of writing out a bunch of instructions that a computer can understand. To phrase this slightly differently, when you write a computer program, you need to write it in a ***programming language*** that the computer knows how to interpret. R is one such language. Although I've been having you type all your commands at the command prompt, and all the commands in this book so far have been shown as if that's what I were doing, it's also quite possible (and as you'll see shortly, shockingly easy) to write a program using these R commands. In other words, if this is the first time reading this book, then you're only one short chapter away from being able to legitimately claim that you can program in R, albeit at a beginner's level.

18.1: Scripts
18.2: Loops
18.3: Conditional Statements
18.4: Writing Functions
18.5: Implicit Loops
18.6: Summary

This page titled 18: Basic Programming is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

18.1: Scripts

Computer programs come in quite a few different forms: the kind of program that we're mostly interested in from the perspective of everyday data analysis using R is known as a ***script***. The idea behind a script is that, instead of typing your commands into the R console one at a time, instead you write them all in a text file. Then, once you've finished writing them and saved the text file, you can get R to execute all the commands in your file by using the `source()` function. In a moment I'll show you exactly how this is done, but first I'd better explain why you should care.

18.1.1 scripts?

Before discussing scripting and programming concepts in any more detail, it's worth stopping to ask why you should bother. After all, if you look at the R commands that I've used everywhere else this book, you'll notice that they're all formatted as if I were typing them at the command line. Outside this chapter you won't actually see any scripts. Do not be fooled by this. The reason that I've done it that way is purely for pedagogical reasons. My goal in this book is to teach statistics and to teach R. To that end, what *I've* needed to do is chop everything up into tiny little slices: each section tends to focus on one kind of statistical concept, and only a smallish number of R functions. As much as possible, I want you to see what each function does in isolation, one command at a time. By forcing myself to write everything as if it were being typed at the command line, it imposes a kind of discipline on me: it *prevents* me from piecing together lots of commands into one big script. From a teaching (and learning) perspective I think that's the right thing to do… but from a *data analysis* perspective, it is not. When you start analysing real world data sets, you will rapidly find yourself needing to write scripts.

To understand why scripts are so very useful, it may be helpful to consider the drawbacks to typing commands directly at the command prompt. The approach that we've been adopting so far, in which you type commands one at a time, and R sits there patiently in between commands, is referred to as the ***interactive*** style. Doing your data analysis this way is rather like having a conversation … a very annoying conversation between you and your data set, in which you and the data aren't directly speaking to each other, and so you have to rely on R to pass messages back and forth. This approach makes a lot of sense when you're just trying out a few ideas: maybe you're trying to figure out what analyses are sensible for your data, or maybe just you're trying to remember how the various R functions work, so you're just typing in a few commands until you get the one you want. In other words, the interactive style is very useful as a tool for exploring your data. However, it has a number of drawbacks:

- *It's hard to save your work effectively*. You can save the workspace, so that later on you can load any variables you created. You can save your plots as images. And you can even save the history or copy the contents of the R console to a file. Taken together, all these things let you create a reasonably decent record of what you did. But it does leave a lot to be desired. It seems like you ought to be able to save a single file that R could use (in conjunction with your raw data files) and reproduce everything (or at least, everything interesting) that you did during your data analysis.
- *It's annoying to have to go back to the beginning when you make a mistake.* Suppose you've just spent the last two hours typing in commands. Over the course of this time you've created lots of new variables and run lots of analyses. Then suddenly you realise that there was a nasty typo in the first command you typed, so all of your later numbers are wrong. Now you have to fix that first command, and then spend another hour or so combing through the R history to try and recreate what you did.
- *You can't leave notes for yourself.* Sure, you can scribble down some notes on a piece of paper, or even save a Word document that summarises what you did. But what you really want to be able to do is write down an English translation of your R commands, preferably right "next to" the commands themselves. That way, you can look back at what you've done and actually remember what you were doing. In the simple exercises we've engaged in so far, it hasn't been all that hard to remember what you were doing or why you were doing it, but only because everything we've done could be done using only a few commands, and you've never been asked to reproduce your analysis six months after you originally did it! When your data analysis starts involving hundreds of variables, and requires quite complicated commands to work, then you really, really need to leave yourself some notes to explain your analysis to, well, yourself.
- *It's nearly impossible to reuse your analyses later, or adapt them to similar problems*. Suppose that, sometime in January, you are handed a difficult data analysis problem. After working on it for ages, you figure out some really clever tricks that can be used to solve it. Then, in September, you get handed a really similar problem. You can sort of remember what you did, but not very well. You'd like to have a clean record of what you did last time, how you did it, and why you did it the way you did. Something like that would really help you solve this new problem.

- *It's hard to do anything except the basics.* There's a nasty side effect of these problems. Typos are inevitable. Even the best data analyst in the world makes a lot of mistakes. So the chance that you'll be able to string together dozens of correct R commands in a row are very small. So unless you have some way around this problem, you'll never really be able to do anything other than simple analyses.
- *It's difficult to share your work other people.* Because you don't have this nice clean record of what R commands were involved in your analysis, it's not easy to share your work with other people. Sure, you can send them all the data files you've saved, and your history and console logs, and even the little notes you wrote to yourself, but odds are pretty good that no-one else will really understand what's going on (trust me on this: I've been handed lots of random bits of output from people who've been analysing their data, and it makes very little sense unless you've got the original person who did the work sitting right next to you explaining what you're looking at)

Ideally, what you'd like to be able to do is something like this… Suppose you start out with a data set `myrawdata.csv`. What you want is a single document – let's call it `mydataanalysis.R` – that stores all of the commands that you've used in order to do your data analysis. Kind of similar to the R history but much more focused. It would only include the commands that you want to keep for later. Then, later on, instead of typing in all those commands again, you'd just tell R to run all of the commands that are stored in `mydataanalysis.R`. Also, in order to help you make sense of all those commands, what you'd want is the ability to add some notes or *comments* within the file, so that anyone reading the document for themselves would be able to understand what each of the commands actually does. But these comments wouldn't get in the way: when you try to get R to run `mydataanalysis.R` it would be smart enough would recognise that these comments are for the benefit of humans, and so it would ignore them. Later on you could tweak a few of the commands inside the file (maybe in a new file called `mynewdatanalaysis.R`) so that you can adapt an old analysis to be able to handle a new problem. And you could email your friends and colleagues a copy of this file so that they can reproduce your analysis themselves.

In other words, what you want is a *script*.

18.1.2 first script

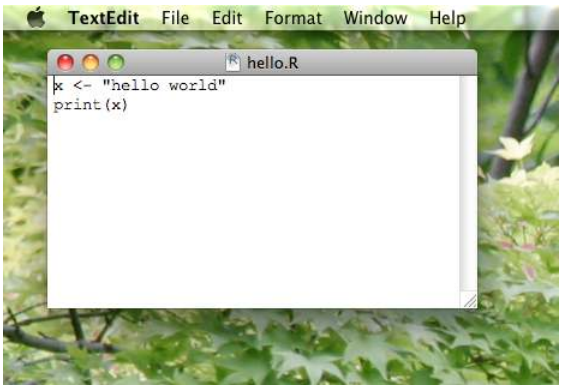


Figure 8.1: A screenshot showing the `hello.R` script if you open in using the default text editor (TextEdit) on a Mac. Using a simple text editor like TextEdit on a Mac or Notepad on Windows isn't actually the best way to write your scripts, but it is the simplest. More to the point, it highlights the fact that a script really is just an ordinary text file.

Okay then. Since scripts are so terribly awesome, let's write one. To do this, open up a simple text editing program, like TextEdit (on a Mac) or Notebook (on Windows). Don't use a fancy word processing program like Microsoft Word or OpenOffice: use the simplest program you can find. Open a new text document, and type some R commands, hitting enter after each command. Let's try using `x <- "hello world"` and `print(x)` as our commands. Then save the document as `hello.R`, and remember to save it as a plain text file: don't save it as a word document or a rich text file. Just a boring old plain text file. Also, when it asks you *where* to save the file, save it to whatever folder you're using as your working directory in R. At this point, you should be looking at something like Figure 8.1. And if so, you have now successfully written your first R program. Because I don't want to take screenshots for every single script, I'm going to present scripts using extracts formatted as follows:

```
## --- hello.R
x <- "hello world"
print(x)
```

The line at the top is the filename, and not part of the script itself. Below that, you can see the two R commands that make up the script itself. Next to each command I've included the line numbers. You don't actually type these into your script, but a lot of text editors (including the one built into Rstudio that I'll show you in a moment) will show line numbers, since it's a very useful convention that allows you to say things like "line 1 of the script creates a new variable, and line 2 prints it out".

So how do we run the script? Assuming that the `hello.R` file has been saved to your working directory, then you can run the script using the following command:

```
source( "hello.R" )
```

If the script file is saved in a different directory, then you need to specify the path to the file, in exactly the same way that you would have to when loading a data file using `load()`. In any case, when you type this command, R opens up the script file: it then reads each command in the file in the same order that they appear in the file, and executes those commands in that order. The simple script that I've shown above contains two commands. The first one creates a variable `x` and the second one prints it on screen. So, when we run the script, this is what we see on screen:

```
source("./rbook-master/scripts/hello.R")
```

```
## [1] "hello world"
```

If we inspect the workspace using a command like `who()` or `objects()`, we discover that R has created the new variable `x` within the workspace, and not surprisingly `x` is a character string containing the text `"hello world"`. And just like that, you've written your first program R. It really is that simple.

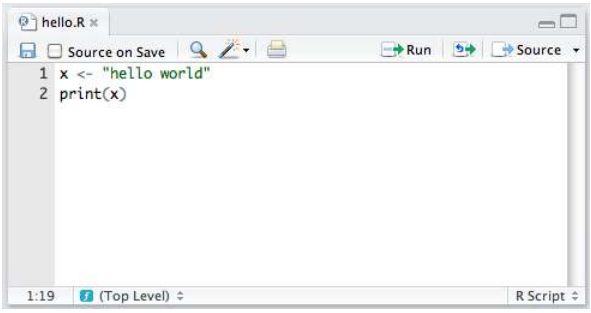


Figure 8.2: A screenshot showing the `hello.R` script open in Rstudio. Assuming that you're looking at this document in colour, you'll notice that the "hello world" text is shown in green. This isn't something that you do yourself: that's Rstudio being helpful. Because the text editor in Rstudio "knows" something about how R commands work, it will highlight different parts of your script in different colours. This is useful, but it's not actually part of the script itself.

18.1.3 Using Rstudio to write scripts

In the example above I assumed that you were writing your scripts using a simple text editor. However, it's usually more convenient to use a text editor that is specifically designed to help you write scripts. There's a lot of these out there, and experienced programmers will all have their own personal favourites. For our purposes, however, we can just use the one built into Rstudio. To create new script file in R studio, go to the "File" menu, select the "New" option, and then click on "R script". This will open a new window within the "source" panel. Then you can type the commands you want (or ***code*** as it is generally called when you're typing the commands into a script file) and save it when you're done. The nice thing about using Rstudio to do this is that it automatically changes the colour of the text to indicate which parts of the code are comments and which are parts are actual R commands (these colours are called ***syntax highlighting***, but they're not actually part of the file – it's just Rstudio trying to be

helpful. To see an example of this, let's open up our `hello.R` script in Rstudio. To do this, go to the "File" menu again, and select "Open…". Once you've opened the file, you should be looking at something like Figure 8.2. As you can see (if you're looking at this book in colour) the character string "hello world" is highlighted in green.

Using Rstudio for your text editor is convenient for other reasons too. Notice in the top right hand corner of Figure 8.2 there's a little button that reads "Source"? If you click on that, Rstudio will construct the relevant `source()` command for you, and send it straight to the R console. So you don't even have to type in the `source()` command, which actually I think is a great thing, because it really bugs me having to type all those extra keystrokes every time I want to run my script. Anyway, Rstudio provide several other convenient little tools to help make scripting easier, but I won't discuss them here.[134]

18.1.4 Commenting your script

When writing up your data analysis as a script, one thing that is generally a good idea is to include a lot of comments in the code. That way, if someone else tries to read it (or if you come back to it several days, weeks, months or years later) they can figure out what's going on. As a beginner, I think it's especially useful to comment thoroughly, partly because it gets you into the habit of commenting the code, and partly because the simple act of typing in an explanation of what the code does will help you keep it clear in your own mind what you're trying to achieve. To illustrate this idea, consider the following script:

```
## --- itngscript.R
# A script to analyse nightgarden.Rdata_
#  author: Dan Navarro_
#  date: 22/11/2011_

# Load the data, and tell the user that this is what we're
# doing.
cat( "loading data from nightgarden.Rdata...\n" )
load( "./rbook-master/data/nightgarden.Rdata" )

# Create a cross tabulation and print it out:
cat( "tabulating data...\n" )
itng.table <- table( speaker, utterance )
print( itng.table )
```

You'll notice that I've gone a bit overboard with my commenting: at the top of the script I've explained the purpose of the script, who wrote it, and when it was written. Then, throughout the script file itself I've added a lot of comments explaining what each section of the code actually does. In real life people don't tend to comment this thoroughly, but the basic idea is a very good one: you really do want your script to explain itself. Nevertheless, as you'd expect R completely ignores all of the commented parts. When we run this script, this is what we see on screen:

```
## --- itngscript.R
# A script to analyse nightgarden.Rdata
#  author: Dan Navarro
#  date: 22/11/2011

# Load the data, and tell the user that this is what we're
# doing.
cat( "loading data from nightgarden.Rdata...\n" )
```

```
## loading data from nightgarden.Rdata...
```

```
load( "./rbook-master/data/nightgarden.Rdata" )

# Create a cross tabulation and print it out:
cat( "tabulating data...\n" )
```

```
## tabulating data...
```

```
itng.table <- table( speaker, utterance )
print( itng.table )
```

```
##              utterance
## speaker       ee onk oo pip
##   makka-pakka  0   2  0   2
##   tombliboo    1   0  1   0
##   upsy-daisy   0   2  0   2
```

Even here, notice that the script announces its behaviour. The first two lines of the output tell us a lot about what the script is actually doing behind the scenes (the code do to this corresponds to the two `cat()` commands on lines 8 and 12 of the script). It's usually a pretty good idea to do this, since it helps ensure that the output makes sense when the script is executed.

18.1.5 Differences between scripts and the command line

For the most part, commands that you insert into a script behave in exactly the same way as they would if you typed the same thing in at the command line. The one major exception to this is that if you want a variable to be printed on screen, you need to explicitly tell R to print it. You can't just type the name of the variable. For example, our original `hello.R` script produced visible output. The following script does not:

```
## --- silenthello.R
x <- "hello world"
x
```

It *does* still create the variable `x` when you `source()` the script, but it won't print anything on screen.

However, apart from the fact that scripts don't use "auto-printing" as it's called, there aren't a lot of differences in the underlying mechanics. There are a few stylistic differences though. For instance, if you want to load a package at the command line, you would generally use the `library()` function. If you want do to it from a script, it's conventional to use `require()` instead. The two commands are basically identical, the only difference being that if the package doesn't exist, `require()` produces a warning whereas `library()` gives you an error. Stylistically, what this means is that if the `require()` command fails in your script, R will boldly continue on and try to execute the rest of the script. Often that's what you'd like to see happen, so it's better to use `require()` . Clearly, however, you can get by just fine using the `library()` command for everyday usage.

18.1.6 Done!

At this point, you've learned the basics of scripting. You are now officially allowed to say that you can program in R, though you probably shouldn't say it too loudly. There's a *lot* more to learn, but nevertheless, if you can write scripts like these then what you are doing is in fact basic programming. The rest of this chapter is devoted to introducing some of the key commands that you need in order to make your programs more powerful; and to help you get used to thinking in terms of scripts, for the rest of this chapter I'll write up most of my extracts as scripts.

This page titled 18.1: Scripts is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **8.1: Scripts** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

18.2: Loops

The description I gave earlier for how a script works was a tiny bit of a lie. Specifically, it's not necessarily the case that R starts at the top of the file and runs straight through to the end of the file. For all the scripts that we've seen so far that's exactly what happens, and unless you insert some commands to explicitly alter how the script runs, that is what will *always* happen. However, you actually have quite a lot of flexibility in this respect. Depending on how you write the script, you can have R repeat several commands, or skip over different commands, and so on. This topic is referred to as ***flow control***, and the first concept to discuss in this respect is the idea of a ***loop***. The basic idea is very simple: a loop is a block of code (i.e., a sequence of commands) that R will execute over and over again until some termination criterion is met. Looping is a very powerful idea. There are three different ways to construct a loop in R, based on the `while` , `for` and `repeat` functions. I'll only discuss the first two in this book.

18.2.1 `while` loop

A `while` loop is a simple thing. The basic format of the loop looks like this:

```
while ( CONDITION ) {
       STATEMENT1
       STATEMENT2
       ETC
    }
```

The code corresponding to CONDITION needs to produce a logical value, either `TRUE` or `FALSE` . Whenever R encounters a `while` statement, it checks to see if the CONDITION is `TRUE` . If it is, then R goes on to execute all of the commands inside the curly brackets, proceeding from top to bottom as usual. However, when it gets to the bottom of those statements, it moves back up to the `while` statement. Then, like the mindless automaton it is, it checks to see if the CONDITION is `TRUE` . If it is, then R goes on to execute all … well, you get the idea. This continues endlessly until at some point the CONDITION turns out to be `FALSE` . Once that happens, R jumps to the bottom of the loop (i.e., to the `}` character), and then continues on with whatever commands appear next in the script.

To start with, let's keep things simple, and use a `while` loop to calculate the smallest multiple of 17 that is greater than or equal to 1000. This is a very silly example since you can actually calculate it using simple arithmetic operations, but the point here isn't to do something novel. The point is to show how to write a `while` loop. Here's the script:

```
## --- whileexample.R
x <- 0
while ( x < 1000 ) {
  x <- x + 17
}
print( x )
```

When we run this script, R starts at the top and creates a new variable called `x` and assigns it a value of 0. It then moves down to the loop, and "notices" that the condition here is `x < 1000` . Since the current value of `x` is zero, the condition is true, so it enters the body of the loop (inside the curly braces). There's only one command here[135] which instructs R to increase the value of `x` by 17. R then returns to the top of the loop, and rechecks the condition. The value of `x` is now 17, but that's still less than 1000, so the loop continues. This cycle will continue for a total of 59 iterations, until finally `x` reaches a value of 1003 (i.e., 59×17=1003). At this point, the loop stops, and R finally reaches line 5 of the script, prints out the value of `x` on screen, and then halts. Let's watch:

```
source( "./rbook-master/scripts/whileexample.R" )
```

```
## [1] 1003
```

Truly fascinating stuff.

18.2.2 `for` loop

The `for` loop is also pretty simple, though not quite as simple as the `while` loop. The basic format of this loop goes like this:

```
 for ( VAR in VECTOR ) {
        STATEMENT1
        STATEMENT2
        ETC
    }
```

In a `for` loop, R runs a fixed number of iterations. We have a VECTOR which has several elements, each one corresponding to a possible value of the variable VAR. In the first iteration of the loop, VAR is given a value corresponding to the first element of VECTOR; in the second iteration of the loop VAR gets a value corresponding to the second value in VECTOR; and so on. Once we've exhausted all of the values in VECTOR, the loop terminates and the flow of the program continues down the script.

Once again, let's use some very simple examples. Firstly, here is a program that just prints out the word "hello" three times and then stops:

```
## --- forexample.R
for ( i in 1:3 ) {
  print( "hello" )
}
```

This is the simplest example of a `for` loop. The vector of possible values for the `i` variable just corresponds to the numbers from 1 to 3. Not only that, the body of the loop doesn't actually depend on `i` at all. Not surprisingly, here's what happens when we run it:

```
source( "./rbook-master/scripts/forexample.R" )
```

```
## [1] "hello"
## [1] "hello"
## [1] "hello"
```

However, there's nothing that stops you from using something non-numeric as the vector of possible values, as the following example illustrates. This time around, we'll use a character vector to control our loop, which in this case will be a vector of `words`. And what we'll do in the loop is get R to convert the word to upper case letters, calculate the length of the word, and print it out. Here's the script:

```
## --- forexample2.R

#the words_
words <- c("it","was","the","dirty","end","of","winter")

#loop over the words_
for ( w in words ) {

  w.length <- nchar( w )     # calculate the number of letters_
  W <- toupper( w )          # convert the word to upper case letters_
  msg <- paste( W, "has", w.length, "letters" )   # a message to print_
  print( msg )               # print it_

}
```

And here's the output:

```
source( "./rbook-master/scripts/forexample2.R" )
```

```
## [1] "IT has 2 letters"
## [1] "WAS has 3 letters"
## [1] "THE has 3 letters"
## [1] "DIRTY has 5 letters"
## [1] "END has 3 letters"
## [1] "OF has 2 letters"
## [1] "WINTER has 6 letters"
```

Again, pretty straightforward I hope.

18.2.3 more realistic example of a loop

To give you a sense of how you can use a loop in a more complex situation, let's write a simple script to simulate the progression of a mortgage. Suppose we have a nice young couple who borrow $300000 from the bank, at an annual interest rate of 5%. The mortgage is a 30 year loan, so they need to pay it off within 360 months total. Our happy couple decide to set their monthly mortgage payment at $1600 per month. Will they pay off the loan in time or not? Only time will tell.[136] Or, alternatively, we could simulate the whole process and get R to tell us. The script to run this is a fair bit more complicated.

```
## --- mortgage.R

# set up
month <- 0        # count the number of months
balance <- 300000 # initial mortgage balance
payments <- 1600  # monthly payments
interest <- 0.05  # 5% interest rate per year
total.paid <- 0   # track what you've paid the bank

# convert annual interest to a monthly multiplier
monthly.multiplier <- (1+interest) ^ (1/12)

# keep looping until the loan is paid off...
while ( balance > 0 ) {

  # do the calculations for this month
  month <- month + 1  # one more month
  balance <- balance * monthly.multiplier  # add the interest
  balance <- balance - payments  # make the payments
  total.paid <- total.paid + payments # track the total paid

  # print the results on screen
  cat( "month", month, ": balance", round(balance), "\n")

} # end of loop

# print the total payments at the end
cat("total payments made", total.paid, "\n" )
```

To explain what's going on, let's go through it carefully. In the first block of code (under `#set up`) all we're doing is specifying all the variables that define the problem. The loan starts with a `balance` of $300,000 owed to the bank on `month` zero, and at that point in time the `total.paid` money is nothing. The couple is making monthly `payments` of $1600, at an annual `interest` rate of 5%. Next, we convert the annual percentage interest into a monthly multiplier. That is, the number that you have to multiply the current balance by each month in order to produce an annual interest rate of 5%. An annual interest rate of 5% implies that, if no payments were made over 12 months the balance would end up being 1.05 times what it was originally, so the *annual* multiplier is 1.05. To calculate the monthly multiplier, we need to calculate the 12th root of 1.05 (i.e., raise 1.05 to the power of 1/12). We store this value in as the `monthly.multiplier` variable, which as it happens corresponds to a value of about 1.004. All of which is a rather long winded way of saying that the *annual* interest rate of 5% corresponds to a *monthly* interest rate of about 0.4%.

Anyway… all of that is really just setting the stage. It's not the interesting part of the script. The interesting part (such as it is) is the loop. The `while` statement on tells R that it needs to keep looping until the `balance` reaches zero (or less, since it might be that the final payment of $1600 pushes the balance below zero). Then, inside the body of the loop, we have two different blocks of code. In the first bit, we do all the number crunching. Firstly we increase the value `month` by 1. Next, the bank charges the interest, so the `balance` goes up. Then, the couple makes their monthly payment and the `balance` goes down. Finally, we keep track of the total amount of money that the couple has paid so far, by adding the `payments` to the running tally. After having done all this number crunching, we tell R to issue the couple with a very terse monthly statement, which just indicates how many months they've been paying the loan and how much money they still owe the bank. Which is rather rude of us really. I've grown attached to this couple and I really feel they deserve better than that. But, that's banks for you.

In any case, the key thing here is the tension between the increase in `balance` on and the decrease. As long as the decrease is bigger, then the balance will eventually drop to zero and the loop will eventually terminate. If not, the loop will continue forever! This is actually very bad programming on my part: I really should have included something to force R to stop if this goes on too long. However, I haven't shown you how to evaluate "if" statements yet, so we'll just have to hope that the author of the book has rigged the example so that the code actually runs. Hm. I wonder what the odds of that are? Anyway, assuming that the loop does eventually terminate, there's one last line of code that prints out the total amount of money that the couple handed over to the bank over the lifetime of the loan.

Now that I've explained everything in the script in tedious detail, let's run it and see what happens:

```
source( "./rbook-master/scripts/mortgage.R" )
```

https://stats.libretexts.org/@go/page/36231

```
## month 1 : balance 299622
## month 2 : balance 299243
## month 3 : balance 298862
## month 4 : balance 298480
## month 5 : balance 298096
## month 6 : balance 297710
## month 7 : balance 297323
## month 8 : balance 296934
## month 9 : balance 296544
## month 10 : balance 296152
## month 11 : balance 295759
## month 12 : balance 295364
## month 13 : balance 294967
## month 14 : balance 294569
## month 15 : balance 294169
## month 16 : balance 293768
## month 17 : balance 293364
## month 18 : balance 292960
## month 19 : balance 292553
## month 20 : balance 292145
## month 21 : balance 291735
## month 22 : balance 291324
## month 23 : balance 290911
## month 24 : balance 290496
## month 25 : balance 290079
## month 26 : balance 289661
## month 27 : balance 289241
## month 28 : balance 288820
## month 29 : balance 288396
## month 30 : balance 287971
## month 31 : balance 287545
## month 32 : balance 287116
## month 33 : balance 286686
## month 34 : balance 286254
## month 35 : balance 285820
## month 36 : balance 285385
## month 37 : balance 284947
## month 38 : balance 284508
## month 39 : balance 284067
## month 40 : balance 283625
## month 41 : balance 283180
## month 42 : balance 282734
## month 43 : balance 282286
## month 44 : balance 281836
## month 45 : balance 281384
## month 46 : balance 280930
## month 47 : balance 280475
## month 48 : balance 280018
## month 49 : balance 279559
## month 50 : balance 279098
```

```
## month 51 : balance 278635
## month 52 : balance 278170
## month 53 : balance 277705
## month 54 : balance 277234
## month 55 : balance 276764
## month 56 : balance 276292
## month 57 : balance 275817
## month 58 : balance 275341
## month 59 : balance 274863
## month 60 : balance 274382
## month 61 : balance 273900
## month 62 : balance 273416
## month 63 : balance 272930
## month 64 : balance 272442
## month 65 : balance 271952
## month 66 : balance 271460
## month 67 : balance 270966
## month 68 : balance 270470
## month 69 : balance 269972
## month 70 : balance 269472
## month 71 : balance 268970
## month 72 : balance 268465
## month 73 : balance 267959
## month 74 : balance 267451
## month 75 : balance 266941
## month 76 : balance 266428
## month 77 : balance 265914
## month 78 : balance 265397
## month 79 : balance 264878
## month 80 : balance 264357
## month 81 : balance 263834
## month 82 : balance 263309
## month 83 : balance 262782
## month 84 : balance 262253
## month 85 : balance 261721
## month 86 : balance 261187
## month 87 : balance 260651
## month 88 : balance 260113
## month 89 : balance 259573
## month 90 : balance 259031
## month 91 : balance 258486
## month 92 : balance 257939
## month 93 : balance 257390
## month 94 : balance 256839
## month 95 : balance 256285
## month 96 : balance 255729
## month 97 : balance 255171
## month 98 : balance 254611
## month 99 : balance 254048
## month 100 : balance 253483
```

So our nice young couple have paid off their $300,000 loan in just 4 months shy of the 30 year term of their loan, at a bargain basement price of $568,046 (since 569600 - 1554 = 568046). A happy ending!

This page titled 18.2.5 Loops is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- 8.2.1 Loops by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

```
## month 101 : balance 252916
## month 102 : balance 252346
## month 103 : balance 251774
## month 104 : balance 251200
## month 105 : balance 250623
## month 106 : balance 250044
## month 107 : balance 249463
## month 108 : balance 248879
## month 109 : balance 248293
## month 110 : balance 247705
## month 111 : balance 247114
## month 112 : balance 246521
## month 113 : balance 245925
## month 114 : balance 245327
## month 115 : balance 244727
## month 116 : balance 244124
## month 117 : balance 243518
## month 118 : balance 242911
## month 119 : balance 242300
## month 120 : balance 241687
## month 121 : balance 241072
## month 122 : balance 240454
## month 123 : balance 239834
## month 124 : balance 239211
## month 125 : balance 238585
## month 126 : balance 237958
## month 127 : balance 237327
## month 128 : balance 236694
## month 129 : balance 236058
## month 130 : balance 235420
## month 131 : balance 234779
## month 132 : balance 234136
## month 133 : balance 233489
## month 134 : balance 232841
## month 135 : balance 232189
## month 136 : balance 231535
## month 137 : balance 230879
## month 138 : balance 230219
## month 139 : balance 229557
## month 140 : balance 228892
## month 141 : balance 228225
## month 142 : balance 227555
## month 143 : balance 226882
## month 144 : balance 226206
## month 145 : balance 225528
## month 146 : balance 224847
## month 147 : balance 224163
## month 148 : balance 223476
## month 149 : balance 222786
## month 150 : balance 222094
## month 151 : balance 221399
```


https://stats.libretexts.org/@go/page/36231

```
## month 151 : balance 221399
## month 152 : balance 220701
## month 153 : balance 220000
## month 154 : balance 219296
## month 155 : balance 218590
## month 156 : balance 217880
## month 157 : balance 217168
## month 158 : balance 216453
## month 159 : balance 215735
## month 160 : balance 215014
## month 161 : balance 214290
## month 162 : balance 213563
## month 163 : balance 212833
## month 164 : balance 212100
## month 165 : balance 211364
## month 166 : balance 210625
## month 167 : balance 209883
## month 168 : balance 209138
## month 169 : balance 208390
## month 170 : balance 207639
## month 171 : balance 206885
## month 172 : balance 206128
## month 173 : balance 205368
## month 174 : balance 204605
## month 175 : balance 203838
## month 176 : balance 203069
## month 177 : balance 202296
## month 178 : balance 201520
## month 179 : balance 200741
## month 180 : balance 199959
## month 181 : balance 199174
## month 182 : balance 198385
## month 183 : balance 197593
## month 184 : balance 196798
## month 185 : balance 196000
## month 186 : balance 195199
## month 187 : balance 194394
## month 188 : balance 193586
## month 189 : balance 192775
## month 190 : balance 191960
## month 191 : balance 191142
## month 192 : balance 190321
## month 193 : balance 189496
## month 194 : balance 188668
## month 195 : balance 187837
## month 196 : balance 187002
## month 197 : balance 186164
## month 198 : balance 185323
## month 199 : balance 184478
## month 200 : balance 183629
## month 201 : balance 182777
```

```
## month 202 : balance 181922
## month 203 : balance 181063
## month 204 : balance 180201
## month 205 : balance 179335
## month 206 : balance 178466
## month 207 : balance 177593
## month 208 : balance 176716
## month 209 : balance 175836
## month 210 : balance 174953
## month 211 : balance 174065
## month 212 : balance 173175
## month 213 : balance 172280
## month 214 : balance 171382
## month 215 : balance 170480
## month 216 : balance 169575
## month 217 : balance 168666
## month 218 : balance 167753
## month 219 : balance 166836
## month 220 : balance 165916
## month 221 : balance 164992
## month 222 : balance 164064
## month 223 : balance 163133
## month 224 : balance 162197
## month 225 : balance 161258
## month 226 : balance 160315
## month 227 : balance 159368
## month 228 : balance 158417
## month 229 : balance 157463
## month 230 : balance 156504
## month 231 : balance 155542
## month 232 : balance 154576
## month 233 : balance 153605
## month 234 : balance 152631
## month 235 : balance 151653
## month 236 : balance 150671
## month 237 : balance 149685
## month 238 : balance 148695
## month 239 : balance 147700
## month 240 : balance 146702
## month 241 : balance 145700
## month 242 : balance 144693
## month 243 : balance 143683
## month 244 : balance 142668
## month 245 : balance 141650
## month 246 : balance 140627
## month 247 : balance 139600
## month 248 : balance 138568
## month 249 : balance 137533
## month 250 : balance 136493
## month 251 : balance 135449
```


https://stats.libretexts.org/@go/page/36231

```
## month 252 : balance 134401
## month 253 : balance 133349
## month 254 : balance 132292
## month 255 : balance 131231
## month 256 : balance 130166
## month 257 : balance 129096
## month 258 : balance 128022
## month 259 : balance 126943
## month 260 : balance 125861
## month 261 : balance 124773
## month 262 : balance 123682
## month 263 : balance 122586
## month 264 : balance 121485
## month 265 : balance 120380
## month 266 : balance 119270
## month 267 : balance 118156
## month 268 : balance 117038
## month 269 : balance 115915
## month 270 : balance 114787
## month 271 : balance 113654
## month 272 : balance 112518
## month 273 : balance 111376
## month 274 : balance 110230
## month 275 : balance 109079
## month 276 : balance 107923
## month 277 : balance 106763
## month 278 : balance 105598
## month 279 : balance 104428
## month 280 : balance 103254
## month 281 : balance 102074
## month 282 : balance 100890
## month 283 : balance 99701
## month 284 : balance 98507
## month 285 : balance 97309
## month 286 : balance 96105
## month 287 : balance 94897
## month 288 : balance 93683
## month 289 : balance 92465
## month 290 : balance 91242
## month 291 : balance 90013
## month 292 : balance 88780
## month 293 : balance 87542
## month 294 : balance 86298
## month 295 : balance 85050
## month 296 : balance 83797
## month 297 : balance 82538
## month 298 : balance 81274
## month 299 : balance 80005
## month 300 : balance 78731
## month 301 : balance 77452
```

```
## month 302 : balance 76168
## month 303 : balance 74878
## month 304 : balance 73583
## month 305 : balance 72283
## month 306 : balance 70977
## month 307 : balance 69666
## month 308 : balance 68350
## month 309 : balance 67029
## month 310 : balance 65702
## month 311 : balance 64369
## month 312 : balance 63032
## month 313 : balance 61688
## month 314 : balance 60340
## month 315 : balance 58986
## month 316 : balance 57626
## month 317 : balance 56261
## month 318 : balance 54890
## month 319 : balance 53514
## month 320 : balance 52132
## month 321 : balance 50744
## month 322 : balance 49351
## month 323 : balance 47952
## month 324 : balance 46547
## month 325 : balance 45137
## month 326 : balance 43721
## month 327 : balance 42299
## month 328 : balance 40871
## month 329 : balance 39438
## month 330 : balance 37998
## month 331 : balance 36553
## month 332 : balance 35102
## month 333 : balance 33645
## month 334 : balance 32182
## month 335 : balance 30713
## month 336 : balance 29238
## month 337 : balance 27758
## month 338 : balance 26271
## month 339 : balance 24778
## month 340 : balance 23279
## month 341 : balance 21773
## month 342 : balance 20262
## month 343 : balance 18745
## month 344 : balance 17221
## month 345 : balance 15691
## month 346 : balance 14155
## month 347 : balance 12613
## month 348 : balance 11064
## month 349 : balance 9509
## month 350 : balance 7948
## month 351 : balance 6380
```

```
## month 352 : balance 4806
## month 353 : balance 3226
## month 354 : balance 1639
## month 355 : balance 46
## month 356 : balance -1554
## total payments made 569600
```

18.3: Conditional Statements

A second kind of flow control that programming languages provide is the ability to evaluate ***conditional statements***. Unlike loops, which can repeat over and over again, a conditional statement only executes once, but it can switch between different possible commands depending on a CONDITION that is specified by the programmer. The power of these commands is that they allow the program itself to make choices, and in particular, to make different choices depending on the context in which the program is run. The most prominent of example of a conditional statement is the `if` statement, and the accompanying `else` statement. The basic format of an `if` statement in R is as follows:

```
if ( CONDITION ) {
        STATEMENT1
        STATEMENT2
        ETC
    }
```

And the execution of the statement is pretty straightforward. If the CONDITION is true, then R will execute the statements contained in the curly braces. If the CONDITION is false, then it dose not. If you want to, you can extend the `if` statement to include an `else` statement as well, leading to the following syntax:

```
 if ( CONDITION ) {
        STATEMENT1
        STATEMENT2
        ETC
    } else {
        STATEMENT3
        STATEMENT4
        ETC
    }
```

As you'd expect, the interpretation of this version is similar. If the CONDITION is true, then the contents of the first block of code (i.e., STATEMENT1, STATEMENT2, ETC) are executed; but if it is false, then the contents of the second block of code (i.e., STATEMENT3, STATEMENT4, ETC) are executed instead.

To give you a feel for how you can use `if` and `else` to do something useful, the example that I'll show you is a script that prints out a different message depending on what day of the week you run it. We can do this making use of some of the tools that we discussed in Section 7.11.3. Here's the script:

```
## --- ifelseexample.R
# find out what day it is...
today <- Sys.Date()       # pull the date from the system clock
day <- weekdays( today )  # what day of the week it is_

# now make a choice depending on the day...
if ( day == "Monday" ) {
  print( "I don't like Mondays" )
} else {
  print( "I'm a happy little automaton" )
}
```

```
## [1] "I'm a happy little automaton"
```

Since today happens to be a Sunday, when I run the script here's what happens:

```
source( "./rbook-master/scripts/ifelseexample.R" )
```

```
## [1] "I'm a happy little automaton"
```

There are other ways of making conditional statements in R. In particular, the `ifelse()` function and the `switch()` functions can be very useful in different contexts. However, my main aim in this chapter is to briefly cover the very basics, so I'll move on.

This page titled 18.3: Conditional Statements is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **8.3: Conditional Statements** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

18.4: Writing Functions

In this section I want to talk about functions again. Functions were introduced in Section 3.5, but you've learned a lot about R since then, so we can talk about them in more detail. In particular, I want to show you how to create your own. To stick with the same basic framework that I used to describe loops and conditionals, here's the syntax that you use to create a function:

```
  FNAME <- function ( ARG1, ARG2, ETC ) {
         STATEMENT1
         STATEMENT2
         ETC
         return( VALUE )
      }
```

What this does is create a function with the name FNAME, which has arguments ARG1, ARG2 and so forth. Whenever the function is called, R executes the statements in the curly braces, and then outputs the contents of VALUE to the user. Note, however, that R does not execute the commands inside the function in the workspace. Instead, what it does is create a temporary local environment: all the internal statements in the body of the function are executed there, so they remain invisible to the user. Only the final results in the VALUE are returned to the workspace.

To give a simple example of this, let's create a function called `quadruple()` which multiplies its inputs by four. In keeping with the approach taken in the rest of the chapter, I'll use a script to do this:

```
## --- functionexample.R
quadruple <- function(x) {
  y <- x*4
  return(y)
}
```

When we run this script, as follows

```
source( "./rbook-master/scripts/functionexample.R" )
```

nothing appears to have happened, but there is a new object created in the workspace called `quadruple`. Not surprisingly, if we ask R to tell us what kind of object it is, it tells us that it is a function:

```
class( quadruple )
```

```
## [1] "function"
```

And now that we've created the `quadruple()` function, we can call it just like any other function And if I want to store the output as a variable, I can do this:

```
my.var <- quadruple(10)
print(my.var)
```

```
## [1] 40
```

An important thing to recognise here is that the two internal variables that the `quadruple()` function makes use of, `x` and `y`, stay internal. That is, if we inspect the contents of the workspace,

```
library(lsr)
```

```
## Warning: package 'lsr' was built under R version 3.5.2
```

```
who()
```

```
##    -- Name --           -- Class --   -- Size --
##    balance              numeric       1
##    day                  character     1
##    i                    integer       1
##    interest             numeric       1
##    itng.table           table         3 x 4
##    month                numeric       1
##    monthly.multiplier   numeric       1
##    msg                  character     1
##    my.var               numeric       1
##    payments             numeric       1
##    quadruple            function
##    speaker              character     10
##    today                Date          1
##    total.paid           numeric       1
##    utterance            character     10
##    w                    character     1
##    W                    character     1
##    w.length             integer       1
##    words                character     7
##    x                    numeric       1
```

we see everything in our workspace from this chapter including the `quadruple()` function itself, as well as the `my.var` variable that we just created.

Now that we know how to create our own functions in R, it's probably a good idea to talk a little more about some of the other properties of functions that I've been glossing over. To start with, let's take this opportunity to type the name of the function at the command line without the parentheses:

```
quadruple
```

```
## function (x)
## {
##     y <- x * 4
##     return(y)
## }
```

As you can see, when you type the name of a function at the command line, R prints out the underlying source code that we used to define the function in the first place. In the case of the `quadruple()` function, this is quite helpful to us – we can read this code and actually see what the function does. For other functions, this is less helpful, as we saw back in Section 3.5 when we tried typing `citation` rather than `citation()` .

18.4.1 Function arguments revisited

Okay, now that we are starting to get a sense for how functions are constructed, let's have a look at two, slightly more complicated functions that I've created. The source code for these functions is contained within the `functionexample2.R` and `functionexample3.R` scripts. Let's start by looking at the first one:

```
## --- functionexample2.R
pow <- function( x, y = 1) {
  out <- x^y  # raise x to the power y
  return( out )
}
```

and if we type `source("functionexample2.R")` to load the `pow()` function into our workspace, then we can make use of it. As you can see from looking at the code for this function, it has two arguments `x` and `y`, and all it does is raise `x` to the power of `y`. For instance, this command

```
pow(x=3, y=2)
```

```
## [1] 9
```

calculates the value of 3^2. The interesting thing about this function isn't what it does, since R already has has perfectly good mechanisms for calculating powers. Rather, notice that when I defined the function, I specified `y=1` when listing the arguments? That's the default value for `y`. So if we enter a command without specifying a value for `y`, then the function assumes that we want `y=1`:

```
pow( x=3 )
```

```
## [1] 3
```

However, since I didn't specify any default value for `x` when I defined the `pow()` function, we always need to input a value for `x`. If we don't R will spit out an error message.

So now you know how to specify default values for an argument. The other thing I should point out while I'm on this topic is the use of the `...` argument. The `...` argument is a special construct in R which is only used within functions. It is used as a way of matching against multiple user inputs: in other words, `...` is used as a mechanism to allow the user to enter as many inputs as they like. I won't talk at all about the low-level details of how this works at all, but I will show you a simple example of a function that makes use of it. To that end, consider the following script:

```
## --- functionexample3.R
doubleMax <- function( ... ) {
  max.val <- max( ... )   # find the largest value in ...
  out <- 2 * max.val      # double it
  return( out )
}
```

When we type `source("functionexample3.R")`, R creates the `doubleMax()` function. You can type in as many inputs as you like. The `doubleMax()` function identifies the largest value in the inputs, by passing all the user inputs to the `max()` function, and then doubles it. For example:

```
doubleMax( 1,2,5 )
```

```
## [1] 10
```

18.4.2 There's more to functions than this

There's a lot of other details to functions that I've hidden in my description in this chapter. Experienced programmers will wonder exactly how the "scoping rules" work in R,[137] or want to know how to use a function to create variables in other environments[138], or if function objects can be assigned as elements of a list[139] and probably hundreds of other things besides. However, I don't want to have this discussion get too cluttered with details, so I think it's best – at least for the purposes of the current book – to stop here.

This page titled 18.4: Writing Functions is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **8.4: Writing Functions** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

18.5: Implicit Loops

There's one last topic I want to discuss in this chapter. In addition to providing the explicit looping structures via `while` and `for`, R also provides a collection of functions for ***implicit loops***. What I mean by this is that these are functions that carry out operations very similar to those that you'd normally use a loop for. However, instead of typing out the whole loop, the whole thing is done with a single command. The main reason why this can be handy is that – due to the way that R is written – these implicit looping functions are usually about to do the same calculations much faster than the corresponding explicit loops. In most applications that beginners might want to undertake, this probably isn't very important, since most beginners tend to start out working with fairly small data sets and don't usually need to undertake extremely time consuming number crunching. However, because you often see these functions referred to in other contexts, it may be useful to very briefly discuss a few of them.

The first and simplest of these functions is `sapply()`. The two most important arguments to this function are `X`, which specifies a vector containing the data, and `FUN`, which specifies the name of a function that should be applied to each element of the data vector. The following example illustrates the basics of how it works:

```
words <- c("along", "the", "loom", "of", "the", "land")
sapply( X = words, FUN = nchar )
```

```
## along   the  loom    of   the  land
##     5     3     4     2     3     4
```

Notice how similar this is to the second example of a `for` loop in Section 8.2.2. The `sapply()` function has implicitly looped over the elements of `words`, and for each such element applied the `nchar()` function to calculate the number of letters in the corresponding word.

The second of these functions is `tapply()`, which has three key arguments. As before `X` specifies the data, and `FUN` specifies a function. However, there is also an `INDEX` argument which specifies a grouping variable.[140] What the `tapply()` function does is loop over all of the different values that appear in the `INDEX` variable. Each such value defines a group: the `tapply()` function constructs the subset of `X` that corresponds to that group, and then applies the function `FUN` to that subset of the data. This probably sounds a little abstract, so let's consider a specific example, using the `nightgarden.Rdata` file that we used in Chapter 7.

```
gender <- c( "male","male","female","female","male" )
age <- c( 10,12,9,11,13 )
tapply( X = age, INDEX = gender, FUN = mean )
```

```
##   female     male
## 10.00000 11.66667
```

In this extract, what we're doing is using `gender` to define two different groups of people, and using their `ages` as the data. We then calculate the `mean()` of the ages, separately for the males and the females. A closely related function is `by()`. It actually does the same thing as `tapply()`, but the output is formatted a bit differently. This time around the three arguments are called `data`, `INDICES` and `FUN`, but they're pretty much the same thing. An example of how to use the `by()` function is shown in the following extract:

```
by( data = age, INDICES = gender, FUN = mean )
```

```
## gender: female
## [1] 10
## ------------------------------------------------------------
## gender: male
## [1] 11.66667
```

The `tapply()` and `by()` functions are quite handy things to know about, and are pretty widely used. However, although I do make passing reference to the `tapply()` later on, I don't make much use of them in this book.

Before moving on, I should mention that there are several other functions that work along similar lines, and have suspiciously similar names: `lapply`, `mapply`, `apply`, `vapply`, `rapply` and `eapply`. However, none of these come up anywhere else in this book, so all I wanted to do here is draw your attention to the fact that they exist.

This page titled 18.5: Implicit Loops is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **8.5: Implicit Loops** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

18.6: Summary

In this chapter I talked about several key programming concepts, things that you should know about if you want to start converting your simple scripts into full fledged programs:

- Writing and using scripts (Section 8.1).
- Using loops (Section 8.2) and implicit loops (Section 8.5).
- Making conditional statements (Section 8.3)
- Writing your own functions (Section 8.4)

As always, there are *lots* of things I'm ignoring in this chapter. It takes a lot of work to become a proper programmer, just as it takes a lot of work to be a proper psychologist or a proper statistician, and this book is certainly not going to provide you with all the tools you need to make that step. However, you'd be amazed at how much you can achieve using only the tools that I've covered up to this point. Loops, conditionals and functions are very powerful things, especially when combined with the various tools discussed in Chapters 3, 4 and 7. Believe it or not, you're off to a pretty good start just by having made it to this point. If you want to keep going, there are (as always!) several other books you might want to look at. One that I've read and enjoyed is "A first course in statistical programming with R" Braun and Murdoch (2007), but quite a few people have suggested to me that "The art of programming with R" Matloff and Matloff (2011) is worth the effort too.

References

Braun, John, and Duncan J Murdoch. 2007. *A First Course in Statistical Programming with R*. Cambridge University Press Cambridge.

Matloff, Norman, and Norman S Matloff. 2011. *The Art of R Programming: A Tour of Statistical Software Design*. No Starch Press.

---

133. The quote comes from *Count Zero* (1986)
134. Okay, I lied. Sue me. One of the coolest features of Rstudio is the support for *R Markdown*, which lets you embed R code inside a Markdown document, and you can automatically publish your R Markdown to the web on Rstudio's servers. If you're the kind of nerd interested in this sort of thing, it's really nice. And, yes, since I'm also that kind of nerd, of course I'm aware that iPython notebooks do the same thing and that R just nicked their idea. So what? It's still cool. And anyway, this book isn't called *Learning Statistics with Python* now, is it? Hm. Maybe I should write a Python version…
135. As an aside: if there's only a single command that you want to include inside your loop, then you don't actually need to bother including the curly braces at all. However, until you're comfortable programming in R I'd advise *always* using them, even when you don't have to.
136. Okay, fine. This example is still a bit ridiculous, in three respects. Firstly, the bank absolutely will not let the couple pay less than the amount required to terminate the loan in 30 years. Secondly, a constant interest rate of 30 years is hilarious. Thirdly, you can solve this much more efficiently than through brute force simulation. However, we're not exactly in the business of being realistic or efficient here.
137. Lexical scope.
138. The `assign()` function.
139. Yes.
140. Or a list of such variables.

This page titled 18.6: Summary is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **8.6: Summary** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

CHAPTER OVERVIEW

19: Bayesian Statistics

> *In our reasonings concerning matter of fact, there are all imaginable degrees of assurance, from the highest certainty to the lowest species of moral evidence. A wise man, therefore, proportions his belief to the evidence. – David Hume*[253].

The ideas I've presented to you in this book describe inferential statistics from the frequentist perspective. I'm not alone in doing this. In fact, almost every textbook given to undergraduate psychology students presents the opinions of the frequentist statistician as *the* theory of inferential statistics, the one true way to do things. I have taught this way for practical reasons. The frequentist view of statistics dominated the academic field of statistics for most of the 20th century, and this dominance is even more extreme among applied scientists. It was and is current practice among psychologists to use frequentist methods. Because frequentist methods are ubiquitous in scientific papers, every student of statistics needs to understand those methods, otherwise they will be unable to make sense of what those papers are saying! Unfortunately – in my opinion at least – the current practice in psychology is often misguided, and the reliance on frequentist methods is partly to blame. In this chapter I explain why I think this, and provide an introduction to Bayesian statistics, an approach that I think is generally superior to the orthodox approach.

This chapter comes in two parts. In Sections 17.1 through 17.3 I talk about what Bayesian statistics are all about, covering the basic mathematical rules for how it works as well as an explanation for why I think the Bayesian approach is so useful. Afterwards, I provide a brief overview of how you can do Bayesian versions of chi-square tests (Section 17.6), t-tests (Section 17.7), regression (Section 17.8) and ANOVA (Section 17.9).

19.1: Probabilistic Reasoning by Rational Agents
19.2: Bayesian Hypothesis Tests
19.3: Why Be a Bayesian?
19.4: Evidentiary Standards You Can Believe
19.5: The p-value Is a Lie.
19.6: Bayesian Analysis of Contingency Tables
19.7: Bayesian t-tests
19.8: Bayesian Regression
19.9: Bayesian ANOVA
19.10: Summary

This page titled 19: Bayesian Statistics is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

19.1: Probabilistic Reasoning by Rational Agents

From a Bayesian perspective, statistical inference is all about *belief revision*. I start out with a set of candidate hypotheses h about the world. I don't know which of these hypotheses is true, but do I have some beliefs about which hypotheses are plausible and which are not. When I observe the data d, I have to revise those beliefs. If the data are consistent with a hypothesis, my belief in that hypothesis is strengthened. If the data inconsistent with the hypothesis, my belief in that hypothesis is weakened. That's it! At the end of this section I'll give a precise description of how Bayesian reasoning works, but first I want to work through a simple example in order to introduce the key ideas. Consider the following reasoning problem:

I'm carrying an umbrella. Do you think it will rain?

In this problem, I have presented you with a single piece of data (d= I'm carrying the umbrella), and I'm asking you to tell me your beliefs about whether it's raining. You have two possible ***hypotheses***, h: either it rains today or it does not. How should you solve this problem?

19.1.1 Priors: what you believed before

The first thing you need to do ignore what I told you about the umbrella, and write down your pre-existing beliefs about rain. This is important: if you want to be honest about how your beliefs have been revised in the light of new evidence, then you *must* say something about what you believed before those data appeared! So, what might you believe about whether it will rain today? You probably know that I live in Australia, and that much of Australia is hot and dry. And in fact you're right: the city of Adelaide where I live has a Mediterranean climate, very similar to southern California, southern Europe or northern Africa. I'm writing this in January, and so you can assume it's the middle of summer. In fact, you might have decided to take a quick look on Wikipedia[254] and discovered that Adelaide gets an average of 4.4 days of rain across the 31 days of January. Without knowing anything else, you might conclude that the probability of January rain in Adelaide is about 15%, and the probability of a dry day is 85%. If this is really what you believe about Adelaide rainfall (and now that I've told it to you, I'm betting that this really *is* what you believe) then what I have written here is your ***prior distribution***, written P(h):

Hypothesis	Degree of Belief
Rainy day	0.15
Dry day	0.85

19.1.2 Likelihoods: theories about the data

To solve the reasoning problem, you need a theory about my behaviour. When does Dan carry an umbrella? You might guess that I'm not a complete idiot,[255] and I try to carry umbrellas only on rainy days. On the other hand, you also know that I have young kids, and you wouldn't be all that surprised to know that I'm pretty forgetful about this sort of thing. Let's suppose that on rainy days I remember my umbrella about 30% of the time (I really am awful at this). But let's say that on dry days I'm only about 5% likely to be carrying an umbrella. So you might write out a little table like this:

Hypothesis	Umbrella	No umbrella
Rainy day	0.30	0.70
Dry day	0.05	0.95

It's important to remember that each cell in this table describes your beliefs about what data d will be observed, *given* the truth of a particular hypothesis h. This "conditional probability" is written P(d|h), which you can read as "the probability of d given h". In Bayesian statistics, this is referred to as ***likelihood*** of data d given hypothesis h.[256]

19.1.3 joint probability of data and hypothesis

At this point, all the elements are in place. Having written down the priors and the likelihood, you have all the information you need to do Bayesian reasoning. The question now becomes, *how* do we use this information? As it turns out, there's a very simple

equation that we can use here, but it's important that you understand why we use it, so I'm going to try to build it up from more basic ideas.

Let's start out with one of the rules of probability theory. I listed it way back in Table 9.1, but I didn't make a big deal out of it at the time and you probably ignored it. The rule in question is the one that talks about the probability that *two* things are true. In our example, you might want to calculate the probability that today is rainy (i.e., hypothesis h is true) *and* I'm carrying an umbrella (i.e., data d is observed). The ***joint probability*** of the hypothesis and the data is written P(d,h), and you can calculate it by multiplying the prior P(h) by the likelihood P(d|h). Mathematically, we say that:

$$P(d,h)=P(d|h)P(h)$$

So, what is the probability that today is a rainy day *and* I remember to carry an umbrella? As we discussed earlier, the prior tells us that the probability of a rainy day is 15%, and the likelihood tells us that the probability of me remembering my umbrella on a rainy day is 30%. So the probability that both of these things are true is calculated by multiplying the two:

$$\begin{aligned}(\text{rainy, umbrella}) &= P(\text{ umbrella }|\text{rainy}) \times P(\text{ rainy }) \\ &= 0.30 \times 0.15 \\ &= 0.045\end{aligned}$$

In other words, before being told anything about what actually happened, you think that there is a 4.5% probability that today will be a rainy day and that I will remember an umbrella. However, there are of course *four* possible things that could happen, right? So let's repeat the exercise for all four. If we do that, we end up with the following table:

	Umbrella	No-umbrella
Rainy	0.045	0.105
Dry	0.0425	0.8075

This table captures all the information about which of the four possibilities are likely. To really get the full picture, though, it helps to add the row totals and column totals. That gives us this table:

	Umbrella	No-umbrella	Total
Rainy	0.0450	0.1050	0.15
Dry	0.0425	0.8075	0.85
Total	0.0875	0.9125	1

This is a very useful table, so it's worth taking a moment to think about what all these numbers are telling us. First, notice that the row sums aren't telling us anything new at all. For example, the first row tells us that if we ignore all this umbrella business, the chance that today will be a rainy day is 15%. That's not surprising, of course: that's our prior. The important thing isn't the number itself: rather, the important thing is that it gives us some confidence that our calculations are sensible! Now take a look at the column sums, and notice that they tell us something that we haven't explicitly stated yet. In the same way that the row sums tell us the probability of rain, the column sums tell us the probability of me carrying an umbrella. Specifically, the first column tells us that on average (i.e., ignoring whether it's a rainy day or not), the probability of me carrying an umbrella is 8.75%. Finally, notice that when we sum across all four logically-possible events, everything adds up to 1. In other words, what we have written down is a proper probability distribution defined over all possible combinations of data and hypothesis.

Now, because this table is so useful, I want to make sure you understand what all the elements correspond to, and how they written:

	Umbrella	No-umbrella	
Rainy	P(Umbrella, Rainy)	P(No-umbrella, Rainy)	P(Rainy)
Dry	P(Umbrella, Dry)	P(No-umbrella, Dry)	P(Dry)
	P(Umbrella)	P(No-umbrella)	

Finally, let's use "proper" statistical notation. In the rainy day problem, the data corresponds to the observation that I do or do not have an umbrella. So we'll let d_1 refer to the possibility that you observe me carrying an umbrella, and d_2 refers to you observing me not carrying one. Similarly, h_1 is your hypothesis that today is rainy, and h_2 is the hypothesis that it is not. Using this notation, the table looks like this:

19.1.4 Updating beliefs using Bayes' rule

The table we laid out in the last section is a very powerful tool for solving the rainy day problem, because it considers all four logical possibilities and states exactly how confident you are in each of them before being given any data. It's now time to consider what happens to our beliefs when we are actually given the data. In the rainy day problem, you are told that I really *am* carrying an umbrella. This is something of a surprising event: according to our table, the probability of me carrying an umbrella is only 8.75%. But that makes sense, right? A guy carrying an umbrella on a summer day in a hot dry city is pretty unusual, and so you really weren't expecting that. Nevertheless, the problem tells you that it is true. No matter how unlikely you thought it was, you must now adjust your beliefs to accommodate the fact that you now *know* that I have an umbrella.[257] To reflect this new knowledge, our *revised* table must have the following numbers:

	Umbrella	No-umbrella
Rainy		0
Dry		0
Total	1	0

In other words, the facts have eliminated any possibility of "no umbrella", so we have to put zeros into any cell in the table that implies that I'm not carrying an umbrella. Also, you know for a fact that I am carrying an umbrella, so the column sum on the left must be 1 to correctly describe the fact that P(umbrella)=1.

What two numbers should we put in the empty cells? Again, let's not worry about the maths, and instead think about our intuitions. When we wrote out our table the first time, it turned out that those two cells had almost identical numbers, right? We worked out that the joint probability of "rain and umbrella" was 4.5%, and the joint probability of "dry and umbrella" was 4.25%. In other words, before I told you that I am in fact carrying an umbrella, you'd have said that these two events were almost identical in probability, yes? But notice that *both* of these possibilities are consistent with the fact that I actually am carrying an umbrella. From the perspective of these two possibilities, very little has changed. I hope you'd agree that it's *still* true that these two possibilities are equally plausible. So what we expect to see in our final table is some numbers that preserve the fact that "rain and umbrella" is *slightly* more plausible than "dry and umbrella", while still ensuring that numbers in the table add up. Something like this, perhaps?

	Umbrella	No-umbrella
Rainy	0.514	0
Dry	0.486	0
Total	1	0

What this table is telling you is that, after being told that I'm carrying an umbrella, you believe that there's a 51.4% chance that today will be a rainy day, and a 48.6% chance that it won't. That's the answer to our problem! The ***posterior probability*** of rain P(h|d) given that I am carrying an umbrella is 51.4%

How did I calculate these numbers? You can probably guess. To work out that there was a 0.514 probability of "rain", all I did was take the 0.045 probability of "rain and umbrella" and divide it by the 0.0875 chance of "umbrella". This produces a table that satisfies our need to have everything sum to 1, and our need not to interfere with the relative plausibility of the two events that are actually consistent with the data. To say the same thing using fancy statistical jargon, what I've done here is divide the joint probability of the hypothesis and the data P(d,h) by the ***marginal probability*** of the data P(d), and this is what gives us the posterior probability of the hypothesis *given* that we know the data have been observed. To write this as an equation:[258]

$$P(h|d) = \frac{P(d,h)}{P(d)}$$

However, remember what I said at the start of the last section, namely that the joint probability P(d,h) is calculated by multiplying the prior P(h) by the likelihood P(d|h). In real life, the things we actually know how to write down are the priors and the likelihood, so let's substitute those back into the equation. This gives us the following formula for the posterior probability:

$$P(h|d) = \frac{P(d|h)P(h)}{P(d)}$$

And this formula, folks, is known as ***Bayes' rule***. It describes how a learner starts out with prior beliefs about the plausibility of different hypotheses, and tells you how those beliefs should be revised in the face of data. In the Bayesian paradigm, all statistical inference flows from this one simple rule.

This page titled 19.1: Probabilistic Reasoning by Rational Agents is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **17.1: Probabilistic Reasoning by Rational Agents** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

19.2: Bayesian Hypothesis Tests

In Chapter 11 I described the orthodox approach to hypothesis testing. It took an entire chapter to describe, because null hypothesis testing is a very elaborate contraption that people find very hard to make sense of. In contrast, the Bayesian approach to hypothesis testing is incredibly simple. Let's pick a setting that is closely analogous to the orthodox scenario. There are two hypotheses that we want to compare, a null hypothesis h_0 and an alternative hypothesis h_1. Prior to running the experiment we have some beliefs P(h) about which hypotheses are true. We run an experiment and obtain data d. Unlike frequentist statistics Bayesian statistics does allow to talk about the probability that the null hypothesis is true. Better yet, it allows us to calculate the ***posterior probability of the null hypothesis***, using Bayes' rule:

$$P(h_0|d) = \frac{P(d|h_0)P(h_0)}{P(d)}$$

This formula tells us exactly how much belief we should have in the null hypothesis after having observed the data d. Similarly, we can work out how much belief to place in the alternative hypothesis using essentially the same equation. All we do is change the subscript:

$$P(h_1|d) = \frac{P(d|h_1)P(h_1)}{P(d)}$$

It's all so simple that I feel like an idiot even bothering to write these equations down, since all I'm doing is copying Bayes rule from the previous section.[259]

19.2.1 Bayes factor

In practice, most Bayesian data analysts tend not to talk in terms of the raw posterior probabilities $P(h_0|d)$ and $P(h_1|d)$. Instead, we tend to talk in terms of the ***posterior odds*** ratio. Think of it like betting. Suppose, for instance, the posterior probability of the null hypothesis is 25%, and the posterior probability of the alternative is 75%. The alternative hypothesis is three times as probable as the null, so we say that the *odds* are 3:1 in favour of the alternative. Mathematically, all we have to do to calculate the posterior odds is divide one posterior probability by the other:

$$\frac{P(h_1|d)}{P(h_0|d)} = \frac{0.75}{0.25} = 3$$

Or, to write the same thing in terms of the equations above:

$$\frac{P(h_1|d)}{P(h_0|d)} = \frac{P(d|h_1)}{P(d|h_0)} \times \frac{P(h_1)}{P(h_0)}$$

Actually, this equation is worth expanding on. There are three different terms here that you should know. On the left hand side, we have the posterior odds, which tells you what you believe about the relative plausibilty of the null hypothesis and the alternative hypothesis *after* seeing the data. On the right hand side, we have the ***prior odds***, which indicates what you thought *before* seeing the data. In the middle, we have the ***Bayes factor***, which describes the amount of evidence provided by the data:

$$\underbrace{\frac{P(h_1|d)}{P(h_0|d)}}_{\text{Posterior odds}} = \underbrace{\frac{P(d|h_1)}{P(d|h_0)}}_{\text{Bayes factor}} \times \underbrace{\frac{P(h_1)}{P(h_0)}}_{\text{Prior odds}}$$

The Bayes factor (sometimes abbreviated as ***BF***) has a special place in the Bayesian hypothesis testing, because it serves a similar role to the p-value in orthodox hypothesis testing: it quantifies the strength of evidence provided by the data, and as such it is the Bayes factor that people tend to report when running a Bayesian hypothesis test. The reason for reporting Bayes factors rather than posterior odds is that different researchers will have different priors. Some people might have a strong bias to believe the null hypothesis is true, others might have a strong bias to believe it is false. Because of this, the polite thing for an applied researcher to do is report the Bayes factor. That way, anyone reading the paper can multiply the Bayes factor by their own *personal* prior odds, and they can work out for themselves what the posterior odds would be. In any case, by convention we like to pretend that we give equal consideration to both the null hypothesis and the alternative, in which case the prior odds equals 1, and the posterior odds becomes the same as the Bayes factor.

19.2.2 Interpreting Bayes factors

One of the really nice things about the Bayes factor is the numbers are inherently meaningful. If you run an experiment and you compute a Bayes factor of 4, it means that the evidence provided by your data corresponds to betting odds of 4:1 in favour of the alternative. However, there have been some attempts to quantify the standards of evidence that would be considered meaningful in a scientific context. The two most widely used are from Jeffreys (1961) and Kass and Raftery (1995). Of the two, I tend to prefer the Kass and Raftery (1995) table because it's a bit more conservative. So here it is:

Bayes factor	Interpretation
1 - 3	Negligible evidence
3 - 20	Positive evidence
20 - 150	Strong evidence
$>$150	Very strong evidence

And to be perfectly honest, I think that even the Kass and Raftery standards are being a bit charitable. If it were up to me, I'd have called the "positive evidence" category "weak evidence". To me, anything in the range 3:1 to 20:1 is "weak" or "modest" evidence at best. But there are no hard and fast rules here: what counts as strong or weak evidence depends entirely on how conservative you are, and upon the standards that your community insists upon before it is willing to label a finding as "true".

In any case, note that all the numbers listed above make sense if the Bayes factor is greater than 1 (i.e., the evidence favours the alternative hypothesis). However, one big practical advantage of the Bayesian approach relative to the orthodox approach is that it also allows you to quantify evidence *for* the null. When that happens, the Bayes factor will be less than 1. You can choose to report a Bayes factor less than 1, but to be honest I find it confusing. For example, suppose that the likelihood of the data under the null hypothesis $P(d|h_0)$ is equal to 0.2, and the corresponding likelihood $P(d|h_0)$ under the alternative hypothesis is 0.1. Using the equations given above, Bayes factor here would be:

$$BF = \frac{P(d|h_1)}{P(d|h_0} = \frac{0.1}{0.2} = 0.5$$

Read literally, this result tells is that the evidence in favour of the alternative is 0.5 to 1. I find this hard to understand. To me, it makes a lot more sense to turn the equation "upside down", and report the amount op evidence in favour of the *null*. In other words, what we calculate is this:

$$BF' = \frac{P(d|h_0)}{P(d|h_1)} = \frac{0.2}{0.1} = 2$$

And what we would report is a Bayes factor of 2:1 in favour of the null. Much easier to understand, and you can interpret this using the table above.

This page titled 19.2: Bayesian Hypothesis Tests is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **17.2: Bayesian Hypothesis Tests** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

19.3: Why Be a Bayesian?

Up to this point I've focused exclusively on the logic underpinning Bayesian statistics. We've talked about the idea of "probability as a degree of belief", and what it implies about how a rational agent should reason about the world. The question that you have to answer for yourself is this: how do *you* want to do your statistics? Do you want to be an orthodox statistician, relying on sampling distributions and p-values to guide your decisions? Or do you want to be a Bayesian, relying on Bayes factors and the rules for rational belief revision? And to be perfectly honest, I can't answer this question for you. Ultimately it depends on what you think is right. It's your call, and your call alone. That being said, I can talk a little about why *I* prefer the Bayesian approach.

19.3.1 Statistics that mean what you think they mean

> *You keep using that word. I do not think it means what you think it means*
> *– Inigo Montoya, The Princess Bride*[260]

To me, one of the biggest advantages to the Bayesian approach is that it answers the right questions. Within the Bayesian framework, it is perfectly sensible and allowable to refer to "the probability that a hypothesis is true". You can even try to calculate this probability. Ultimately, isn't that what you *want* your statistical tests to tell you? To an actual human being, this would seem to be the whole *point* of doing statistics: to determine what is true and what isn't. Any time that you aren't exactly sure about what the truth is, you should use the language of probability theory to say things like "there is an 80% chance that Theory A is true, but a 20% chance that Theory B is true instead".

This seems so obvious to a human, yet it is explicitly forbidden within the orthodox framework. To a frequentist, such statements are a nonsense because "the theory is true" is not a repeatable event. A theory is true or it is not, and no probabilistic statements are allowed, no matter how much you might want to make them. There's a reason why, back in Section 11.5, I repeatedly warned you *not* to interpret the p-value as the probability of that the null hypothesis is true. There's a reason why almost every textbook on statstics is forced to repeat that warning. It's because people desperately *want* that to be the correct interpretation. Frequentist dogma notwithstanding, a lifetime of experience of teaching undergraduates and of doing data analysis on a daily basis suggests to me that most actual humans thing that "the probability that the hypothesis is true" is not only meaningful, it's the thing we care *most* about. It's such an appealing idea that even trained statisticians fall prey to the mistake of trying to interpret a p-value this way. For example, here is a quote from an official Newspoll report in 2013, explaining how to interpret their (frequentist) data analysis:[261]

Throughout the report, where relevant, statistically significant changes have been noted. All significance tests have been based on the 95 percent level of confidence. ***This means that if a change is noted as being statistically significant, there is a 95 percent probability that a real change has occurred****, and is not simply due to chance variation. (emphasis added)*

Nope! That's *not* what p<.05 means. That's *not* what 95% confidence means to a frequentist statistician. The bolded section is just plain wrong. Orthodox methods cannot tell you that "there is a 95% chance that a real change has occurred", because this is not the kind of event to which frequentist probabilities may be assigned. To an ideological frequentist, this sentence should be meaningless. Even if you're a more pragmatic frequentist, it's still the wrong definition of a p-value. It is simply not an allowed or correct thing to say if you want to rely on orthodox statistical tools.

On the other hand, let's suppose you are a Bayesian. Although the bolded passage is the wrong definition of a p-value, it's pretty much exactly what a Bayesian means when they say that the posterior probability of the alternative hypothesis is greater than 95%. And here's the thing. If the Bayesian posterior is actually thing you *want* to report, why are you even trying to use orthodox methods? If you want to make Bayesian claims, all you have to do is be a Bayesian and use Bayesian tools.

Speaking for myself, I found this to be a the most liberating thing about switching to the Bayesian view. Once you've made the jump, you no longer have to wrap your head around counterinuitive definitions of p-values. You don't have to bother remembering why you can't say that you're 95% confident that the true mean lies within some interval. All you have to do is be honest about what you believed before you ran the study, and then report what you learned from doing it. Sounds nice, doesn't it? To me, this is the big promise of the Bayesian approach: you do the analysis you really want to do, and express what you really believe the data are telling you.

This page titled 19.3: Why Be a Bayesian? is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **17.3: Why Be a Bayesian?** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

19.4: Evidentiary Standards You Can Believe

> *If [p] is below .02 it is strongly indicated that the [null] hypothesis fails to account for the whole of the facts. We shall not often be astray if we draw a conventional line at .05 and consider that [smaller values of p] indicate a real discrepancy.*
> *– Sir Ronald Fisher (1925)*

Consider the quote above by Sir Ronald Fisher, one of the founders of what has become the orthodox approach to statistics. If anyone has ever been entitled to express an opinion about the intended function of p-values, it's Fisher. In this passage, taken from his classic guide *Statistical Methods for Research Workers*, he's pretty clear about what it means to reject a null hypothesis at p<.05. In his opinion, if we take p<.05 to mean there is "a real effect", then "we shall not often be astray". This view is hardly unusual: in my experience, most practitioners express views very similar to Fisher's. In essence, the p<.05 convention is assumed to represent a fairly stringent evidentiary standard.

Well, how true is that? One way to approach this question is to try to convert p-values to Bayes factors, and see how the two compare. It's not an easy thing to do because a p-value is a fundamentally different kind of calculation to a Bayes factor, and they don't measure the same thing. However, there have been some attempts to work out the relationship between the two, and it's somewhat surprising. For example, Johnson (2013) presents a pretty compelling case that (for t-tests at least) the p<.05 threshold corresponds roughly to a Bayes factor of somewhere between 3:1 and 5:1 in favour of the alternative. If that's right, then Fisher's claim is a bit of a stretch. Let's suppose that the null hypothesis is true about half the time (i.e., the prior probability of H_0 is 0.5), and we use those numbers to work out the posterior probability of the null hypothesis given that it has been rejected at p<.05. Using the data from Johnson (2013), we see that if you reject the null at p<.05, you'll be correct about 80% of the time. I don't know about you, but in my opinion an evidentiary standard that ensures you'll be wrong on 20% of your decisions isn't good enough. The fact remains that, quite contrary to Fisher's claim, if you reject at p<.05 you shall quite often go astray. It's not a very stringent evidentiary threshold at all.

This page titled 19.4: Evidentiary Standards You Can Believe is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **17.4: Evidentiary Standards You Can Believe** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

19.5: The p-value Is a Lie.

> *The cake is a lie.*
> *The cake is a lie.*
> *The cake is a lie.*
> *The cake is a lie.*
> *– Portal*[262]

Okay, at this point you might be thinking that the real problem is not with orthodox statistics, just the p<.05 standard. In one sense, that's true. The recommendation that Johnson (2013) gives is not that "everyone must be a Bayesian now". Instead, the suggestion is that it would be wiser to shift the conventional standard to something like a p<.01 level. That's not an unreasonable view to take, but in my view the problem is a little more severe than that. In my opinion, there's a fairly big problem built into the way most (but not all) orthodox hypothesis tests are constructed. They are grossly naive about how humans actually do research, and because of this most p-values are wrong.

Sounds like an absurd claim, right? Well, consider the following scenario. You've come up with a really exciting research hypothesis and you design a study to test it. You're very diligent, so you run a power analysis to work out what your sample size should be, and you run the study. You run your hypothesis test and out pops a p-value of 0.072. Really bloody annoying, right?

What should you do? Here are some possibilities:

1. You conclude that there is no effect, and try to publish it as a null result
2. You guess that there might be an effect, and try to publish it as a "borderline significant" result
3. You give up and try a new study
4. You collect some more data to see if the p value goes up or (preferably!) drops below the "magic" criterion of p<.05

Which would *you* choose? Before reading any further, I urge you to take some time to think about it. Be honest with yourself. But don't stress about it too much, because you're screwed no matter what you choose. Based on my own experiences as an author, reviewer and editor, as well as stories I've heard from others, here's what will happen in each case:

- Let's start with option 1. If you try to publish it as a null result, the paper will struggle to be published. Some reviewers will think that p=.072 is not really a null result. They'll argue it's borderline significant. Other reviewers will agree it's a null result, but will claim that even though some null results *are* publishable, yours isn't. One or two reviewers might even be on your side, but you'll be fighting an uphill battle to get it through.
- Okay, let's think about option number 2. Suppose you try to publish it as a borderline significant result. Some reviewers will claim that it's a null result and should not be published. Others will claim that the evidence is ambiguous, and that you should collect more data until you get a clear significant result. Again, the publication process does not favour you.
- Given the difficulties in publishing an "ambiguous" result like p=.072, option number 3 might seem tempting: give up and do something else. But that's a recipe for career suicide. If you give up and try a new project else every time you find yourself faced with ambiguity, your work will never be published. And if you're in academia without a publication record you can lose your job. So that option is out.
- It looks like you're stuck with option 4. You don't have conclusive results, so you decide to collect some more data and re-run the analysis. Seems sensible, but unfortunately for you, if you do this all of your p-values are now incorrect. *All* of them. Not just the p-values that you calculated for *this* study. All of them. All the p-values you calculated in the past and all the p-values you will calculate in the future. Fortunately, no-one will notice. You'll get published, and you'll have lied.

Wait, what? How can that last part be true? I mean, it sounds like a perfectly reasonable strategy doesn't it? You collected some data, the results weren't conclusive, so now what you want to do is collect more data until the the results *are* conclusive. What's wrong with that?

Honestly, there's nothing wrong with it. It's a reasonable, sensible and rational thing to do. In real life, this is exactly what every researcher does. Unfortunately, the theory of null hypothesis testing as I described it in Chapter 11 *forbids* you from doing this.[263] The reason is that the theory assumes that the experiment is finished and all the data are in. And because it assumes the experiment is over, it only considers *two* possible decisions. If you're using the conventional p<.05 threshold, those decisions are:

Outcome	Action
p less than .05	Reject the null
p greater than .05	Retain the null

What *you're* doing is adding a third possible action to the decision making problem. Specifically, what you're doing is using the p-value itself as a reason to justify continuing the experiment. And as a consequence you've transformed the decision-making procedure into one that looks more like this:

Outcome	Action
p less than .05	Stop the experiment and reject the null
p between .05 and .1	Continue the experiment
p greater than .1	Stop the experiment and retain the null

The "basic" theory of null hypothesis testing isn't built to handle this sort of thing, not in the form I described back in Chapter 11. If you're the kind of person who would choose to "collect more data" in real life, it implies that you are *not* making decisions in accordance with the rules of null hypothesis testing. Even if you happen to arrive at the same decision as the hypothesis test, you aren't following the decision *process* it implies, and it's this failure to follow the process that is causing the problem.[264] Your p-values are a lie.

Worse yet, they're a lie in a dangerous way, because they're all *too small*. To give you a sense of just how bad it can be, consider the following (worst case) scenario. Imagine you're a really super-enthusiastic researcher on a tight budget who didn't pay any attention to my warnings above. You design a study comparing two groups. You desperately want to see a significant result at the $p<.05$ level, but you really don't want to collect any more data than you have to (because it's expensive). In order to cut costs, you start collecting data, but every time a new observation arrives you run a t-test on your data. If the t-tests says $p<.05$ then you stop the experiment and report a significant result. If not, you keep collecting data. You keep doing this until you reach your pre-defined spending limit for this experiment. Let's say that limit kicks in at N=1000 observations. As it turns out, the truth of the matter is that there is no real effect to be found: the null hypothesis is true. So, what's the chance that you'll make it to the end of the experiment and (correctly) conclude that there is no effect? In an ideal world, the answer here should be 95%. After all, the whole *point* of the $p<.05$ criterion is to control the Type I error rate at 5%, so what we'd hope is that there's only a 5% chance of falsely rejecting the null hypothesis in this situation. However, there's no guarantee that will be true. You're breaking the rules: you're running tests repeatedly, "peeking" at your data to see if you've gotten a significant result, and all bets are off.

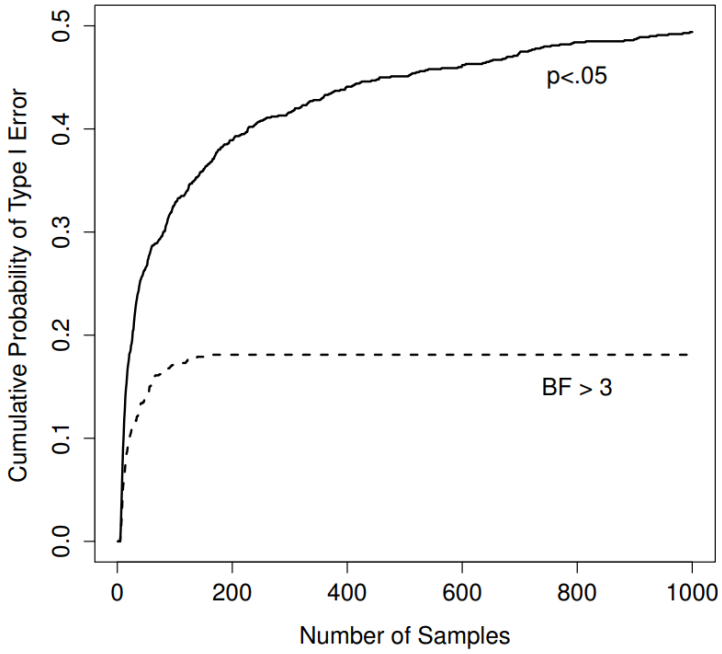


Figure 17.1: How badly can things go wrong if you re-run your tests every time new data arrive? If you are a frequentist, the answer is "very wrong".

So how bad is it? The answer is shown as the solid black line in Figure 17.1, and it's *astoundingly* bad. If you peek at your data after every single observation, there is a 49% chance that you will make a Type I error. That's, um, quite a bit bigger than the 5% that it's supposed to be. By way of comparison, imagine that you had used the following strategy. Start collecting data. Every single time an observation arrives, run a *Bayesian* t-test (Section 17.7 and look at the Bayes factor. I'll assume that Johnson (2013) is right, and I'll treat a Bayes factor of 3:1 as roughly equivalent to a p-value of .05.[265] This time around, our trigger happy researcher uses the following procedure: if the Bayes factor is 3:1 or more in favour of the null, stop the experiment and retain the null. If it is 3:1 or more in favour of the alternative, stop the experiment and reject the null. Otherwise continue testing. Now, just like last time, let's assume that the null hypothesis is true. What happens? As it happens, I ran the simulations for this scenario too, and the results are shown as the dashed line in Figure 17.1. It turns out that the Type I error rate is much much lower than the 49% rate that we were getting by using the orthodox t-test.

In some ways, this is remarkable. The entire *point* of orthodox null hypothesis testing is to control the Type I error rate. Bayesian methods aren't actually designed to do this at all. Yet, as it turns out, when faced with a "trigger happy" researcher who keeps running hypothesis tests as the data come in, the Bayesian approach is much more effective. Even the 3:1 standard, which most Bayesians would consider unacceptably lax, is much safer than the p<.05 rule.

19.5.1 really this bad?

The example I gave in the previous section is a pretty extreme situation. In real life, people don't run hypothesis tests every time a new observation arrives. So it's not fair to say that the p<.05 threshold "really" corresponds to a 49% Type I error rate (i.e., p=.49). But the fact remains that if you want your p-values to be honest, then you either have to switch to a completely different way of doing hypothesis tests, or you must enforce a strict rule: *no peeking*. You are *not* allowed to use the data to decide when to terminate the experiment. You are *not* allowed to look at a "borderline" p-value and decide to collect more data. You aren't even allowed to change your data analyis strategy after looking at data. You are strictly required to follow these rules, otherwise the p-values you calculate will be nonsense.

And yes, these rules are surprisingly strict. As a class exercise a couple of years back, I asked students to think about this scenario. Suppose you started running your study with the intention of collecting N=80 people. When the study starts out you follow the rules, refusing to look at the data or run any tests. But when you reach N=50 your willpower gives in… and you take a peek. Guess what? You've got a significant result! Now, sure, you know you *said* that you'd keep running the study out to a sample size of N=80, but it seems sort of pointless now, right? The result is significant with a sample size of N=50, so wouldn't it be wasteful and

inefficient to keep collecting data? Aren't you tempted to stop? Just a little? Well, keep in mind that if you do, your Type I error rate at p<.05 just ballooned out to 8%. When you report p<.05 in your paper, what you're *really* saying is p<.08. That's how bad the consequences of "just one peek" can be.

Now consider this … the scientific literature is filled with t-tests, ANOVAs, regressions and chi-square tests. When I wrote this book I didn't pick these tests arbitrarily. The reason why these four tools appear in most introductory statistics texts is that these are the bread and butter tools of science. None of these tools include a correction to deal with "data peeking": they all assume that you're not doing it. But how realistic is that assumption? In real life, how many people do you think have "peeked" at their data before the experiment was finished and adapted their subsequent behaviour after seeing what the data looked like? Except when the sampling procedure is fixed by an external constraint, I'm guessing the answer is "most people have done it". If that has happened, you can infer that the reported p-values are wrong. Worse yet, because we don't know what decision process they actually followed, we have no way to know what the p-values *should* have been. You can't compute a p-value when you don't know the decision making procedure that the researcher used. And so the reported p-value remains a lie.

Given all of the above, what is the take home message? It's not that Bayesian methods are foolproof. If a researcher is determined to cheat, they can always do so. Bayes' rule cannot stop people from lying, nor can it stop them from rigging an experiment. That's not my point here. My point is the same one I made at the very beginning of the book in Section 1.1: the reason why we run statistical tests is to protect us from ourselves. And the reason why "data peeking" is such a concern is that it's so tempting, *even for honest researchers*. A theory for statistical inference has to acknowledge this. Yes, you might try to defend p-values by saying that it's the fault of the researcher for not using them properly. But to my mind that misses the point. A theory of statistical inference that is so completely naive about humans that it doesn't even consider the possibility that the researcher might *look at their own data* isn't a theory worth having. In essence, my point is this:

> *Good laws have their origins in bad morals.*
> *– Ambrosius Macrobius*[266]

Good rules for statistical testing have to acknowledge human frailty. None of us are without sin. None of us are beyond temptation. A good system for statistical inference should still work even when it is used by actual humans. Orthodox null hypothesis testing does not.[267]

This page titled 19.5: The p-value Is a Lie. is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **17.5: The p-value Is a Lie.** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

19.6: Bayesian Analysis of Contingency Tables

Time to change gears. Up to this point I've been talking about what Bayesian inference is and why you might consider using it. I now want to briefly describe how to do Bayesian versions of various statistical tests. The discussions in the next few sections are not as detailed as I'd like, but I hope they're enough to help you get started. So let's begin.

The first kind of statistical inference problem I discussed in this book appeared in Chapter 12, in which we discussed categorical data analysis problems. In that chapter I talked about several different statistical problems that you might be interested in, but the one that appears most often in real life is the analysis of *contingency tables*. In this kind of data analysis situation, we have a cross-tabulation of one variable against another one, and the goal is to find out if there is some *association* between these variables. The data set I used to illustrate this problem is found in the `chapek9.Rdata` file, and it contains a single data frame `chapek9`

```
load("./rbook-master/data/chapek9.Rdata")
head(chapek9)
```

```
##   species choice
## 1   robot flower
## 2   human   data
## 3   human   data
## 4   human   data
## 5   robot   data
## 6   human flower
```

In this data set, we supposedly sampled 180 beings and measured two things. First, we checked whether they were humans or robots, as captured by the `species` variable. Second, we asked them to nominate whether they most preferred flowers, puppies, or data. When we produce the cross-tabulation, we get this as the results:

```
crosstab <- xtabs( ~ species + choice, chapek9 )
crosstab
```

```
##         choice
## species puppy flower data
##   robot    13     30   44
##   human    15     13   65
```

Surprisingly, the humans seemed to show a much stronger preference for data than the robots did. At the time we speculated that this might have been because the questioner was a large robot carrying a gun, and the humans might have been scared.

19.6.1 orthodox text

Just to refresh your memory, here's how we analysed these data back in Chapter@refch:chisquare. Because we want to determine if there is some *association* between `species` and `choice` , we used the `associationTest()` function in the `lsr` package to run a chi-square test of association. The results looked like this:

```
library(lsr)
```

```
## Warning: package 'lsr' was built under R version 3.5.2
```

```
associationTest( ~species + choice, chapek9 )
```

```
##
##      Chi-square test of categorical association
##
## Variables:   species, choice
##
## Hypotheses:
##    null:        variables are independent of one another
##    alternative: some contingency exists between variables
##
## Observed contingency table:
##        choice
## species puppy flower data
##   robot    13     30   44
##   human    15     13   65
##
## Expected contingency table under the null hypothesis:
##        choice
## species puppy flower data
##   robot  13.5   20.8 52.7
##   human  14.5   22.2 56.3
##
## Test results:
##    X-squared statistic:  10.722
##    degrees of freedom:  2
##    p-value:  0.005
##
## Other information:
##    estimated effect size (Cramer's v):  0.244
```

Because we found a small p value (in this case p<.01), we concluded that the data are inconsistent with the null hypothesis of no association, and we rejected it.

19.6.2 Bayesian test

How do we run an equivalent test as a Bayesian? Well, like every other bloody thing in statistics, there's a lot of different ways you *could* do it. However, for the sake of everyone's sanity, throughout this chapter I've decided to rely on one R package to do the work. Specifically, I'm going to use the `BayesFactor` package written by Jeff Rouder and Rich Morey, which as of this writing is in version 0.9.10.

For the analysis of contingency tables, the `BayesFactor` package contains a function called `contingencyTableBF()`. The data that you need to give to this function is the contingency table itself (i.e., the `crosstab` variable above), so you might be expecting to use a command like this:

```
library( BayesFactor )           # ...because we have to load the package
contingencyTableBF( crosstab )   # ...because that makes sense, right?
```

However, if you try this you'll get an error message. This is because the `contingencyTestBF()` function needs one other piece of information from you: it needs to know what *sampling plan* you used to run your experiment. You can specify the sampling plan using the `sampleType` argument. So I should probably tell you what your options are! The `contingencyTableBF()` function distinguishes between four different types of experiment:

- **Fixed sample size**. Suppose that in our `chapek9` example, our experiment was designed like this: we deliberately set out to test 180 people, but we didn't try to control the number of humans or robots, nor did we try to control the choices they made. In this design, the total number of observations N is fixed, but everything else is random. This is referred to as "joint multinomial" sampling, and if that's what you did you should specify `sampleType = "jointMulti"`. In the case of the `chapek9` data, that's actually what I had in mind when I invented the data set.
- **Fixed row (or column) totals**. A different kind of design might work like this. We decide ahead of time that we want 180 people, but we try to be a little more systematic about it. Specifically, the *experimenter* constrains it so that we get a predetermined number of humans and robots (e.g., 90 of each). In this design, *either* the row totals or the column totals are fixed, but not both. This is referred to as "independent multinomial" sampling, and if that's what you did you should specify `sampleType = "indepMulti"`.
- **Both row and column totals fixed**. Another logical possibility is that you designed the experiment so that *both* the row totals and the column totals are fixed. This doesn't make any sense at all in the `chapek9` example, but there are other deisgns that can work this way. Suppose that I show you a collection of 20 toys, and then given them 10 stickers that say `boy` and another 10 that say `girl`. I then give them 10 `blue` stickers and 10 `pink` stickers. I then ask you to put the stickers on the 20 toys such that every toy has a colour and every toy has a gender. No matter how you assign the stickers, the total number of pink and blue toys will be 10, as will the number of boys and girls. In this design *both* the rows and columns of the contingency table are fixed. This is referred to as "hypergeometric" sampling, and if that's what you've done you should specify `sampleType = "hypergeom"`.
- **Nothing is fixed**. Finally, it might be the case that *nothing* is fixed. Not the row columns, not the column totals, and not the total sample size either. For instance, in the `chapek9` scenario, suppose what I'd done is run the study for a fixed length of *time*. By chance, it turned out that I got 180 people to turn up to study, but it could easily have been something else. This is referred to as "Poisson" sampling, and if that's what you've done you should specify `sampleType="poisson"`.

Okay, so now we have enough knowledge to actually run a test. For the `chapek9` data, I implied that we designed the study such that the total sample size N was fixed, so we should set `sampleType = "jointMulti"`. The command that we need is,

```
library( BayesFactor )
```

```
## Warning: package 'BayesFactor' was built under R version 3.5.2
```

```
## Loading required package: coda
```

```
## Warning: package 'coda' was built under R version 3.5.2
```

```
## Loading required package: Matrix
```

```
## ************
## Welcome to BayesFactor 0.9.12-4.2. If you have questions, please contact Richard M
##
## Type BFManual() to open the manual.
## ************
```

```
contingencyTableBF( crosstab, sampleType = "jointMulti" )
```

```
## Bayes factor analysis
## --------------
## [1] Non-indep. (a=1) : 15.92684 ±0%
##
## Against denominator:
##   Null, independence, a = 1
## ---
## Bayes factor type: BFcontingencyTable, joint multinomial
```

As with most R commands, the output initially looks suspiciously similar to utter gibberish. Fortunately, it's actually pretty simple once you get past the initial impression. Firstly, note that the stuff at the top and bottom are irrelevant fluff. You already know that you're doing a Bayes factor analysis. You already know that you're analysing a contingency table, and you already know that you specified a joint multinomial sampling plan. So let's strip that out and take a look at what's left over:

```
[1] Non-indep. (a=1) : 15.92684 @plusorminus0%

Against denominator:
  Null, independence, a = 1
```

Let's also ignore those two `a=1` bits, since they're technical details that you don't need to know about at this stage.[268] The rest of the output is actually pretty straightforward. At the bottom, the output defines the null hypothesis for you: in this case, the null hypothesis is that there is no relationship between `species` and `choice`. Or, to put it another way, the null hypothesis is that these two variables are *independent*. Now if you look at the line above it, you might (correctly) guess that the `Non-indep.` part refers to the *alternative* hypothesis. In this case, the alternative is that there *is* a relationship between `species` and `choice`: that is, they are not independent. So the only thing left in the output is the bit that reads

```
15.92684 @plusorminus0%
```

The 15.9 part is the Bayes factor, and it's telling you that the odds for the alternative hypothesis against the null are about 16:1. The ±0% part is not very interesting: essentially, all it's telling you is that R has calculated an exact Bayes factor, so the uncertainty about the Bayes factor is 0%.[269] In any case, the data are telling us that we have moderate evidence for the alternative hypothesis.

19.6.3 Writing up the results

When writing up the results, my experience has been that there aren't quite so many "rules" for how you "should" report Bayesian hypothesis tests. That might change in the future if Bayesian methods become standard and some task force starts writing up style guides, but in the meantime I would suggest using some common sense. For example, I would avoid writing this:

A Bayesian test of association found a significant result (BF=15.92)

To my mind, this write up is unclear. Even assuming that you've already reported the relevant descriptive statistics, there are a number of things I am unhappy with. First, the concept of "statistical significance" is pretty closely tied with p-values, so it reads slightly strangely. Second, the "BF=15.92" part will only make sense to people who already understand Bayesian methods, and not everyone does. Third, it is somewhat unclear exactly which test was run and what software was used to do so.

On the other hand, unless precision is *extremely* important, I think that this is taking things a step too far:

We ran a Bayesian test of association using version 0.9.10-1 of the BayesFactor package using default priors and a joint multinomial sampling plan. The resulting Bayes factor of 15.92 to 1 in favour of the alternative hypothesis indicates that there is moderately strong evidence for the non-independence of species and choice.

Everything about that passage is correct, of course. Morey and Rouder (2015) built their Bayesian tests of association using the paper by Gunel and Dickey (1974), the specific test we used assumes that the experiment relied on a joint multinomial sampling plan, and indeed the Bayes factor of 15.92 is moderately strong evidence. It's just far too wordy.

In most situations you just don't need that much information. My preference is usually to go for something a little briefer. First, if you're reporting multiple Bayes factor analyses in your write up, then somewhere you only need to cite the software once, at the beginning of the results section. So you might have one sentence like this:

All analyses were conducted using the BayesFactor package in R , and unless otherwise stated default parameter values were used

Notice that I don't bother including the version number? That's because the citation itself includes that information (go check my reference list if you don't believe me). There's no need to clutter up your results with redundant information that almost no-one will actually need. When you get to the actual test you can get away with this:

A test of association produced a Bayes factor of 16:1 in favour of a relationship between species and choice.

Short and sweet. I've rounded 15.92 to 16, because there's not really any important difference between 15.92:1 and 16:1. I spelled out "Bayes factor" rather than truncating it to "BF" because not everyone knows the abbreviation. I indicated exactly what the effect is (i.e., "a relationship between species and choice") and how strong the evidence was. I *didn't* bother indicating whether this was "moderate" evidence or "strong" evidence, because the odds themselves tell you! There's nothing stopping you from including that information, and I've done so myself on occasions, but you don't strictly need it. Similarly, I didn't bother to indicate that I ran the "joint multinomial" sampling plan, because I'm assuming that the method section of my write up would make clear how the experiment was designed. (I might change my mind about that if the method section was ambiguous.) Neither did I bother indicating that this was a *Bayesian* test of association: if your reader can't work that out from the fact that you're reporting a Bayes factor and the fact that you're citing the `BayesFactor` package for all your analyses, then there's no chance they'll understand anything you've written. Besides, if you keep writing the word "Bayes" over and over again it starts to look stupid. Bayes Bayes Bayes Bayes Bayes. See?

19.6.4 Other sampling plans

Up to this point all I've shown you is how to use the `contingencyTableBF()` function for the joint multinomial sampling plan (i.e., when the total sample size N is fixed, but nothing else is). For the Poisson sampling plan (i.e., nothing fixed), the command you need is identical except for the `sampleType` argument:

```
contingencyTableBF(crosstab, sampleType = "poisson" )
```

```
## Bayes factor analysis
## --------------
## [1] Non-indep. (a=1) : 28.20757 ±0%
##
## Against denominator:
##   Null, independence, a = 1
## ---
## Bayes factor type: BFcontingencyTable, poisson
```

Notice that the Bayes factor of 28:1 here is *not* the identical to the Bayes factor of 16:1 that we obtained from the last test. The sampling plan actually does matter.

What about the design in which the row columns (or column totals) are fixed? As I mentioned earlier, this corresponds to the "independent multinomial" sampling plan. Again, you need to specify the `sampleType` argument, but this time you need to specify whether you fixed the rows or the columns. For example, suppose I deliberately sampled 87 humans and 93 robots, then I would need to indicate that the `fixedMargin` of the contingency table is the `"rows"` . So the command I would use is:

```
contingencyTableBF(crosstab, sampleType = "indepMulti", fixedMargin="rows")
```

```
## Bayes factor analysis
## --------------
## [1] Non-indep. (a=1) : 8.605897 ±0%
##
## Against denominator:
##   Null, independence, a = 1
## ---
## Bayes factor type: BFcontingencyTable, independent multinomial
```

Again, the Bayes factor is different, with the evidence for the alternative dropping to a mere 9:1. As you might expect, the answers would be diffrent again if it were the columns of the contingency table that the experimental design fixed.

Finally, if we turn to hypergeometric sampling in which everything is fixed, we get…

```
contingencyTableBF(crosstab, sampleType = "hypergeom")
#Error in contingencyHypergeometric(as.matrix(data2), a) :
#  hypergeometric contingency tables restricted to 2 x 2 tables; see help for conting
```

… an error message. Okay, some quick reading through the help files hints that support for larger contingency tables is coming, but it's not been implemented yet. In the meantime, let's imagine we have data from the "toy labelling" experiment I described earlier in this section. Specifically, let's say our data look like this:

```
toys <- data.frame(stringsAsFactors=FALSE,
        gender = c("girl", "boy"),
        pink = c(8, 2),
        blue = c(2, 8)
        )
```

The Bayesian test with hypergeometric sampling gives us this:

```
contingencyTableBF(toys, sampleType = "hypergeom")

#Bayes factor analysis
#--------------
#[1] Non-indep. (a=1) : 8.294321 @plusorminus0%
#
#Against denominator:
#  Null, independence, a = 1
#---
#Bayes factor type: BFcontingencyTable, hypergeometric
```

The Bayes factor of 8:1 provides modest evidence that the labels were being assigned in a way that correlates gender with colour, but it's not conclusive.

This page titled 19.6: Bayesian Analysis of Contingency Tables is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **17.6: Bayesian Analysis of Contingency Tables** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

19.7: Bayesian t-tests

The second type of statistical inference problem discussed in this book is the comparison between two means, discussed in some detail in the chapter on t-tests (Chapter 13. If you can remember back that far, you'll recall that there are several versions of the t-test. The `BayesFactor` package contains a function called `ttestBF()` that is flexible enough to run several different versions of the t-test. I'll talk a little about Bayesian versions of the independent samples t-tests and the paired samples t-test in this section.

19.7.1 Independent samples t-test

The most common type of t-test is the independent samples t-test, and it arises when you have data that look something like this:

```
load( "./rbook-master/data/harpo.Rdata" )
head(harpo)
```

```
##   grade      tutor
## 1    65  Anastasia
## 2    72 Bernadette
## 3    66 Bernadette
## 4    74  Anastasia
## 5    73  Anastasia
## 6    71 Bernadette
```

In this data set, we have two groups of students, those who received lessons from Anastasia and those who took their classes with Bernadette. The question we want to answer is whether there's any difference in the grades received by these two groups of student. Back in Chapter@refch:ttest I suggested you could analyse this kind of data using the `independentSamplesTTest()` function in the `lsr` package. For example, if you want to run a Student's t-test, you'd use a command like this:

```
independentSamplesTTest(
    formula = grade ~ tutor,
    data = harpo,
    var.equal = TRUE
 )
```

```
##
##    Student's independent samples t-test
##
## Outcome variable:   grade
## Grouping variable:  tutor
##
## Descriptive statistics:
##             Anastasia Bernadette
##    mean        74.533     69.056
##    std dev.     8.999      5.775
##
## Hypotheses:
##    null:        population means equal for both groups
##    alternative: different population means in each group
##
## Test results:
##    t-statistic:  2.115
##    degrees of freedom:  31
##    p-value:  0.043
##
## Other information:
##    two-sided 95% confidence interval:  [0.197, 10.759]
##    estimated effect size (Cohen's d):  0.74
```

Like most of the functions that I wrote for this book, the `independentSamplesTTest()` is very wordy. It prints out a bunch of descriptive statistics and a reminder of what the null and alternative hypotheses are, before finally getting to the test results. I wrote it that way deliberately, in order to help make things a little clearer for people who are new to statistics.

Again, we obtain a p-value less than 0.05, so we reject the null hypothesis.

What does the Bayesian version of the t-test look like? Using the `ttestBF()` function, we can obtain a Bayesian analog of Student's independent samples t-test using the following command:

```
ttestBF( formula = grade ~ tutor, data = harpo )
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 1.754927 ±0%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

Notice that format of this command is pretty standard. As usual we have a `formula` argument in which we specify the outcome variable on the left hand side and the grouping variable on the right. The `data` argument is used to specify the data frame containing the variables. However, notice that there's no analog of the `var.equal` argument. This is because the `BayesFactor` package does not include an analog of the Welch test, only the Student test.[270] In any case, when you run this command you get this as the output:

So what does all this mean? Just as we saw with the `contingencyTableBF()` function, the output is pretty dense. But, just like last time, there's not a lot of information here that you actually need to process. Firstly, let's examine the bottom line. The `BFindepSample` part just tells you that you ran an independent samples t-test, and the `JZS` part is technical information that is a little beyond the scope of this book.[271] Clearly, there's nothing to worry about in that part. In the line above, the text `Null, mu1-mu2 = 0` is just telling you that the null hypothesis is that there are no differences between means. But you already knew that. So the only part that really matters is this line here:

```
[1] Alt., r=0.707 : 1.754927 @plusorminus0%
```

Ignore the `r=0.707` part: it refers to a technical detail that we won't worry about in this chapter.[272] Instead, you should focus on the part that reads `1.754927`. This is the Bayes factor: the evidence provided by these data are about 1.8:1 in favour of the alternative.

Before moving on, it's worth highlighting the difference between the orthodox test results and the Bayesian one. According to the orthodox test, we obtained a significant result, though only barely. Nevertheless, many people would happily accept p=.043 as reasonably strong evidence for an effect. In contrast, notice that the Bayesian test doesn't even reach 2:1 odds in favour of an effect, and would be considered very weak evidence at best. In my experience that's a pretty typical outcome. Bayesian methods usually require more evidence before rejecting the null.

19.7.2 Paired samples t-test

Back in Section 13.5 I discussed the `chico` data frame in which students grades were measured on two tests, and we were interested in finding out whether grades went up from test 1 to test 2. Because every student did both tests, the tool we used to analyse the data was a paired samples t-test. To remind you of what the data look like, here's the first few cases:

```
load("./rbook-master/data/chico.rdata")
head(chico)
```

```
##         id grade_test1 grade_test2
## 1 student1        42.9        44.6
## 2 student2        51.8        54.0
## 3 student3        71.7        72.3
## 4 student4        51.6        53.4
## 5 student5        63.5        63.8
## 6 student6        58.0        59.3
```

We originally analysed the data using the `pairedSamplesTTest()` function in the `lsr` package, but this time we'll use the `ttestBF()` function from the `BayesFactor` package to do the same thing. The easiest way to do it with this data set is to use the `x` argument to specify one variable and the `y` argument to specify the other. All we need to do then is specify `paired=TRUE` to tell R that this is a paired samples test. So here's our command:

```
ttestBF(
    x = chico$grade_test1,
    y = chico$grade_test2,
    paired = TRUE
 )
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 5992.05 ±0%
##
## Against denominator:
##   Null, mu = 0
## ---
## Bayes factor type: BFoneSample, JZS
```

At this point, I hope you can read this output without any difficulty. The data provide evidence of about 6000:1 in favour of the alternative. We could probably reject the null with some confidence!

This page titled 19.7: Bayesian t-tests is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **17.7: Bayesian t-tests** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

19.8: Bayesian Regression

Okay, so now we've seen Bayesian equivalents to orthodox chi-square tests and t-tests. What's next? If I were to follow the same progression that I used when developing the orthodox tests you'd expect to see ANOVA next, but I think it's a little clearer if we start with regression.

19.8.1 quick refresher

In Chapter 15 I used the `parenthood` data to illustrate the basic ideas behind regression. To remind you of what that data set looks like, here's the first six observations:

```
load("./rbook-master/data/parenthood.Rdata")
head(parenthood)
```

```
##   dan.sleep baby.sleep dan.grump day
## 1      7.59      10.18        56   1
## 2      7.91      11.66        60   2
## 3      5.14       7.92        82   3
## 4      7.71       9.61        55   4
## 5      6.68       9.75        67   5
## 6      5.99       5.04        72   6
```

Back in Chapter 15 I proposed a theory in which my grumpiness (`dan.grump`) on any given day is related to the amount of sleep I got the night before (`dan.sleep`), and possibly to the amount of sleep our baby got (`baby.sleep`), though probably not to the `day` on which we took the measurement. We tested this using a regression model. In order to estimate the regression model we used the `lm()` function, like so:

```
model <- lm(
  formula = dan.grump ~ dan.sleep + day + baby.sleep,
  data = parenthood
)
```

The hypothesis tests for each of the terms in the regression model were extracted using the `summary()` function as shown below:

```
summary(model)
```

```
##
## Call:
## lm(formula = dan.grump ~ dan.sleep + day + baby.sleep, data = parenthood)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.906  -2.284  -0.295   2.652  11.880
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 126.278707   3.242492  38.945   <2e-16 ***
## dan.sleep    -8.969319   0.560007 -16.016   <2e-16 ***
## day          -0.004403   0.015262  -0.288    0.774
## baby.sleep    0.015747   0.272955   0.058    0.954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.375 on 96 degrees of freedom
## Multiple R-squared:  0.8163, Adjusted R-squared:  0.8105
## F-statistic: 142.2 on 3 and 96 DF,  p-value: < 2.2e-16
```

When interpreting the results, each row in this table corresponds to one of the possible predictors. The `(Intercept)` term isn't usually interesting, though it is highly significant. The important thing for our purposes is the fact that `dan.sleep` is significant at p<.001 and neither of the other variables are.

19.8.2 Bayesian version

Okay, so how do we do the same thing using the `BayesFactor` package? The easiest way is to use the `regressionBF()` function instead of `lm()`. As before, we use `formula` to indicate what the full regression model looks like, and the `data` argument to specify the data frame. So the command is:

```
regressionBF(
  formula = dan.grump ~ dan.sleep + day + baby.sleep,
  data = parenthood
)
```

```
## Bayes factor analysis
## --------------
## [1] dan.sleep                    : 1.622545e+34 ±0.01%
## [2] day                          : 0.2724027    ±0%
## [3] baby.sleep                   : 10018411     ±0%
## [4] dan.sleep + day              : 1.016576e+33 ±0%
## [5] dan.sleep + baby.sleep       : 9.77022e+32  ±0%
## [6] day + baby.sleep             : 2340755      ±0%
## [7] dan.sleep + day + baby.sleep : 7.835625e+31 ±0%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

So that's pretty straightforward: it's exactly what we've been doing throughout the book. The output, however, is a little different from what you get from `lm()`. The format of this is pretty familiar. At the bottom we have some techical rubbish, and at the top we have some information about the Bayes factors. What's new is the fact that we seem to have *lots* of Bayes factors here. What's all this about?

The trick to understanding this output is to recognise that if we're interested in working out which of the 3 predictor variables are related to `dan.grump`, there are actually 8 possible regression models that could be considered. One possibility is the *intercept only model*, in which none of the three variables have an effect. At the other end of the spectrum is the *full model* in which all three variables matter. So what `regressionBF()` does is treat the *intercept only* model as the null hypothesis, and print out the Bayes factors for all other models when compared against that null. For example, if we look at line 4 in the table, we see that the evidence is about 1033 to 1 in favour of the claim that a model that includes both `dan.sleep` and `day` is better than the intercept only model. Or if we look at line 1, we can see that the odds are about 1.6×1034 that a model containing the `dan.sleep` variable (but no others) is better than the intercept only model.

19.8.3 Finding the best model

In practice, this isn't super helpful. In most situations the intercept only model is one that you don't really care about at all. What I find helpful is to start out by working out which model is the *best* one, and then seeing how well all the alternatives compare to it. Here's how you do that. In this case, it's easy enough to see that the best model is actually the one that contains `dan.sleep` only (line 1), because it has the largest Bayes factor. However, if you've got a lot of possible models in the output, it's handy to know that you can use the `head()` function to pick out the best few models. First, we have to go back and save the Bayes factor information to a variable:

```
models <- regressionBF(
  formula = dan.grump ~ dan.sleep + day + baby.sleep,
  data = parenthood
)
```

Let's say I want to see the best three models. To do this, I use the `head()` function specifying `n=3`, and here's what I get as the result:

```
head( models, n = 3)
```

```
## Bayes factor analysis
## --------------
## [1] dan.sleep              : 1.622545e+34 ±0.01%
## [2] dan.sleep + day        : 1.016576e+33 ±0%
## [3] dan.sleep + baby.sleep : 9.77022e+32  ±0%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

This is telling us that the model in line 1 (i.e., `dan.grump ~ dan.sleep`) is the best one. That's *almost* what I'm looking for, but it's still comparing all the models against the intercept only model. That seems silly. What I'd like to know is how big the difference is between the best model and the other good models. For that, there's this trick:

```
head( models/max(models), n = 3)
```

```
## Bayes factor analysis
## --------------
## [1] dan.sleep              : 1         ±0%
## [2] dan.sleep + day        : 0.0626532 ±0.01%
## [3] dan.sleep + baby.sleep : 0.0602154 ±0.01%
##
## Against denominator:
##   dan.grump ~ dan.sleep
## ---
## Bayes factor type: BFlinearModel, JZS
```

Notice the bit at the bottom showing that the "denominator" has changed. What that means is that the Bayes factors are now comparing each of those 3 models listed against the `dan.grump ~ dan.sleep` model. Obviously, the Bayes factor in the first line is exactly 1, since that's just comparing the best model to itself. More to the point, the other two Bayes factors are both less than 1, indicating that they're all worse than that model. The Bayes factors of 0.06 to 1 imply that the odds for the best model over the second best model are about 16:1. You can work this out by simple arithmetic (i.e., 0.06/1≈16), but the other way to do it is to directly compare the models. To see what I mean, here's the original output:

```
models
```

```
## Bayes factor analysis
## --------------
## [1] dan.sleep                    : 1.622545e+34 ±0.01%
## [2] day                          : 0.2724027    ±0%
## [3] baby.sleep                   : 10018411     ±0%
## [4] dan.sleep + day              : 1.016576e+33 ±0%
## [5] dan.sleep + baby.sleep       : 9.77022e+32  ±0%
## [6] day + baby.sleep             : 2340755      ±0%
## [7] dan.sleep + day + baby.sleep : 7.835625e+31 ±0%
##
## Against denominator:
##   Intercept only
## ---
## Bayes factor type: BFlinearModel, JZS
```

The best model corresponds to row 1 in this table, and the second best model corresponds to row 4. All you have to do to compare these two models is this:

```
models[1] / models[4]
```

```
## Bayes factor analysis
## --------------
## [1] dan.sleep : 15.96088 ±0.01%
##
## Against denominator:
##   dan.grump ~ dan.sleep + day
## ---
## Bayes factor type: BFlinearModel, JZS
```

And there you have it. You've found the regression model with the highest Bayes factor (i.e., `dan.grump ~ dan.sleep`), and you know that the evidence for that model over the next best alternative (i.e., `dan.grump ~ dan.sleep + day`) is about 16:1.

19.8.4 Extracting Bayes factors for all included terms

Okay, let's say you've settled on a specific regression model. What Bayes factors should you report? In this example, I'm going to pretend that you decided that `dan.grump ~ dan.sleep + baby.sleep` is the model you think is best. Sometimes it's sensible to do this, even when it's not the one with the highest Bayes factor. Usually this happens because you have a substantive theoretical reason to prefer one model over the other. However, in this case I'm doing it because I want to use a model with more than one predictor as my example!

Having figured out which model you prefer, it can be really useful to call the `regressionBF()` function and specifying `whichModels="top"`. You use your "preferred" model as the `formula` argument, and then the output will show you the Bayes factors that result when you try to drop predictors from this model:

```
regressionBF(
 formula = dan.grump ~ dan.sleep + baby.sleep,
 data = parenthood,
 whichModels = "top"
)
```

```
## Bayes factor top-down analysis
## --------------
## When effect is omitted from dan.sleep + baby.sleep , BF is...
## [1] Omit baby.sleep : 16.60705     ±0.01%
## [2] Omit dan.sleep  : 1.025403e-26 ±0.01%
##
## Against denominator:
##   dan.grump ~ dan.sleep + baby.sleep
## ---
## Bayes factor type: BFlinearModel, JZS
```

Okay, so now you can see the results a bit more clearly. The Bayes factor when you try to drop the `dan.sleep` predictor is about 10−26, which is very strong evidence that you *shouldn't* drop it. On the other hand, the Bayes factor actually goes up to 17 if you drop `baby.sleep` , so you'd usually say that's pretty strong evidence for dropping that one.

This page titled 19.8: Bayesian Regression is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **17.8: Bayesian Regression** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

19.9: Bayesian ANOVA

As you can tell, the `BayesFactor` package is pretty flexible, and it can do Bayesian versions of pretty much everything in this book. In fact, it can do a few other neat things that I haven't covered in the book at all. However, I have to stop somewhere, and so there's only one other topic I want to cover: Bayesian ANOVA.

quick refresher

As with the other examples, I think it's useful to start with a reminder of how I discussed ANOVA earlier in the book. First, let's remind ourselves of what the data were. The example I used originally is the `clin.trial` data frame, which looks like this

```
load("./rbook-master/data/clinicaltrial.Rdata")
head(clin.trial)
```

```
##       drug    therapy mood.gain
## 1  placebo no.therapy       0.5
## 2  placebo no.therapy       0.3
## 3  placebo no.therapy       0.1
## 4 anxifree no.therapy       0.6
## 5 anxifree no.therapy       0.4
## 6 anxifree no.therapy       0.2
```

To run our orthodox analysis in earlier chapters we used the `aov()` function to do all the heavy lifting. In Chapter 16 I recommended using the `Anova()` function from the `car` package to produce the ANOVA table, because it uses Type II tests by default. If you've forgotten what "Type II tests" are, it might be a good idea to re-read Section 16.10, because it will become relevant again in a moment. In any case, here's what our analysis looked like:

```
library(car)
```

```
## Loading required package: carData
```

```
model <- aov( mood.gain ~ drug * therapy, data = clin.trial )
Anova(model)
```

```
## Anova Table (Type II tests)
##
## Response: mood.gain
##              Sum Sq Df F value    Pr(>F)
## drug         3.4533  2 31.7143 1.621e-05 ***
## therapy      0.4672  1  8.5816   0.01262 *
## drug:therapy 0.2711  2  2.4898   0.12460
## Residuals    0.6533 12
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

That's pretty clearly showing us evidence for a main effect of `drug` at p<.001, an effect of `therapy` at p<.05 and no interaction.

Bayesian version

How do we do the same thing using Bayesian methods? The `BayesFactor` package contains a function called `anovaBF()` that does this for you. It uses a pretty standard `formula` and `data` structure, so the command should look really familiar. Just like we did with regression, it will be useful to save the output to a variable:

```
models <- anovaBF(
 formula = mood.gain ~ drug * therapy,
 data = clin.trial
)
```

The output is quite different to the traditional ANOVA, but it's not too bad once you understand what you're looking for. Let's take a look:

```
models
```

This looks very similar to the output we obtained from the `regressionBF()` function, and with good reason. Remember what I said back in Section 16.6: under the hood, ANOVA is no different to regression, and both are just different examples of a linear model. Becasue of this, the `anovaBF()` reports the output in much the same way. For instance, if we want to identify the best model we could use the same commands that we used in the last section. One variant that I find quite useful is this:

```
models/max(models)
```

```
## Bayes factor analysis
## --------------
## [1] drug                          : 0.3521042   ±0.94%
## [2] therapy                       : 0.001047568 ±0.94%
## [3] drug + therapy                : 1           ±0%
## [4] drug + therapy + drug:therapy : 0.978514    ±1.29%
##
## Against denominator:
##   mood.gain ~ drug + therapy
## ---
## Bayes factor type: BFlinearModel, JZS
```

By "dividing" the `models` output by the best model (i.e., `max(models)`), what R is doing is using the best model (which in this case is `drugs + therapy`) as the denominator, which gives you a pretty good sense of how close the competitors are. For instance, the model that contains the interaction term is almost as good as the model without the interaction, since the Bayes factor is 0.98. In other words, the data do not clearly indicate whether there is or is not an interaction.

Constructing Bayesian Type II tests

Okay, that's all well and good, you might be thinking, but what do I report as the alternative to the p-value? In the classical ANOVA table, you get a single p-value for every predictor in the model, so you can talk about the significance of each effect. What's the Bayesian analog of this?

It's a good question, but the answer is tricky. Remember what I said in Section 16.10 about ANOVA being complicated. Even in the classical version of ANOVA there are several different "things" that ANOVA might correspond to. Specifically, I discussed how you get different p-values depending on whether you use Type I tests, Type II tests or Type III tests. To work out which Bayes factor is analogous to "the" p-value in a classical ANOVA, you need to work out which version of ANOVA you want an analog for. For the purposes of this section, I'll assume you want Type II tests, because those are the ones I think are most sensible in general.

As I discussed back in Section 16.10, Type II tests for a two-way ANOVA are reasonably straightforward, but if you have forgotten that section it wouldn't be a bad idea to read it again before continuing.

Assuming you've had a refresher on Type II tests, let's have a look at how to pull them from the Bayes factor table. Suppose we want to test the main effect of `drug`. The null hypothesis for this test corresponds to a model that includes an effect of `therapy`, but no effect of `drug`. The alternative hypothesis is the model that includes both. In other words, what we want is the Bayes factor corresponding to this comparison:

```
knitr::kable(tibble::tribble(
                  ~V1,                              ~V2,
         "Null model:",        "`mood.gain ~ therapy`",
  "Alternative model:", "`mood.gain ~ therapy + drug`"
  ), col.names = c("", ""))
```

Null model:	mood.gain ~ therapy
Alternative model:	mood.gain ~ therapy + drug

As it happens, we can read the answer to this straight off the table because it corresponds to a comparison between the model in line 2 of the table and the model in line 3: the Bayes factor in this case represents evidence *for* the null of 0.001 to 1. Or, more helpfully, the odds are about 1000 to 1 against the null.

The main effect of `therapy` can be calculated in much the same way. In this case, the null model is the one that contains only an effect of drug, and the alternative is the model that contains both. So the relevant comparison is between lines 2 and 1 in the table. The odds in favour of the null here are only 0.35 to 1. Again, I find it useful to frame things the other way around, so I'd refer to this as evidence of about 3 to 1 in favour of an effect of `therapy`.

Finally, in order to test an interaction effect, the null model here is one that contains both main effects but no interaction. The alternative model adds the interaction. That is:

```
knitr::kable(tibble::tribble(
                  ~V1,                              ~V2,
         "Null model:",        "`mood.gain ~ drug + therapy`",
  "Alternative model:", "`mood.gain ~ drug + therapy + drug:therapy`"
  ), col.names = c("", ""))
```

Null model:	mood.gain ~ drug + therapy
Alternative model:	mood.gain ~ drug + therapy + drug:therapy

If we look those two models up in the table, we see that this comparison is between the models on lines 3 and 4 of the table. The odds of 0.98 to 1 imply that these two models are fairly evenly matched.

You might be thinking that this is all pretty laborious, and I'll concede that's true. At some stage I might consider adding a function to the `lsr` package that would automate this process and construct something like a "Bayesian Type II ANOVA table" from the output of the `anovaBF()` function. However, I haven't had time to do this yet, nor have I made up my mind about whether it's really a good idea to do this. In the meantime, I thought I should show you the trick for how I do this in practice. The command that I use when I want to grab the right Bayes factors for a Type II ANOVA is this one:

```
max(models)/models
```

```
##                denominator
## numerator            drug  therapy drug + therapy
##   drug + therapy 2.840068 954.5918             1
##                denominator
## numerator        drug + therapy + drug:therapy
##   drug + therapy                      1.021958
```

The output isn't quite so pretty as the last one, but the nice thing is that you can read off everything you need. The best model is `drug + therapy`, so all the other models are being compared to that. What's the Bayes factor *for* the main effect of `drug`? The relevant null hypothesis is the one that contains only `therapy`, and the Bayes factor in question is 954:1. The main effect of `therapy` is weaker, and the evidence here is only 2.8:1. Finally, the evidence *against* an interaction is very weak, at 1.01:1.

Reading the results off this table is sort of counterintuitive, because you have to read off the answers from the "wrong" part of the table. For instance, the evidence for an effect of `drug` can be read from the column labelled `therapy`, which is pretty damned weird. To be fair to the authors of the package, I don't think they ever intended for the `anovaBF()` function to be used this way. My understanding[273] is that their view is simply that you should find the best model and report that model: there's no inherent reason why a Bayesian ANOVA should try to follow the exact same design as an orthodox ANOVA.[274]

In any case, if you know what you're looking for, you can look at this table and then report the results of the Bayesian analysis in a way that is pretty closely analogous to how you'd report a regular Type II ANOVA. As I mentioned earlier, there's still no convention on how to do that, but I usually go for something like this:

A Bayesian Type II ANOVA found evidence for main effects of drug (Bayes factor: 954:1) and therapy (Bayes factor: 3:1), but no clear evidence for or against an interaction (Bayes factor: 1:1).

This page titled 19.9: Bayesian ANOVA is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **17.9: Bayesian ANOVA** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

19.10: Summary

The first half of this chapter was focused primarily on the theoretical underpinnings of Bayesian statistics. I introduced the mathematics for how Bayesian inference works (Section 17.1), and gave a very basic overview of how Bayesian hypothesis testing is typically done (Section 17.2). Finally, I devoted some space to talking about why I think Bayesian methods are worth using (Section 17.3.

The second half of the chapter was a lot more practical, and focused on tools provided by the `BayesFactor` package. Specifically, I talked about using the `contingencyTableBF()` function to do Bayesian analogs of chi-square tests (Section 17.6, the `ttestBF()` function to do Bayesian t-tests, (Section 17.7), the `regressionBF()` function to do Bayesian regressions, and finally the `anovaBF()` function for Bayesian ANOVA.

If you're interested in learning more about the Bayesian approach, there are many good books you could look into. John Kruschke's book *Doing Bayesian Data Analysis* is a pretty good place to start (Kruschke 2011), and is a nice mix of theory and practice. His approach is a little different to the "Bayes factor" approach that I've discussed here, so you won't be covering the same ground. If you're a cognitive psychologist, you might want to check out Michael Lee and E.J. Wagenmakers' book *Bayesian Cognitive Modeling* (Lee and Wagenmakers 2014). I picked these two because I think they're especially useful for people in my discipline, but there's a lot of good books out there, so look around!

References

Jeffreys, Harold. 1961. *The Theory of Probability*. 3rd ed. Oxford.

Kass, Robert E., and Adrian E. Raftery. 1995. "Bayes Factors." *Journal of the American Statistical Association* 90: 773–95.

Fisher, R. 1925. *Statistical Methods for Research Workers*. Edinburgh, UK: Oliver; Boyd.

Johnson, Valen E. 2013. "Revised Standards for Statistical Evidence." *Proceedings of the National Academy of Sciences*, no. 48: 19313–7.

Morey, Richard D., and Jeffrey N. Rouder. 2015. *BayesFactor: Computation of Bayes Factors for Common Designs*. http://CRAN.R-project.org/package=BayesFactor.

Gunel, Erdogan, and James Dickey. 1974. "Bayes Factors for Independence in Contingency Tables." *Biometrika*, 545–57.

Kruschke, J. K. 2011. *Doing Bayesian Data Analysis: A Tutorial with R and BUGS*. Burlington, MA: Academic Press.

Lee, Michael D, and Eric-Jan Wagenmakers. 2014. *Bayesian Cognitive Modeling: A Practical Course*. Cambridge University Press.

253. http://en.wikiquote.org/wiki/David_Hume
254. http://en.Wikipedia.org/wiki/Climate_of_Adelaide
255. It's a leap of faith, I know, but let's run with it okay?
256. Um. I hate to bring this up, but some statisticians would object to me using the word "likelihood" here. The problem is that the word "likelihood" has a very specific meaning in frequentist statistics, and it's not quite the same as what it means in Bayesian statistics. As far as I can tell, Bayesians didn't originally have any agreed upon name for the likelihood, and so it became common practice for people to use the frequentist terminology. This wouldn't have been a problem, except for the fact that the way that Bayesians use the word turns out to be quite different to the way frequentists do. This isn't the place for yet another lengthy history lesson, but to put it crudely: when a Bayesian says "*a* likelihood function" they're usually referring one of the *rows* of the table. When a frequentist says the same thing, they're referring to the same table, but to them "*a* likelihood function" almost always refers to one of the *columns*. This distinction matters in some contexts, but it's not important for our purposes.
257. If we were being a bit more sophisticated, we could extend the example to accommodate the possibility that I'm lying about the umbrella. But let's keep things simple, shall we?
258. You might notice that this equation is actually a restatement of the same basic rule I listed at the start of the last section. If you multiply both sides of the equation by P(d), then you get P(d)P(h|d)=P(d,h), which is the rule for how joint probabilities are

calculated. So I'm not actually introducing any "new" rules here, I'm just using the same rule in a different way.

259. Obviously, this is a highly simplified story. All the complexity of real life Bayesian hypothesis testing comes down to how you calculate the likelihood P(d|h) when the hypothesis h is a complex and vague thing. I'm not going to talk about those complexities in this book, but I do want to highlight that although this simple story is true as far as it goes, real life is messier than I'm able to cover in an introductory stats textbook.
260. http://www.imdb.com/title/tt0093779/quotes. I should note in passing that I'm not the first person to use this quote to complain about frequentist methods. Rich Morey and colleagues had the idea first. I'm shamelessly stealing it because it's such an awesome pull quote to use in this context and I refuse to miss any opportunity to quote *The Princess Bride*.
261. http://about.abc.net.au/reports-publications/appreciation-survey-summary-report-2013/
262. http://knowyourmeme.com/memes/the-cake-is-a-lie
263. In the interests of being completely honest, I should acknowledge that not all orthodox statistical tests that rely on this silly assumption. There are a number of *sequential analysis* tools that are sometimes used in clinical trials and the like. These methods are built on the assumption that data are analysed as they arrive, and these tests aren't horribly broken in the way I'm complaining about here. However, sequential analysis methods are constructed in a very different fashion to the "standard" version of null hypothesis testing. They don't make it into any introductory textbooks, and they're not very widely used in the psychological literature. The concern I'm raising here is valid for every single orthodox test I've presented so far, and for almost every test I've seen reported in the papers I read.
264. A related problem: http://xkcd.com/1478/
265. Some readers might wonder why I picked 3:1 rather than 5:1, given that Johnson (2013) suggests that p=.05 lies somewhere in that range. I did so in order to be charitable to the p-value. If I'd chosen a 5:1 Bayes factor instead, the results would look even better for the Bayesian approach.
266. http://www.quotationspage.com/quotes/Ambrosius_Macrobius/
267. Okay, I just *know* that some knowledgeable frequentists will read this and start complaining about this section. Look, I'm not dumb. I absolutely know that if you adopt a sequential analysis perspective you can avoid these errors within the orthodox framework. I also know that you can explictly design studies with interim analyses in mind. So yes, in one sense I'm attacking a "straw man" version of orthodox methods. However, the straw man that I'm attacking is the one that *is used by almost every single practitioner*. If it ever reaches the point where sequential methods become the norm among experimental psychologists and I'm no longer forced to read 20 extremely dubious ANOVAs a day, I promise I'll rewrite this section and dial down the vitriol. But until that day arrives, I stand by my claim that *default* Bayes factor methods are much more robust in the face of data analysis practices as they exist in the real world. *Default* orthodox methods suck, and we all know it.
268. If you're desperate to know, you can find all the gory details in Gunel and Dickey (1974). However, that's a pretty technical paper. The help documentation to the `contingencyTableBF()` gives this explanation: "the argument `priorConcentration` indexes the expected deviation from the null hypothesis under the alternative, and corresponds to Gunel and Dickey's (1974) a parameter." As I write this I'm about halfway through the Gunel and Dickey paper, and I agree that setting a=1 is a pretty sensible default choice, since it corresponds to an assumption that you have very little *a priori* knowledge about the contingency table.
269. In some of the later examples, you'll see that this number is not always 0%. This is because the `BayesFactor` package often has to run some simulations to compute approximate Bayes factors. So the answers you get won't always be identical when you run the command a second time. That's why the output of these functions tells you what the margin for error is.
270. Apparently this omission is deliberate. I have this vague recollection that I spoke to Jeff Rouder about this once, and his opinion was that when homogeneity of variance is violated the results of a t-test are uninterpretable. I can see the argument for this, but I've never really held a strong opinion myself. (Jeff, if you never said that, I'm sorry)
271. Just in case you're interested: the "JZS" part of the output relates to how the Bayesian test expresses the prior uncertainty about the variance $\sigma 2$, and it's short for the names of three people: "Jeffreys Zellner Siow". See Rouder et al. (2009) for details.
272. Again, in case you care … the null hypothesis here specifies an effect size of 0, since the two means are identical. The alternative hypothesis states that there *is* an effect, but it doesn't specify exactly how big the effect will be. The r value here relates to how big the effect is expected to be according to the alternative. You can type `?ttestBF` to get more details.
273. Again, guys, sorry if I've misread you.
274. I don't even disagree with them: it's *not* at all obvious why a Bayesian ANOVA should reproduce (say) the same set of model comparisons that the Type II testing strategy uses. It's precisely because of the fact that I haven't really come to any strong

conclusions that I haven't added anything to the `lsr` package to make Bayesian Type II tests easier to produce.

This page titled 19.10: Summary is shared under a CC BY-SA 4.0 license and was authored, remixed, and/or curated by Danielle Navarro via source content that was edited to the style and standards of the LibreTexts platform.

- **17.10: Summary** by Danielle Navarro is licensed CC BY-SA 4.0. Original source: https://bookdown.org/ekothe/navarro26/.

CHAPTER OVERVIEW

20: Case Studies and Data

20.1: Angry Moods
20.2: Flatulence
20.3: Physicians Reactions
20.4: Teacher Ratings
20.5: Diet and Health
20.6: Smiles and Leniency
20.7: Animal Research
20.8: ADHD Treatment
20.9: Weapons and Aggression
20.10: SAT and College GPA
20.11: Stereograms
20.12: Driving
20.13: Stroop Interference
20.14: TV Violence
20.15: Obesity and Bias
20.16: Shaking and Stirring Martinis
20.17: Adolescent Lifestyle Choices
20.18: Chocolate and Body Weight
20.19: Bedroom TV and Hispanic Children
20.20: Weight and Sleep Apnea
20.21: Misusing SEM
20.22: School Gardens and Vegetable Consumption
20.23: TV and Hypertension
20.24: Dietary Supplements
20.25: Young People and Binge Drinking
20.26: Sugar Consumption in the US Diet
20.27: Nutrition Information Sources and Older Adults
20.28: Mind Set - Exercise and the Placebo Effect
20.29: Predicting Present and Future Affect
20.30: Exercise and Memory
20.31: Parental Recognition of Child Obesity
20.32: Educational Attainment and Racial, Ethnic, and Gender Disparity

This page titled 20: Case Studies and Data is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

20.1: Angry Moods

Learning Objectives

- Ways to improve an angry mood: A look at gender and sports participation

Research conducted by

Emily Zitek and Mindy Ater, Rice University

Case study prepared by

Emily Zitek

Overview

People have different ways of improving their mood when angry. We have all seen people punch a wall when mad, and indeed, previous research has indicated that some people aggress to improve their mood (Bushman, Baumeister & Phillips, 2001). What do the top athletes do when angry? Striegel (1994) found that anger often hurts an athlete's performance and that capability to control anger is what makes good athletes even better. This study adds to the past research and examines the difference in ways to improve an angry mood by gender and sports participation.

The participants were 78 Rice University undergraduates, ages 17 to 23. Of these 78 participants, 48 were females and 30 were males and 25 were athletes and 53 were non-athletes. People who did not play a varsity or club sport were considered non-athletes. The 13 contact sport athletes played soccer, football, rugby, or basketball, and the 12 non-contact sport athletes participated in Ultimate Frisbee, baseball, tennis, swimming, volleyball, crew, or dance.

The participants were asked to respond to a questionnaire that asked about what they do to improve their mood when angry or furious. Then they filled out a demographics questionnaire.

Note

This study used the most recent version of the State-Trait Anger Expression Inventory (STAXI-2) (Spielberger, Sydeman, Owen & Marsh, 1999) which was modified to create an Angry Mood Improvement Inventory similar to that created by Bushman et al. (2001).

Questions to Answer

Do athletes and non-athletes deal with anger in the same way? Are there any gender differences? Specifically, are men more likely to believe that aggressive behavior can improve an angry mood?

Design Issues

This study has an extremely unbalanced design. There were a lot more non-athletes than athletes in the sample. In the future, more athletes should be used. This study originally wanted to look at contact and non-contact athletes separately, but there were not enough participants to do this. Future studies could look at this.

Descriptions of Variables

Table 20.1.1: Description of Variables. Note that the description of the items comes from Spielberger et

al. (1999)

Variable	Description
Sports	1 = athletes, 2 = non-athletes
Gender	1 = males, 2 = females
Anger-Out (AO)	high scores demonstrate that people deal with anger by expressing it in a verbally or physically aggressive fashion
Anger-In (AI)	high scores demonstrate that people experience anger but do not express it (suppress their anger)
Control-Out (CO)	high scores demonstrate that people control the outward expression of angry feelings
Control-In (CI)	high scores demonstrate that people control angry feelings by calming down or cooling off
Expression (AE)	index of general anger expression: (Anger-Out) + (Anger-In) - (Control-Out) - (Control-In) + 48

Data files

angry_moods.xls

References

- Bushman, B.J., Baumeister, R.F. & Phillips, C.M. (2001). Do people aggress to improve their mood? Catharsis beliefs, affect regulation opportunity, and aggressive responding. Journal of Personality and Social Psychology, 81(1), 17-32.
- Spielberger, C. D., Sydeman, S. J., Owen, A. E., Marsh, B. J. (1999). Measuring anxiety and anger with the State-Trait Anxiety Inventory (STAI) and the State-Trait Anger Expression Inventory (STAXI). In M. E. Maruish (Ed.), The use of psychological testing for treatment planning and outcomes assessment (2nd ed., pp. 993-1021). Mahwah: Lawrence Erlbaum Associates.
- Striegel, D. (1994). Anger in tennis: Part 2. Effects of anger on performance, coping with anger, and using anger to one's benefit. Journal of Performance Psychology, 2, 56-92.

Links

Publisher's description

This page titled 20.1: Angry Moods is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.1: Angry Moods** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.2: Flatulence

Learning Objectives

- Flatulence: Are you embarrassed by your flatus?

Research conducted by

Shannon E. Collins, UH-D undergraduate,

Faculty Advisor: Heidi Ziemer

Case study prepared by

Shannon E. Collins

Overview

The purpose of this study was to find out whether or not people are embarrassed by their flatulence. The participants were 35 University of Houston – Downtown students. Flatulence is a normal part of being human, but it can cause an alarming rate of embarrassment in certain situations. How many times have you been subjected to the unpleasant odor emitted from someone around you? Medical research indicates that it is normal to have anywhere from 7 to 20 episodes of gas in a day.

Would you believe that women produce more of the bad smelling stuff than men do, and that women are more likely to complain to their doctors about the smell of their flatulence? The smell comes from sulfur gasses, the most offensive of which is hydrogen sulfide; it smells like rotten eggs. Still, everybody does it, we just don't know how embarrassed they are by it.

Questions to Answer

Are people without male siblings more embarrassed by their flatulence than people with one or more male siblings? Do people that come from households where flatulence was acceptable report less embarrassment than people that come from households where it was not acceptable? Are women or men more embarrassed by their flatus?

Design Issues

Embarrassment scores were reported on 14 different measures and tallied as a total embarrassment score, then divided into seven categories, producing one number per category. Only the scores on the seven categories are reported here. The data in this research is self report data, and because the topic is sensitive some people may have been less than honest about their reported flatulence.

Descriptions of Variables

Table 20.2.1: Description of variables

Variable	Description
Gender	1 = male, 2 = female
famaccp	Household acceptance of flatus 1 to 7, 1=very acceptable and 7=very unacceptable
brother	Number of brothers the participant has
howlong	How long before farting in front of partner? 1 = 1 year, .5= 6 months, .25=3 months, and smaller decimals represent portions of a year. Numbers larger than 1 indicate longer than 1 year.
perday	Number per day
Embarrassing Situations	The following variables were rated on this scale: 1=extremely embarrassed and 7= not really embarrassed
mtgwork	Meeting at work
talkprof	Talking to a professor
romint	Romantic interest

Data files

flatulence.xls

References

- Chapman, S. (2001 December 22). Hot Air? BMJ: British Medical Journal. 323(7327). Retrieved from, Health source database.
- Gases of the Gut. Harvard Health Letter. August 2002. 27(10).
- Hughes, L. (October 1999). 10 Tips on How to Fend Off Embarrassing Flatulence. Environmental Nutrition. 22(10). Retrieved from, Health source database.
- Lecture Theatre Etiquette. (June 2000). Student BMJ. Vol. 8. Retrieved from, Health source database.
- Manley, W. (October 1999). Noisome Rumblings. American Libraries Online. Retrieved from, www.search.epnet.com/direct.a...4&db=hch&tg=AN
- Robb-Nicholson, C. (March 2003). By The Way Doctor. Harvard Women's Health Watch. 10(7). Retrieved from, Academic Edition of Health Source database.

This page titled 20.2: Flatulence is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.2: Flatulence** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.3: Physicians Reactions

Learning Objectives

- Physicians' Reactions to Patient Size

Research conducted by

Mikki Hebl and Jingping Xu

Case study prepared by

Emily Zitek

Overview

Obese people face discrimination on a daily basis in employment, education, and relationship contexts. Past research has shown that even doctors, who are trained to treat all their patients warmly and have access to literature suggesting uncontrollable and hereditary aspects of obesity, believe obese individuals are undisciplined and suffer from controllability issues. This case study examines how doctors treat overweight as compared to normal weight patients.

Various doctors at one of three major hospitals in the Texas Medical Center of Houston participated in the study. These doctors were sent a packet containing a medical chart similar to the one they view upon seeing a patient. This chart portrayed a patient who was displaying symptoms of a migraine headache but was otherwise healthy. This chart also contained a measure of the patient's weight. Doctors were randomly assigned to receive the chart of a patient who was overweight or the chart of a patient who was of normal weight. After reviewing the chart, the doctors then had to indicate how much time they believed they would spend with the patient.

Questions to Answer

Do doctors discriminate against overweight patients? Specifically, do the doctors who review charts of overweight patients say they would spend the same amount of time with their patients as the doctors who review charts of normal weight patients?

Design Issues

The method and data described here are only a small part of a larger study. See the reference below for a full description of the study.

Descriptions of Variables

Table **20.3.1**: Description of variables

Variable	Description
Patient weight	1 = average weight, 2 = overweight
Time	represents how long the doctors said they would spend with the patient

Data files

Weight.xls

References

- Hebl, M., & Xu, J., "Weighing the care: Physicians' reactions to the size of a patient," International Journal of Obesity, 25 (2001): 1246-1252

This page titled 20.3: Physicians Reactions is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.3: Physicians Reactions** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.4: Teacher Ratings

Learning Objectives

- Teacher Ratings

Research conducted by

Annette Towler and Robert Dipboye

Case study prepared by

Emily Zitek

Overview

How powerful are rumors? Frequently, students ask friends and/or look at instructor evaluations to decide if a class is worth taking. Kelley (1950) found that instructor reputation has a profound impact on actual teaching ratings, and Towler and Dipboye (1998) replicated and extended this study.

Subjects were randomly assigned to one of two conditions. Before viewing the lecture, students were given a summary of the instructors' prior teaching evaluations. There were two conditions: Charismatic instructor and Punitive instructor.

Then all subjects watched the **same** twenty-minute lecture given by the exact same lecturer. Following the lecture, subjects answered three questions about the leadership qualities of the lecturer. A summary rating score was computed and used as the variable "rating" here.

Questions to Answer

Does an instructor's prior reputation affect student ratings?

Design Issues

The data presented here are part of a larger study. See the references below to learn more.

Descriptions of Variables

Table 20.4.1: Description of Variables

Variable	Description
Condition	this represents the content of the description that the students were given about the professor (1 = charismatic, 2 = punitive)
Rating	how favorably the subjects rated the professor after hearing the lecture (higher ratings are more favorable)

Data files

Ratings.xls

References

- Kelley, H. H.(1950). The warm-cold variable in first impression of persons. Journal of Personality, 18, 431-439.
- Towler, A., & Dipboye, R. L. (1998). The effect of instructor reputation and need for cognition on student behavior (poster presented at American Psychological Society conference, May 1998).

This page titled 20.4: Teacher Ratings is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.4: Teacher Ratings** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.5: Diet and Health

Learning Objectives

- Mediterranean Diet and Health



Research conducted by

De Longerill et al

Case study prepared by

David Lane and Emily Zite

Overview

Most doctors would probably agree that a Mediterranean diet, rich in vegetables, fruits, and grains, is healthier than a high-saturated fat diet. Indeed, previous research has found that the diet can lower risk of heart disease. However, there is still considerable uncertainty about whether the Mediterranean diet is superior to a low-fat diet recommended by the American Heart Association. This study is the first to compare these two diets.

The subjects, **605** survivors of a heart attack, were randomly assigned follow either

1. a diet close to the "prudent diet step 1" of the American Heart Association (control group) or
2. a Mediterranean-type diet consisting of more bread and cereals, more fresh fruit and vegetables, more grains, more fish, fewer delicatessen foods, less meat.

An experimental canola-oil-based margarine was used instead of butter or cream. The oils recommended for salad and food preparation were canola and olive oils exclusively. Moderate red wine consumption was allowed.

Over a four-year period, patients in the experimental condition were initially seen by the dietician, two months later, and then once a year. Compliance with the dietary intervention was checked by a dietary survey and analysis of plasma fatty acids. Patients in the control group were expected to follow the dietary advice given by their physician.

The researchers collected information on number of deaths from cardiovascular causes e.g., heart attack, strokes, as well as number of nonfatal heart-related episodes. The occurrence of malignant and nonmalignant tumors was also carefully monitored.

Questions to Answer

Is the Mediterranean diet superior to a low-fat diet recommended by the American Heart Association?

Design Issues

The strength of the design is that subjects were randomly assigned to conditions. A possible weakness is that compliance rates depended on reports rather than observation since observation is impractical in this type of research.

Descriptions of Variables

Table 20.5.1: Description of Variables

Variable	Description
Type of diet	AHA or Mediterranean
Various outcome measures of health and disease	does the patient have cancer, etc.?

Data Files

Diet.xls

Links

More on the Mediterranean Diet

References

- De Longerill, M., Salen, P., Martin, J., Monjaud, I., Boucher, P., Mamelle, N. (1998). Mediterranean Dietary pattern in a Randomized Trial. Archives of Internal Medicine, 158, 1181-1187.

This page titled 20.5: Diet and Health is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.5: Diet and Health** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.6: Smiles and Leniency

Learning Objectives

- To study the research on effects of smiling

Research conducted by

Marianne LaFrance and Marvin Hecht

Case study prepared by

David Lane

Overview

Dale Carnegie stated that smiling helps win friends and influence people. Research on the effects of smiling has backed this up and shown that a smiling person is judged to be more pleasant, attractive, sincere, sociable, and competent than a non-smiling person.

There is evidence that smiling can attenuate judgments of possible wrongdoing. This phenomenon termed the "smile-leniency effect" was the focus of a study by Marianne LaFrance & Marvin Hecht in 1995.

Questions to Answer

Does smiling increase leniency? Are different types of smiles differentially effective?

Design Issues

There was a single person used for all the conditions. This may limit the generalizeability of the results.

Descriptions of Variables

Table 20.6.1: Description of Variables

Variable	Description	
Smile	1 is false smile 2 is felt smile 3 is miserable smile 4 is neutral control	
Leniency	A measure of how lenient the judgments were.	

Data Files

Leniency.xls

References

- LaFrance, M., & Hecht, M. A. (1995) Why smiles generate leniency. Personality and Social Psychology Bulletin, 21, 207-214.

This page titled 20.6: Smiles and Leniency is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.6: Smiles and Leniency** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.7: Animal Research

Learning Objectives

- Gender difference in attitudes toward the use of animals in research



Research conducted by

Nicole Hilliard, Faculty Advisor: Heidi Ziemer

Case study prepared by

Emily Zitek

Overview

The use of animals in research is a controversial and emotionally charged issue. Personal feelings regarding the use of animals in research vary widely. While many believe that the use of animals in research has been and continues to be essential, others want the practice stopped by cutting off funding or the passing of legislative restrictions. Research on human attitudes toward the use of animals in research has consistently shown systematic differences of opinion with gender differences among the largest.

In this study, a convenience sample of **34** University of Houston - Downtown students completed a simple survey that asked their gender and how much they agreed with the following two statements: "The use of animals in research is wrong," and "The use of animals in research is necessary". They rated their agreement with each of these statements on a **7**-point scale from strongly disagree (**1**) to strongly agree (**7**).

Questions to Answer

Is there a gender difference with respect to the belief that animal research is wrong? Is there a gender difference with respect to the belief that animal research is necessary?

Design Issues

This is self-report data. It is possible that the willingness to admit to thinking animal research is wrong or necessary is what differs by gender, not how the participants actually feel.

Descriptions of Variables

Table **20.7.1**: Description of Variables

Variable	Description
Gender	1 = female, 2 = male
Wrong	high scores indicate that the participant believes that animal research is wrong
Necessary	high scores indicate that the participant believes that animal research is necessary

Data Files

Animals.xls

Links

American Association for the Advancement of Science

References

- Eldridge, J.J. & Gluck, J.P. (1996) Gender differences in attitudes toward animal research. Ethics & Behavior, 6(3), 239-256.
- Nickell, D & Herzog, H.A. (1996). Ethical ideology and moral persuasion: Personal moral philosophy, gender, and judgements of pro- and anti-animal research propaganda. Society & Animals, 4(1), 53-64.
- Pifer, L. K. (1996). Exploring the gender gap in young adults' attitudes about animal research. Society & Animals, 4(1), 37-52.
- Wuensch, K. L. & Poteat, G.M. (1998). Evaluating the morality of animal research: Effects of ethical ideology, gender, and purpose. *Journal of Social Behavior & Personality, 13*(1), 139-151.

This page titled 20.7: Animal Research is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.7: Animal Research** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.8: ADHD Treatment

Learning Objectives

- Treatment Effects of a Drug on Cognitive Functioning in Children with Mental Retardation and ADHD

Research conducted by

Pearson et al. (2003, see reference below)

Case study prepared by

David Lane and Emily Zitek

Overview

This study investigated the cognitive effects of stimulant medication in children with mental retardation and Attention-Deficit/Hyperactivity Disorder. This case study shows the data for the Delay of Gratification (DOG) task. Children were given various dosages of a drug, methylphenidate (MPH) and then completed this task as part of a larger battery of tests. The order of doses was counterbalanced so that each dose appeared equally often in each position. For example, six children received the lowest dose first, six received it second, etc. The children were on each dose one week before testing.

This task, adapted from the preschool delay task of the Gordon Diagnostic System (Gordon, 1983), measures the ability to suppress or delay impulsive behavioral responses. Children were told that a star would appear on the computer screen if they waited "long enough" to press a response key. If a child responded sooner in less than four seconds after their previous response, they did not earn a star, and the 4-second counter restarted. The DOG differentiates children with and without ADHD of normal intelligence (e.g., Mayes et al., 2001), and is sensitive to MPH treatment in these children (Hall & Kataria, 1992).

Questions to Answer

Does higher dosage lead to higher cognitive performance (measured by the number of correct responses to the DOG task)?

Design Issues

This is a repeated-measures design because each participant performed the task after each dosage.

Descriptions of Variables

Table 20.8.1: Description of Variables

Variable	Description
d0	Number of correct responses after taking a placebo
d15	Number of correct responses after taking .15 mg/kg of the drug
d30	Number of correct responses after taking .30 mg/kg of the drug
d60	Number of correct responses after taking .60 mg/kg of the drug

Data Files

ADHD.xls

Links

Methylphenidate

References

- Gordon M (1983), The Gordon Diagnostic System. DeWitt, NY: Gordon Systems
- Hall CW, Kataria S (1992), Effects of two treatment techniques on delay and vigilance tasks with attention deficit hyperactive disorder (ADHD) children. J Psychol 126:17-25
- Mayes SD, Calhoun SL, Crowell, EW (2002), The Gordon Diagnostic System and WISC-III Freedom from Distractibility index: Validity in identifying clinic-referred children with and without ADHD. Psychol Rep ,91, 575-587.
- Pearson DA, Santos CW, Roache JD, Casat CD, Loveland KA, Lachar D, Lane DM, Faria, LF, Cleveland LA (2003), Treatment effects of methylphenidate on behavioral adjustment in children with mental retardation and ADHD. J Am Acad Child Adolesc Psychiatry, 42, 209-216.
- Pearson, D.A., Santos, C.W., Jerger, S.W., Casat, C.D., Roache, J., Loveland, K.A., Lane, D.M., Lachar, D., Faria, L.P., & Getchell, C. (2003) Treatment effects of methylphenidate on cognitive functioning in children with mental retardation and ADHD. Journal of the American Academy of Child and Adolescent Psychiatry, 43, 677-685.

This page titled 20.8: ADHD Treatment is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.8: ADHD Treatment** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.9: Weapons and Aggression

Learning Objectives

- Study of the "Weapons" effect

Research conducted by

Anderson, Benjamin, and Bartholow

Case study prepared by

David Lane

Overview

The "weapons effect" is the finding that the presence of a weapon or even a picture of a weapon can cause people to behave more aggressively. Although once a controversial finding, the weapons effect is now a well-established phenomenon. Based on this, Anderson, Benjamin, and Bartholow (1998) hypothesize that the presence of a weapon-word prime (such as "dagger" or "bullet") should increase the accessibility of an aggressive word (such as "destroy" or "wound"). The accessibility of a word can be measured by the time it takes to name a word presented on computer screen.

The subjects were undergraduate students ranging in age from 18 to 24 years. They were told that the purpose of this study was to test reading ability of various words. On each of the 192 trials, a computer presented a priming stimulus word (either a weapon or non-weapon word) for 1.25 seconds, a blank screen for 0.5 seconds, and then a target word (aggressive or non-aggressive word). Each subject named both aggressive and non-aggressive words following both weapon and non-weapon "primes." The experimenter instructed the subjects to read the first word to themselves and then to read the second word out loud as quickly as they could. The computer recorded response times and computed mean response times for each participant for each of the four conditions.

Examples of the four types of words

- Weapon word primes: shotgun, grenade
- Non-weapon word primes: rabbit, fish
- Aggressive word: injure, shatter
- Non-aggressive word: consider, relocate

Questions to Answer

Does the mere presence of a weapon increase the accessibility of aggressive thoughts? More specifically, can a person name an aggressive word more quickly if it is preceded by a weapon word prime than if it is preceded by a neutral (non-aggressive) word prime?

Design Issues

This is a within-subjects design, and each participant provided four scores to the analysis.

Descriptions of Variables

Table 20.9.1: Description of Variables

Variable	Description
gender	1 = female, 2 = male
aw	The time in milliseconds (msec) to name aggressive word following a weapon word prime.
an	The time in milliseconds (msec) to name aggressive word following a non-weapon word prime.
cw	The time in milliseconds (msec) to name a control word following a weapon word prime.
cn	The time in milliseconds (msec) to name a control word following a non-weapon word prime.

Data Files

Guns.xls

References

- Anderson, C.A., Benjamin, A.J., & Bartholow, B.D. (1998). Does the gun pull the trigger? Automatic priming effects of weapon pictures and weapon names. Psychological Science, 9, 308-314.

This page titled 20.9: Weapons and Aggression is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.9: Weapons and Aggression** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.10: SAT and College GPA

Learning Objectives

- Predicting college GPA from high school scores

Research conducted by

Thomas W. MacFarland

Case study prepared by

Emily Zitek

Overview

When deciding whether to admit an applicant, colleges take lots of factors, such as grades, sports, activities, leadership positions, awards, teacher recommendations, and test scores, into consideration. Using SAT scores as a basis of whether to admit a student or not has created some controversy. Among other things, people question whether the SATs are fair and whether they predict college performance.

This study examines the SAT and GPA information of **105** students who graduated from a state university with a B.S. in computer science. Using the grades and test scores from high school, can you predict a student's college grades?

Questions to Answer

Can the math and verbal SAT scores be used to predict college GPA? Are the high school and college GPAs related?

Design Issues

The conclusions from this study should not be generalized to students of other majors.

Descriptions of Variables

Table **20.10.1**: Description of Variables

Variable	Description
high_GPA	High school grade point average
math_SAT	Math SAT score
verb_SAT	Verbal SAT score
comp_GPA	Computer science grade point average
univ_GPA	Overall university grade point average

Data Files

SAT.xls

Links

Want a job? Hand over your SAT results! Is the SAT a fair test?

References

- None

This page titled 20.10: SAT and College GPA is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.10: SAT and College GPA** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.11: Stereograms

Learning Objectives

- Study to determine the effects of information for an embedded image given ahead of time to a person

Research conducted by

Frisby, J. P. and Clatworthy, J.L.

Case study prepared by

Emily Zitek from DASL story contributed by Michael Friendly

Overview

The rectangles below appear to be composed of random dots. However, if the images are viewed with a stereo viewer, the separate images will fuse and reveal an embedded ***3D*** figure. In this example, fusing the images of these random dot stereograms will reveal a diamond. (Another way for you to fuse the images is to fixate on a point in between them and defocus your eyes. This technique takes practice, but you can try it out with the links below.)

This experiment sought to determine whether giving someone information about the embedded image can help speed up how long it takes to view this image. Seventy-eight participants were given no information, verbal information, and/or visual information (a drawing of the object) about what the embedded image should look like before attempting to fuse the images and actually view the 3D design.



Figure **20.11.1**: Random dots form an embedded image when viewed with a stereo viewer

Questions to Answer

Does giving someone information about an embedded image in a stereogram affect the amount of time it takes to see this image? More specifically, does the amount of time it takes to fuse the image in a stereogram differ when the person is given both verbal and visual information about what the image should look like as opposed to when the person is only given verbal information or no information at all?

Descriptions of Variables

Table **20.11.1**: Description of Variables

Variable	Description
Time	Time to produce a fused image of the random dot stereogram
Group	Treatment group divided by type of information received: 1 = no information or only verbal information 2 = both verbal and visual information

Data Files

Fusion.xls

Links

View random dot stereograms. Information about random dot stereograms

References

- Frisby, J. P. & Clatworthy, J.L., (1975) Learning to see complex random-dot stereograms, Perception, 4, 173-178.

This page titled 20.11: Stereograms is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.11: Stereograms** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.12: Driving

Learning Objectives

- Driving in inclement weather







Research conducted by

Darin Baskin

Case study prepared by

Emily Zitek

Overview

Many people believe that weather patterns influence driving safety. As a result, there are many web sites and other publications dedicated to giving people tips about how to drive in various weather conditions (see references and links below). Additionally, car accidents are often attributed to bad weather (e.g., see Taylor & Quinn, 1991). This study examines the beliefs and behaviors of people with respect to the important topic of driving in inclement weather.

The participants in this study filled out a questionnaire consisting of some demographic questions and then questions asking about their transportation habits and other beliefs concerning inclement weather. This questionnaire was administered to a convenience sample of **61** University of Houston - Downtown students at various locations (i.e., classrooms, hallways, and the food court).

Questions to Answer

Is gender or age related to the likelihood of driving in inclement weather? Does the number of accidents that someone thinks occur during inclement weather relate to how often he or she takes public transportation or chooses to drive during inclement weather?

Design Issues

This is a correlational study, so we cannot infer causation.

Descriptions of Variables

Variable	Description
Age	The age of the participant in years
Gender	1 = female, 2 = male
Cho2drive	How often he or she chooses to drive in inclement weather 1 = always, 3 = sometimes, 5 = never
Pubtran	% of travel time spent on public transportation in inclement weather
Accident	% of accidents thought to occur from driving in inclement weather

Data Files

Driving.xls

Links

Driving on Wet Roads. Jokes about Driving in Inclement Weather.

References

- Galski, T., Ehle, H. T, & Bradley, W. J. (1998). Estimates of driving abilities and skills in different conditions. American Journal of Occupational Therapy, 52, 268-275.
- Griffin, J., & Murdock, G. (1993, August). Wet weather driving. Consumers' Research Magazine, 76, 2.
- Taylor, G. W., & Quinn, H. (1991, January 14). An arctic winter rage. Maclean's, 104, 12-13.

This page titled 20.12: Driving is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.12: Driving** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.13: Stroop Interference

Learning Objectives

- Strrop Interference demonstration

Research conducted by

Statistics Class

Case study prepared by

David Lane

Overview

Naming the ink color of color words can be difficult. For example, if asked to name the color of the word "blue" is difficult because the answer (red) conflicts with the word "blue." This interference is called "Stroop Interference" after the researcher who first discovered the phenomenon.

This case study is a classroom demonstration. Students in an introductory statistics class were each given three tasks. In the "words" task, students read the names of 60 color words written in black ink; in the "color" task, students named the colors of 60 rectangles; in the "interference" task, students named the ink color of 60 conflicting color words. The times to read the stimuli were recorded. There were 31 female and 16 male students.

Questions to Answer

Is naming conflicting color names faster or slower than naming color rectangles? Which is faster, naming color rectangles or reading color names? Are there gender differences?

Design Issues

This was not a well-controlled experiment since it was just a classroom demonstration. The order in which the students performed the tasks may not have been counterbalanced or randomized.

Descriptions of Variables

Table 20.13.1: Description of Variables

Variable	Description
Gender	1 for female, 2 for male
Words	Time in seconds to read 60 color words
Colors	Time in seconds to name 60 color rectangles
Interfer	Time in seconds to name colors of conflicting words

Data Files

Stroop.xls

Links

Full text of the above reference.

References

- Stroop, J.R. (1935). Studies of interference in serial verbal reactions. Journal of Experimental Psychology, 28, 643-662.

This page titled 20.13: Stroop Interference is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.13: Stroop Interference** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.14: TV Violence

Learning Objectives

- Does Television Viewing Encourage Aggression in Children?



Research conducted by

Mariana Fernandez, University of Houston-Downtown undergraduate

Case study prepared by

Nichole Rivera

Overview

How much television is too much for children? Television advocates espouse the educational benefits that children may reap from instructive programming. However, many researchers say that excess television watching may contribute to aggressive behavior in children. Young boys, in particular may be susceptible to this effect. What are the effects, if any, on children's behavior when television is used as a babysitter?

In a survey of University of Houston-Downtown students, parents reported their children's age, characteristic behavior, and television viewing habits. Convenience sampling was used to gather 30 subjects ($N = 30$).

Questions to Answer

Is there a relationship between hours of television watched and child's obedience? Will a child be more or less aggressive if he/she watches a lot of television?

Design Issues

This survey offered a very limited sample ($N = 30$), which was further hindered by reporting participants' filling out an individual survey for each individual child. This contributes to some lack of true variability in responses because participants tended to report similar behavior for each child. This may magnify errors associated with self-reported data. The sample would provide greater reliability if each participant reported on only one child's behavior.

The survey has broad questions which do not provide much context for reported behaviors. In some instances aggression may be positively rated, but this survey treats all aggression as a negative characteristic. In addition, the instrument itself measures largely nominal data, making in depth analysis difficult.

Descriptions of Variables

Table 20.14.1: Description of Variables

Variable	Description
TV hours	Total number of TV hours watched per day
Obedience	How obedient the child is 1 = very obedient, 5 = not obedient
Attitude	Attitude while playing with other children 1 = non-aggressive, 5 = very aggressive

Data Files

TV.xls

Links

TV Guide - Mighty Morphin' Power Rangers v. Teenage Mutant Ninja Turtles

References

- Boyatzis, Chris J. and Matillo Gina M. (1995). Effects of "The Mighty Morphin Power Rangers" on Children's Aggression with Peers. Child Study Journal, 25 (1), 45-57.
- Charlton, Davie. (2001). Monitoring Children's Behavior in Remote Community Before and Six Years After the Availability of Broadcast TV. North America Journal of Psychology, 3, 429-441.
- Huesmann, Rowell L., Moise-Titus, Jessica, Podolski, Cheryl-Lynn, Eron, Leonard D. (2003). Longitudinal Relations Between Children's Exposure to TV Violence and their Aggressive and Violent Behavior in Young Adulthood: 1977-1992. Developmental Psychology 39(2), 201-221.
- Troseth, Georgene L. (2003). TV Guide: Two-Year-Old Children Learn to Use Video as a Source of Information. Developmental Psychology 39 (1), 140-150.

This page titled 20.14: TV Violence is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.14: TV Violence** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.15: Obesity and Bias

Learning Objectives

- Bias Against Associates of the Obese

Research conducted by

Mikki Hebl and Laura Mannix

Case study prepared by

Emily Zitek

Overview

Obesity is a major stigma in our society. People who are obese face a great deal of prejudice and discrimination. For example, Roehling (1999) showed that obese people experience a lot of discrimination in the workplace (e.g., they are less likely to be hired and get lower wages). We know that people who are obese are stigmatized, but what about people who are somehow associated with an obese person? Neuberg et al. (1994) found that friends of gay men and lesbians suffer from "stigma by association". Perhaps the negative effects of the obesity stigma can also spread to other people. This study seeks to examine how the stigma of obesity can spread to a job applicant of average weight.

As part of a larger study, participants had to rate how qualified a particular job applicant was. This applicant was sitting by a woman. The researchers manipulated the following two variables: the weight of the woman and the relationship between the woman and the applicant. The woman was either obese or of average weight. This woman was also portrayed as being the applicant's girlfriend or a woman simply waiting to participate in a different experiment.

Questions to Answer

Are male applicants who are seated next to an obese woman rated as less qualified for a job? Are applicants who are seated next to their girlfriend rated differently from applicants seated next to a woman with whom they do not have an intimate relationship? Finally, does the effect of the type of relationship differ depending on the weight of the woman?

Design Issues

This study only looked how at how an obese woman seated next to a male job applicant could affect qualification ratings. Future research could address other gender combinations.

Descriptions of Variables

Table 20.15.1: Description of Variables

Variable	Description
Weight	The weight of the woman sitting next to the job applicant 1 = obese, 2 = average weight
Relate	Type of relationship between the job application and the woman seated next to him 1 = girlfriend, 2 = acquaintance (waiting for another experiment)
Qualified	Larger numbers represent higher professional qualification ratings

Data Files

Weight2.xls

Links

The Obesity Society

References

- Hebl, M. R., & Mannix, L. M. (2003). The weight of obesity in evaluating others: A mere proximity effect. Personality and Social Psychology Bulletin, 29, 28-38.
- Neuberg, S. L., Smith, D. M., Hoffman, J. C., & Russell, F. J. (1994). When we observe stigmatized and "normal" individuals interacting: Stigma by association. Personality and Social Psychology Bulletin, 20, 196-209.
- Roehling, M. (1999). Weight-based discrimination in employment: Psychological and legal aspects. Personnel Psychology, 52, 969-1016.

This page titled 20.15: Obesity and Bias is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.15: Obesity and Bias** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.16: Shaking and Stirring Martinis

Learning Objectives

- To test the difference between shaken and stirred martinis



Research conducted by

This is just made up data.

Case study prepared by

David Lane

Overview

This is an example to illustrate hypothesis testing and the binomial distribution. The statistician R. Fisher explained the concept of hypothesis testing with a story of a lady tasting tea. Here is an example based on James Bond who insisted that Martinis should be shaken rather than stirred. In this hypothetical experiment to determine whether Mr. Bond could tell the difference between a shaken and a stirred martini, we gave Mr. Bond a series of **16** taste tests. In each test, we flipped a fair coin to determine whether to stir or shake the martini. Then we presented the martini to Mr. Bond and asked him to decide whether it was shaken or stirred. Mr. Bond was correct on **13/16** trials.

Questions to Answer

Does Mr. Bond have the ability to tell the difference between a Martini that is shaken and one that is stirred?

Design Issues

This is only a made-up study.

Descriptions of Variables

Table **20.16.1**: Description of Variables

Variable	Description
Y	0 = incorrect, 1 = correct

Data Files

Martini.xls

Links

The Lady Tasting Tea

References

- Salsburg, D. (2002) The Lady Tasting Tea: How Statistics Revolutionized Science in the Twentieth Century. Owl Books

---

This page titled 20.16: Shaking and Stirring Martinis is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.16: Shaking and Stirring Martinis** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.17: Adolescent Lifestyle Choices

Learning Objectives

- Adolescents and Healthy Lifestyle Choices

Research conducted by

Ka He, Ellen Kramer, Robert F. Houser, Virginia R. Chomitz, and Karen A. Hacker

Case study prepared by

Robert F. Houser, Alyssa Koomas, and Georgette Baghdady

Overview

Teen pregnancy, sexually transmitted disease, drug abuse, and suicide are some of the behaviorally-mediated negative health outcomes that can occur during adolescence. Identifying the characteristics of adolescents who are able to make healthy lifestyle choices is imperative toward understanding positive health behaviors in this age group. This information could be used to develop targeted interventions that support at-risk adolescents in making healthy lifestyle choices, and hopefully prevent such negative outcomes.

This study collected survey data from **1487** high school students in an urban Massachusetts community. The survey assessed health-related behaviors, stressful events, demographics, familial characteristics, perceptions of peer and parental support, and academic performance. In collaboration with community stakeholders and parents, the researchers selected six health-related behaviors and developed two sets of criteria to define positive health behaviors. One set used “strict” definitions, namely, not drinking alcohol in the last **30** days, no attempted suicide in the past **12** months, and no experience at all with tobacco, hard illegal drugs, marijuana, and sexual partners. The second set used “broad” definitions that allowed for mild use and safe experimentation (except for suicidal behavior). Students who adhered to all six health-related behaviors according to the “strict” definitions formed one subgroup for analysis, and those who reported behaviors in accordance with the “broad” definitions formed another subgroup. These two lifestyle subgroups were analyzed separately in relation to the personal and social-environmental factors assessed by the survey.

Questions to Answer

What personal and social-environmental characteristics are associated with adolescents who practice healthy lifestyle behaviors according to the “strict” definitions? How much more likely are adolescents with these characteristics to be practicing healthy behaviors than adolescents without these characteristics?

Design Issues

The results of this study may not be applicable to adolescents in non-urban schools, as the sample was drawn from a diverse, urban school. As well, the definitions that make up positive health behaviors may vary by region and social group. Adolescents self-reported their health-related behaviors and other information via the survey. Missing responses may have caused bias in the results.

Descriptions of Variables

Table 20.17.1: Description of Variables

Variable	Description
Healthy behaviors based on the “strict” definitions	Whether or not the adolescent practices all 6 health-related behaviors according to the “strict” definitions
Immigration status	Whether the adolescent was born in the US or is an immigrant
Stress score	An index from 0 to 14 assessing 14 possible stressful events in the adolescent’s life, such as failing grades, moving, death in the family, divorce in the family, abuse, and violence
Stress index	Whether the adolescent’s stress score is at or above the median stress score of 2, or below
Academic performance	The adolescent’s average academic letter grade (A, B, C, D, F)

Links

1 in 3 Teens Text While Driving

2011 Youth Risk Behavior Surveillance Survey

References

- He, K., Kramer, E., Houser, R. F., Chomitz, V. R., Hacker, K. A. (2004). Defining and understanding healthy lifestyles choices for adolescents. Journal of Adolescent Health, 35, 26-33.

This page titled 20.17: Adolescent Lifestyle Choices is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.17: Adolescent Lifestyle Choices** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.18: Chocolate and Body Weight

Learning Objectives

- To study chocolate's healthful metabolic mechanisms

Research conducted by

Beatrice A. Golomb, Sabrina Koperski, and Halbert L. White

Case study prepared by

Robert F. Houser and Georgette Baghdady

Overview

Recent research has brought to light the beneficial health effects of chocolate. Studies have linked chocolate with lower blood pressure, lower bad cholesterol, improved insulin sensitivity, and reductions in the risks of diabetes, heart disease, and stroke. The authors of this study hypothesized that chocolate's healthful metabolic mechanisms might also reduce fat deposition in spite of its high caloric content.

This study used the baseline data from a clinical study that examined noncardiac effects of cholesterol-lowering drugs in healthy adults. The baseline data included body mass index (BMI), chocolate consumption frequency, age, sex, physical activity frequency, depression, and some dietary variables. Chocolate consumption frequency was assessed with the question: "How many times a week do you consume chocolate?" Dietary intakes of total calories, fruits and vegetables, and saturated fat were assessed with a validated food frequency questionnaire. A food frequency questionnaire is a limited checklist of foods and beverages with a frequency response section for subjects to report how often each item was consumed over a specified period of time. Depression was measured with a validated scale related to mood. BMI is a measure of body fatness that is associated with many adverse health conditions.

Questions to Answer

What can we conclude from the researchers' findings that there is an association between consuming chocolate frequently and lower BMI? How do we interpret regression models?

Design Issues

The authors used baseline data from an unrelated clinical study examining noncardiac effects of cholesterol-lowering drugs. That clinical study included men ranging in age from 20 to 85 years, but only postmenopausal women. The results of the chocolate study cannot, therefore, be generalized to younger adult women. Except for BMI, the data for all of the study variables were "self-reported" by the subjects via questionnaires. The assessment of critical variables, such as chocolate consumption frequency and vigorous physical activity frequency, could differ when using different measurement tools. The study was cross-sectional in nature, precluding conclusions about causation.

Descriptions of Variables

Table 20.18.1: Description of Variables

VARIABLE	DESCRIPTION
BMI	Body mass index, calculated as: (weight in kilograms) / (height in meters)^2
Chocolate consumption frequency	Number of times per week a subject consumed chocolate
Calories	Overall calorie intake of a subject determined via food frequency questionnaire
Age	Range of 20 to 85 years, postmenopausal if female
Sex	68% male, 32% female
Activity	Number of times per 7-day period a subject engaged in vigorous physical activity for at least 20 minutes

Links

Golomb et al. article

Rose et al. article

What is body mass index (BMI)?

References

- Golomb, B. A., Koperski, S., White, H. L. (2012). Association between more frequent chocolate consumption and lower body mass index. Archives of Internal Medicine, 172, 519-521.
- Rose, N., Koperski, S., Golomb, B. A. (2010). Mood food: chocolate and depressive symptoms in a cross-sectional analysis. Archives of Internal Medicine, 170, 699-703.

This page titled 20.18: Chocolate and Body Weight is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.18: Chocolate and Body Weight** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.19: Bedroom TV and Hispanic Children

Learning Objectives

- Study of overweight and obesity in Hispanic children

Research conducted by

Du Feng, Debra B. Reed, M. Christina Esperat, and Mitsue Uchida

Case study prepared by

Robert F. Houser, Alyssa Koomas, and Georgette Baghdady

Overview

The prevalence of overweight and obesity in children in the U.S. is a growing public health concern that disproportionately affects Hispanic youth. As noted by the authors, in 2005 to 2006, 15.5% of all U.S. children aged 2 to 19 years were overweight or obese, compared with 23.2% for boys and 18.5% for girls among Mexican-Americans in this age group. Past research has revealed diverse environmental and behavioral factors that may contribute to this disparity. For example, studies have shown that Hispanic children watch more television than white children.

This study examined TV viewing among 314 Hispanic children aged 5 to 9 years in West Texas and the possible effects of having a TV in the child's bedroom. Children's weights and heights were measured, body mass indexes (BMI) calculated, and sex- and age-adjusted BMI percentiles obtained. The 2000 CDC Growth Charts were used to assess whether or not a child was overweight or at risk for becoming overweight. Their parents completed a family survey assessing demographics, acculturation, parental support of physical activity, dietary practices, the presence of a TV in the participating child's bedroom, and the child's TV/DVD viewing time.

Questions to Answer

Do children with a TV in their bedroom spend more time watching TV/DVDs on a daily basis than children without a TV in their bedroom? Do children with a TV in their bedroom have less support from their parents for physical activity than children without a TV in their bedroom? What might account for missing responses to survey questions?

Design Issues

Except for BMI, the data for all of the study variables were "self-reported" by the parents. The study used a cross-sectional design, which cannot be relied upon to provide conclusive evidence of causal relationships.

Descriptions of Variables

Table 20.19.1: Description of Variables

VARIABLE	DESCRIPTION
TVIB, No TVIB	Presence or absence of a TV in the participating child's bedroom
Daily TV/DVD time	Average number of hours the child spent watching TV and DVDs per day
Parental support of physical activity	Scale score calculated as the average of parent's responses to 8 survey items assessing the parent's support of physical activity for the child. Items rated on 4-point Likert scale (0 = never, 3 = always). Research has shown a significant positive relationship between parental support of physical activity and children's physical activity level
Daily fruit and vegetable intake	Average number of cups of fruits and vegetables (fresh, frozen, dried, canned, and 100% juice) consumed by the child per day
Daily sweetened beverages	Average number of ounces of soda, fruit drink, sports drink, tea, and lemonade consumed by the child per day

Links

New York Times article

Television and Children information guide

References

- Feng, D., Reed, D. B., Esperat, M. C., Uchida, M. (2011). Effects of TV in the bedroom on young Hispanic children. American Journal of Health Promotion, 25, 310-318.

This page titled 20.19: Bedroom TV and Hispanic Children is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.19: Bedroom TV and Hispanic Children** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.20: Weight and Sleep Apnea

Learning Objectives

- Excess Body Weight and Sleep Apnea

Research conducted by

Kari Johansson, Erik Hemmingsson, Richard Harlid, Ylva Trolle Lagerros, Fredrik Granath, Stephan Rössner, and Martin Neovius

Statistical article authored by

Philip Sedgwick

Case study prepared by

Robert F. Houser and Georgette Baghdady

Overview

In his statistical article, "Standard deviation versus standard error," UK researcher Philip Sedgwick presents us with an interesting discussion of the proper use of standard deviation (SD) and standard error of the mean (SEM). He uses an example of a weight loss study of 63 obese men suffering from obstructive sleep apnea who were being treated with continuous positive airway pressure (CPAP). The weight loss program lasted one year. Outcome measures included change in body weight measured in kilograms (kg).

More than 60% of people experiencing obstructive sleep apnea are obese. CPAP therapy is the most common treatment. It uses a machine and mask to prevent the airway from collapsing, thus enabling a person to breathe more easily during sleep. Weight loss is an effective treatment for sleep apnea.

Questions to Answer

What is the proper use of the SD? What is the proper use of the SEM?

Design Issues

None for the Sedgwick article.

Descriptions of Variables

Table 20.20.1: Description of Variables

Variable	Description
Weight	Body weight at baseline in kg
Weight change	Change in body weight at one year from baseline in kg

Links

What Is Sleep Apnea?

t Table (two-tailed) for significance and calculation of confidence interval

Johansson et al. article

References

- Sedgwick, P. (2011). Standard deviation versus standard error. BMJ, 343, d8010.
- Johansson, K., Hemmingsson, E., Harlid, R., Lagerros, Y. T., Granath, F., Rössner, S., Neovius, M. (2011). Longer term effects of very low energy diet on obstructive sleep apnoea in cohort derived from randomised controlled trial: prospective

observational follow-up study. BMJ, 342, d3017.

This page titled 20.20: Weight and Sleep Apnea is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.20: Weight and Sleep Apnea** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.21: Misusing SEM

Learning Objectives

- Misusing Standard Error of the Mean (SEM)

Research conducted by

Peter Nagele

Case study prepared by

Robert F. Houser, Georgette Baghdady, and Jennifer E. Konick

Overview

Authors of published research articles often erroneously use the standard error of the mean to describe the variability of their study sample. Nagele demonstrated this misuse of the standard error of the mean as a descriptive statistic by manually searching four leading anesthesia journals in 2001.

Here are quotes on key points from Nagele's article and our notes:

"Descriptive statistics aim to describe a given study sample without regard to the entire population."

"If normally distributed, the study sample can be described entirely by two parameters: the mean and the standard deviation (SD)." However, a study sample variable is never exactly normally distributed. When a variable is close to normally distributed, the mean and median are quite similar. Therefore, the mean and SD would be sufficient.

"The SD represents the variability within the sample." It tells us about "the distribution of individual data points around the mean." The latter statement, however, is a generalization since the SD cannot tell us exactly where each data point lies relative to the mean.

"[I]nferential statistics generalize about a population on the basis of data from a sample of this population."

The standard error of the mean (SEM) "is used in inferential statistics to give an estimate of how the mean of the sample is related to the mean of the underlying population." It "informs us how precise our estimate of the [population] mean is."

Thus, "the SEM estimates the precision and uncertainty [with which] the study sample represents the underlying population."

The standard error of the mean is calculated by dividing the sample standard deviation by the square root of the sample size ($SEM = SD/\sqrt{n}$).

"[T]he SEM is always smaller than the SD." However, this is only true as long as the sample size is greater than 1.

"In general, the use of the SEM should be limited to inferential statistics [for which] the author explicitly wants to inform the reader about the precision of the study, and how well the sample truly represents the entire population [of interest]." A sample never truly represents the population.

Questions to Answer

How prevalent is the inappropriate use of the SEM in describing the variability of the study sample in research publications? What is the proper use of the SEM?

Design Issues

The author focused on four leading anesthesia journals in his field of expertise. The misapplication of the SEM in descriptive statistics can be found in professional journals of many, if not all, fields of research.

Descriptions of Variables

Table 20.21.1: Description of Variables

Variable	Description
Incorrect use of SEM; total	Total frequency of misuse of SEM; expressed as number of articles and percent
Laboratory studies using SEM incorrectly	A subset of the above variable; expressed as number of articles and percent
Correct use of SD	Frequency of correct use of standard deviation; expressed as number of articles and percent

Data Files

Sem.xls

Links

Nagele article

References

- Nagele, P. (2003). Misuse of standard error of the mean (SEM) when reporting variability of a sample. A critical evaluation of four anaesthesia journals. British Journal of Anaesthesia, 90, 514-516.
- Hassani, H., Ghodsi, M., Howell, G. (2010). A note on standard deviation and standard error. Teaching Mathematics and Its Applications, 29, 108-112.
- Altman, D. G., Bland, J. M. (2005). Standard deviations and standard errors. BMJ, 331, 903.

This page titled 20.21: Misusing SEM is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.21: Misusing SEM** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.22: School Gardens and Vegetable Consumption

Learning Objectives

- School garden program benefits

Research conducted by

Michelle M. Ratcliffe, Kathleen A. Merrigan, Beatrice L. Rogers, and Jeanne P. Goldberg

Case study prepared by

Robert F. Houser and Georgette Baghdady

Overview

School garden programs are gaining popularity because of their numerous benefits for children: outdoor exercise, social skills, connecting with nature, environmental stewardship, active learning, experiential science education, higher academic achievement, and transformed attitudes and habits related to fruits and vegetables. By integrating the regular science class with gardening activities in which students plant, nurture, harvest, prepare, and consume produce grown in the schoolyard, studies are showing that garden-based learning can improve children's consumption of fruits and vegetables.

This study investigated the impact of participating in a school garden program on the ability to identify, willingness to taste, preference for, and consumption of vegetables. Subjects were 320 sixth-grade students aged 11 to 13 years at two intervention schools and one control school. At the intervention schools, garden-based learning activities were incorporated into the regular science class for a period of four months. The control school did not include a garden program as part of its science class. Two questionnaires – Garden Vegetable Frequency Questionnaire and taste test – assessed the outcome variables using vegetables typically grown in school gardens that were also ethnically and culturally appropriate for the study population. The Garden Vegetable Frequency Questionnaire assessed the types of vegetables consumed the day before as well as usual consumption frequency. The taste test involved tasting five raw vegetables (carrots, string beans, snow peas, broccoli, and Swiss chard). Both questionnaires were administered at the outset and end of the study. Change scores (posttest minus pretest) were compared between the garden (intervention) group and the control group.

Questions to Answer

Do hands-on school garden programs increase vegetable consumption in children? What are some of the potential sources of bias in research studies?

Design Issues

This study used a "quasi-experimental" design, which differs from an experiment in that the students were selected and assigned to the intervention group and control group by a method other than random assignment. With this type of design, there is a greater chance that the intervention and control groups might differ at the outset of the study in ways that could bias the results of the study. Since the study population was middle-school students living in low-income, urban communities, the results of the study cannot be generalized to other settings. The study did not measure the actual amounts of vegetables that students consumed, so no conclusions can be drawn about number or size of servings.

Descriptions of Variables

Tables* 20.22.1*: *Description of Variables*

Variable	Description
School garden program group	Garden (intervention) and Control groups: Whether or not a student experiences hands-on gardening activities at school
Consumption of vegetables at school	Assessed by the taste test, it measures whether or not a student ate each of five specific vegetables at school
Consumption of vegetables at home	Assessed by the taste test, it measures whether or not a student also ate each of the five specific vegetables at home

Links

Ratcliffe et al. article

The benefits of school gardens

First Lady Michelle Obama Hosts White House Garden Spring 2011 Planting

References

- Ratcliffe, M. M., Merrigan, K. A., Rogers, B. L., Goldberg, J. P. (2011). The effects of school garden experiences on middle school-aged students' knowledge, attitudes, and behaviors associated with vegetable consumption. Health Promotion Practice, 12, 36-43.

This page titled 20.22: School Gardens and Vegetable Consumption is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.22: School Gardens and Vegetable Consumption** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.23: TV and Hypertension

Learning Objectives

- TV viewing time and adverse health

Research conducted by

Perrie E. Pardee, Gregory J. Norman, Robert H. Lustig, Daniel Preud'homme, and Jeffrey B. Schwimmer

Case study prepared by

Robert F. Houser and Andrew Kennedy

Overview

A strong, evidence-based association exists between TV viewing time and the risk of being obese in children and adolescents. Little or no research, however, has explored adverse health outcomes associated with TV viewing among obese children. This study aimed at identifying whether or not time spent watching TV is associated with hypertension (high blood pressure) in obese children.

Obese children aged 4 to 17 years were recruited and evaluated at three pediatric centers. Obesity was defined as a body mass index (BMI) greater than or equal to the 95th percentile for the child's age and gender.

Questions to Answer

Is TV watching associated with hypertension in obese children?

Design Issues

The study involved a cross-sectional design, which prevented the determination of possible causality among the associations found. There could be unmeasured factors that play a role in the association between TV viewing and hypertension.

Descriptions of Variables

Table $20.23.1$: Description of Variables

Variable	Description
Hypertension	Defined as a systolic and/or diastolic blood pressure greater than or equal to the 95th percentile for the child's age, gender, and height
Age	A child's age in years
BMI	A child's body mass index, calculated as: (weight in kilograms) / (height in meters)^2
Hours of TV/day	An estimate of a child's average daily time spent watching TV in hours

Links

Pardee et al. article

Luma et al. article

References

- Pardee, P. E., Norman, G. J., Lustig, R. H., Preud'homme, D., Schwimmer, J. B. (2007). Television viewing and hypertension in obese children. American Journal of Preventive Medicine, 33, 439-443.
- Luma, G. B., Spiotta, R. T. (2006). Hypertension in children and adolescents. American Family Physician, 73, 1558-1568.

This page titled 20.23: TV and Hypertension is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.23: TV and Hypertension** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.24: Dietary Supplements

Learning Objectives

- Dietary supplements and health risk behaviors

Research conducted by

Wen-Bin Chiou, Chao-Chin Yang, and Chin-Sheng Wan

Case study prepared by

Robert F. Houser and Georgette Baghdady

Overview

Although the dietary-supplement market in the U.S. is enormous, there is no apparent association between the use of dietary supplements and improved public health. The researchers of this study explored this paradox under the hypothesis that taking dietary supplements triggers a phenomenon called the "licensing effect," namely, the tendency for positive choices to license subsequent self-indulgent, risky or unhealthful choices. The researchers hypothesized that supplement use confers "perceived health credentials," leading people to feel invulnerable to health hazards and thus more likely to engage in risky, health-related behaviors.

The study involved two experiments. In the first experiment, 82 participants were randomly assigned to either a vitamin-pill (multivitamin) group or control (placebo) group and were told the kind of pill they would be taking. However, only the control group was given correct information. In actuality, both groups received the placebo pill. After taking the pills, the participants completed a survey on leisure-time activities, rating the desirability of nine hedonic (pleasurable) activities, such as excessive drinking and wild parties, and nine exercise activities, such as yoga and running, on 7-point scales. The survey also included a general invulnerability scale to assess a participant's perceived invulnerability to harm and disease. After completing the survey, the participants were offered a free lunch, choosing freely between a buffet and a healthful, organic meal.

The second experiment involved different participants. The vitamin-pill (multivitamin) group again unknowingly took placebo pills. After completing a questionnaire that included the general invulnerability scale and reading a medical report on the health benefits of walking, the distance participants walked in one hour was measured with a pedometer.

Questions to Answer

Does taking dietary supplements disinhibit unhealthy behaviors, such as eating unhealthful meals? Is the study sufficiently powered to detect significant differences between males and females?

Design Issues

The research was conducted in Taiwan, where cultural attitudes and behaviors related to dietary supplements may differ from those in the U.S. It is possible that the results might not generalize to other countries, so more research is needed. Participants in Experiment 1 had a wide range in age, from 18 to 46 years, with a mean (SD) of 30.9 (7.8) years. It would be helpful to consider age in the analysis, especially if age is associated with invulnerability scores. Leisure-time activities and invulnerability were assessed only post-intervention; future studies should also measure these variables before the intervention to see if the two groups had similar scores at the start of the study. The general invulnerability scale used to assess perceived invulnerability to harm and disease has been validated only for adolescents.

Descriptions of Variables

Table 20.24.1: Description of Variables

VARIABLE	DESCRIPTION
Experimental condition	Vitamin-pill (multivitamin) condition or Control (placebo) condition
Meal choice	Either a buffet meal or a healthful, organic meal
Gender	The sex of participants

Links

The licensing effect

No Significant Difference … Says Who?

References

- Chiou, WB, Yang, CC, Wan, CS. (2011). Ironic effects of dietary supplementation: Illusory invulnerability created by taking dietary supplements licenses health-risk behaviors. Psychological Science, 22, 1081-1086.
- Trout, A. T., Kaufmann, T. J., Kallmes, D. F. (2007). No significant difference … Says who? Editorial. American Journal of Neuroradiology, 28, 195-197.

This page titled 20.24: Dietary Supplements is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.24: Dietary Supplements** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.25: Young People and Binge Drinking

Learning Objectives

- Binge drinking and serious public health problems

Research conducted by

Richard O. de Visser and Julian D. Birch

Case study prepared by

Robert F. Houser and Georgette Baghdady

Overview

Binge drinking is a serious public health problem bringing harm to both the individual and society. It compromises a person's health, increasing the risk of many diseases, injury, and death. It also results in a greater incidence of motor vehicle crashes, violence, the spread of sexually-transmitted diseases, and unintended pregnancies. Binge drinking is prevalent among both young and older adults, men and women, and high and low income levels. Governments have formulated guidelines for moderate or sensible drinking levels. The government of the United Kingdom (UK) issued guidelines for sensible drinking as $2-3$ alcohol units per day for women and $3-4$ units per day for men, an alcohol unit being 10 milliliters of ethanol. A binge drinking episode is when a person drinks above double the recommended daily guidelines in a short period of time.

Questions to Answer

What can we learn about the binge drinking patterns of university students in England? Do the bingers and non-bingers differ in their knowledge of the sensible drinking guidelines issued by the UK government?

Design Issues

The university students in the sample "self-selected" to participate in the study by responding to recruiting efforts made via email messages and requests in lectures.

Descriptions of Variables

Table 20.25.1: Description of Variables

Variable	Description
Sex	Female or male
mo_binge_n	Number of times the university students did binge drinking in the last month (using sex-specific definitions)
modrunk	Number of times the university students drank in the last month
wk_unit_prop	Familiarity with alcohol unit-based guidelines (measured on a 5-point scale)
k_unit_sum	Knowledge of alcohol unit-based guidelines (score out of 7)
u_fam	Familiarity with alcohol unit-based guidelines (measured on a 5-point scale)

Data Files

Binge.xls

Links

de Visser et al. article

References

- de Visser, R. O., Birch, J. D. (2012). My cup runneth over:Young people's lack of knowledge of low-risk drinking guidelines. Drug and Alcohol Review, 31, 206-212.

This page titled 20.25: Young People and Binge Drinking is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.25: Young People and Binge Drinking** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.26: Sugar Consumption in the US Diet

Learning Objectives

- Sugar Consumption in the US Diet between 1822 and 2005

Research conducted by

Stephan Guyenet and Jeremy Landen

Case study prepared by

Robert F. Houser and Georgette Baghdady

Overview

Sugar has many forms: cane sugar, beet sugar, honey, molasses, fruit juice concentrate, glucose, sucrose, fructose, high-fructose corn syrup, maple syrup, brown rice syrup, barley malt syrup, agave nectar, to list a few. High-fructose corn syrup, in particular, was introduced into the US food industry in the early $1970s$ and has become ubiquitous in processed foods and soft drinks. Many of the added sugars in packaged foods and beverages could be considered "hidden sugar" because, if we do not examine the ingredients list on food labels or know sugar's many aliases, we are most likely unaware of how much sugar we consume each day.

To explore sugar consumption trends in the US, researchers Stephan Guyenet and Jeremy Landen compiled data on caloric sweetener sales spanning 184 years. They extracted annual caloric sweetener sales per capita for 1822 to 1908 from US Department of Commerce and Labor reports, and for 1909 to 2005 from the US Department of Agriculture (USDA) web site. The researchers adjusted the sales data for post-production losses using the USDA's $1970-2005$ loss estimate of 28.8 percent to obtain reasonable estimates of annual per capita consumption of added sugars. Post-production losses of a food commodity occur at the retail, foodservice and consumer levels from, for example, spoilage, pests, cooking losses and plate waste.

Guyenet presents a striking graph and regression analysis of sugar consumption in the US from 1822 to 2005 in a blog to promote awareness and discussion.

Questions to Answer

Do different time periods between 1822 and 2005 reveal different trends in sugar consumption in the US diet? Can a regression graph be used to make predictions outside the range of the study data?

Design Issues

The data represent added caloric sugars such as cane sugar, high-fructose corn syrup and maple syrup, not naturally occurring sugars such as those in fruits and vegetables. Thus the data do not represent total sugar consumption. The data are not direct measures of consumption, but rather estimates derived from sales figures by adjusting for losses before consumption. The adjustment, applied across all years, is based on the USDA loss estimate from $1970-2005$, which may or may not underestimate sugar consumption in earlier time periods.

Descriptions of Variables

Table $20.26.1$: Description of Variables

Variable	Description
year	All years from 1822 to 2005
sugar_consum	Estimated consumption of added sugars in the US diet in pounds per year per person

Data Files

Sugar.xls

Links

By 2606, the US Diet will be 100 Percent Sugar, a blog by Stephan Guyenet

How to Spot Added Sugar on Food Labels

Dietary Sugars Intake and Cardiovascular Health: A Scientific Statement From the American Heart Association

Sugar: The Bitter Truth, a lecture by Robert H. Lustig

60 Minutes: Is Sugar Toxic?

References

- Johnson, R. K., Appel, L. J., Brands, M., Howard, B. V., Lefevre, M., Lustig, R. H., Sacks, F., Steffen, L. M., Wylie-Rosett, J. (2009). Dietary sugars intake and cardiovascular health: A scientific statement from the American Heart Association. Circulation, 120, 1011-1020.

This page titled 20.26: Sugar Consumption in the US Diet is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.26: Sugar Consumption in the US Diet** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.27: Nutrition Information Sources and Older Adults

Learning Objectives

- Better educated people and information sources

Research conducted by

Diane L. McKay, Robert F. Houser, Jeffrey B. Blumberg, and Jeanne P. Goldberg

Case study prepared by

Robert F. Houser, Alyssa Koomas, Georgette Baghdady, and Jennifer E. Konick

Overview

Various socioeconomic factors, such as occupation, income, race, and education level, are associated with health outcomes. Prominent among them, education level has proved to be a strong predictor of diet quality, health behavior patterns, and disease risk. Studies have found that better-educated people have healthier diets than those with less education, leading some researchers to hypothesize that better-educated people may obtain nutrition information from more reliable sources than less-educated people.

This study examined that hypothesis among a sample of **176** adults aged **50** years or older. The participants completed a survey which asked whether or not they primarily relied upon each of the following sources for information about nutrition: doctors, other medical professionals, newspapers, magazines, television, radio, friends, relatives, and neighbors. Analysis involved comparing the sources by education level. Older adults are highly vulnerable to diet-related disease. Knowing which sources they rely on can enable nutrition educators and professionals to target those sources with high-quality nutrition messages, tailored to the needs and education level of the older-adult audience.

Questions to Answer

What sources of nutrition information do older adults rely on? Do these sources differ according to the educational attainment and gender of the adults? Are these sources of nutrition information related to dietary practices, such as taking supplements?

Design Issues

Given that the sample was drawn only from the New England area and that **93%** were Caucasian, the results of this study should not be generalized to older adults in other regions or racial and ethnic groups. The Internet as a source of nutrition information was not included in the survey; it is likely a primary source among today's older adults.

Descriptions of Variables

Table 20.27.1: Description of Variables

Variable	Description
coll4yrplus	Highest level of education completed: 0 = "< 4 years of college" (i.e., secondary school, high school, vocational school, community or junior college) 1 = "≥ 4 years of college" (i.e., four-year college, graduate or professional school)
gender	1 = female, 2 = male
doctor	Is your doctor a primary source of information about nutrition? 1 = yes, 2 = no
magazine	Are magazines a primary source of information about nutrition? 1 = yes, 2 = no
tv	Is TV a primary source of information about nutrition? 1 = yes, 2 = no
friends	Are friends a primary source of information about nutrition? 1 = yes, 2 = no
supps	Are you taking any dietary supplements? 1 = yes, 2 = no

Data Files

Nutrition_information.xls

Links

Nutrition Information For You

Evaluating Nutrition Information (see pages 36-43)

Nutrition Accuracy in Popular Magazines

References

- McKay, D. L., Houser, R. F., Blumberg, J. B., Goldberg, J. P. (2006). Nutrition information sources vary with education level in a population of older adults. Journal of the American Dietetic Association, 106, 1108-1111.

This page titled 20.27: Nutrition Information Sources and Older Adults is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.27: Nutrition Information Sources and Older Adults** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.28: Mind Set - Exercise and the Placebo Effect

Learning Objectives

- The "placebo" effect

Research conducted by

Alia J. Crum and Ellen J. Langer

Case study prepared by

Robert F. Houser and Alyssa Koomas

Overview

The "placebo effect" is an effect that cannot be attributed to a drug or remedy, but rather to a change in a person's mind-set or perception. The placebo effect is widely accepted in clinical trials and its effects may shock you. For instance, one study found that subjects developed real rashes after being exposed to fake poison ivy (Blakeslee, 1998)! This study examined the placebo effect with relation to physical activity and health. Could becoming aware of how much you exercise result in weight loss even if you didn't make any changes to your diet or exercise routine?

The subjects were 84 female maids of ages 19 to 65 years at seven hotels. They were told the purpose of the study was to improve the health and happiness of hotel maids. According to the authors, "[e]ach of seven hotels was randomly assigned to one of two conditions: informed or control" (page 166). "Four hotels were assigned to the informed condition, and three were assigned to the control condition" (pages 166 – 167). Each subject filled out a questionnaire asking about her perceived amount of exercise during and outside of work. Physiological measurements were taken for weight, body mass index, body-fat percentage, waist-to-hip ratio, and blood pressure. The maids in the informed condition were then given an oral presentation and handouts explaining how their work as hotel maids is good exercise, so good in fact that it meets or exceeds the Surgeon General's recommendations for physical activity. The maids in the control condition were not given this information. After four weeks, the researchers re-administered the questionnaire and took follow-up physiological measurements.

Questions to Answer

Does the placebo effect play a role in the health benefits of exercise? If we alter a person's perception of the exercise she performs, does it result in weight loss?

Design Issues

Instead of assigning individual maids randomly to either the informed or control condition, all of the maids in the same hotel were assigned to the same condition. This was done in an effort to prevent information contamination. This type of study design is known as a "cluster randomized trial," and calls for advanced statistical practices that we will not worry about in this case study.

Simple random sampling with a sufficient number of subjects randomly assigned to intervention and control groups ideally leads to intervention and control groups that are similar with respect to many demographic characteristics. Simple random sampling of individuals and random assignment of individuals to conditions were not used in this study. The authors of this study pointed out that "[s]ubjects in the informed group were significantly younger than subjects in the control group." Consequently, they attempted to control for age differences in their statistical analysis.

The questionnaire asked about self-reported levels of exercise and dietary intake. Future research should use more rigorous methods to assess physical activity and diet.

Descriptions of Variables

Table 20.28.1: Description of Variables

VARIABLE	DESCRIPTION
cond	Condition: Either Informed or Control
age	Age in years
ex1	Perceived amount of exercise at Time 1 (On a scale from 0 to 10 with 0 = “none” and 10 = “a great deal”)
ex2	Perceived amount of exercise at Time 2 (On a scale from 0 to 10 with 0 = “none” and 10 = “a great deal”)
wt1	Weight in pounds at Time 1
wt2	Weight in pounds at Time 2
aex	Change score for exercise equal to the perceived amount of exercise at Time 2 minus the perceived amount of exercise at Time 1
awt	Weight change equal to the weight at Time 2 minus the weight at Time 1

Data Files

Mindset.xls

Links

Crum et al. article

New York Times article

References

- Crum, A. J., Langer, E. J. (2007). Mind-set matters: Exercise and the placebo effect. Psychological Science, 18, 165-171.

This page titled 20.28: Mind Set - Exercise and the Placebo Effect is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.28: Mind Set - Exercise and the Placebo Effect** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.29: Predicting Present and Future Affect

Learning Objectives

- To explore the phenomenon of future anhedonia

Research conducted by

Karim S. Kassam, Daniel T. Gilbert, Andrew Boston, and Timothy D. Wilson

Case study prepared by

Robert F. Houser and Georgette Baghdady

Overview

In Aesop's fable, "The Ant and the Grasshopper," an ant toils all summer to gather food for the winter while a grasshopper sunbathes and enjoys the present abundance of food without concern for the upcoming winter. Consequently when winter arrives, the grasshopper despairs that it has no food. The moral of the fable is that it is best to prepare for the days of necessity. Clearly the grasshopper failed to predict accurately how it would feel in the winter while it sunbathed with a full belly in the summer.

The authors of this study explored the intriguing phenomenon of future anhedonia and its relation to the concept of time discounting in order to understand people's predictions about how they might feel when a future event happens. Time discounting occurs when people put less value on future events than present events. Future anhedonia refers to people's mistaken belief that a future event would elicit a less intense affective reaction than if the same event happened in the present. In six experiments, the authors asked participants to predict how happy they would feel both in the present and in the future upon receiving either 20 dollars outright or 25 dollars in the form of a Starbucks coffeehouse gift card. The difference between the scores of present and future happiness is a measure of future anhedonia.

Questions to Answer

Do people expect their affective reactions to an event to be less intense in the future than in the present?

Design Issues

The monetary amount ($\$20, \$25$) may not have been enough to psychologically engage a large number of participants. The wide age range in **Experiment 1b** of 15 to 72 years is unusual for a psychological study. Also in **Experiment 1b**, several participants reported that they would pay $\$25$ for a $\$25$-gift card that Starbucks was considering selling at a discounted price, which might indicate that they did not fully understand the question.

Descriptions of Variables

Table $20.29.1$: Description of Variables

VARIABLE	DESCRIPTION
Gender	The sex of a participant
Happiness score	A participant’s estimate of his/her affective reaction to an event using a 9-point scale with endpoints 1 = “not at all happy” and 9 = “extremely happy”

diff_happy	A difference score equal to a participant's predicted present happiness score for a present event minus his/her predicted future happiness score for the same event in the future. A positive difference indicates future anhedonia.
diff_money	The difference in the maximum amount of money that a participant predicted as his/her willingness to pay in the present minus the predicted amount he/she would pay at a future time for a $25 Starbucks coffeehouse gift card. A positive difference indicates future anhedonia.
cond	Condition: Whether an event was expected or unexpected
today	A participant's predicted present happiness score for a present event
future	A participant's predicted future happiness score for a future event

Data Files

Predicting.xls

Links

Kassam et al. article

Aesop's Fable: The Ant and the Grasshopper

Prospection: Experiencing the Future

References

- Kassam, K. S., Gilbert, D. T., Boston, A., Wilson, T. D. (2008). Future anhedonia and time discounting. Journal of Experimental Social Psychology, 44, 1533-1537.
- Gilbert, D. T., Wilson, T. D. (2007). Prospection: Experiencing the future. Science, 317, 1351-1354

This page titled 20.29: Predicting Present and Future Affect is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.29: Predicting Present and Future Affect** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.30: Exercise and Memory

Learning Objectives

- To study the benefits of exercise on memory

Research conducted by

M. E. Hopkins, F. C. Davis, M. R. Van Tieghem, P. J. Whalen, and D. J. Bucci

Case study prepared by

Robert F. Houser and Georgette Baghdady

Overview

Physical exercise has many beneficial effects on physiological processes, including those that affect cognition and memory. Exercise increases brain-derived neurotrophic factor (**BDNF**), which is a protein found in the learning and memory centers of the brain where it supports nerve cell survival and the growth of new neurons and neuronal connections. A polymorphism of **BDNF** (a variant genotype) alters the release of **BDNF** during exercise. The researchers of this study sought to compare the effects of a single bout of exercise versus a **4**-week exercise regimen on cognition and memory and to determine if **BDNF** genotype influences the intensity of those effects of exercise.

Questions to Answer

How do regular exercise and/or an acute bout of exercise affect cognitive memory? Does type of **BDNF** genotype (Val/Val or Met carrier) mediate the effect of exercise on memory? How do we calculate a one-way ANOVA by hand and how do different post-hoc tests compare?

Design Issues

The group sample sizes are small, perhaps limiting the power to detect significant differences between the four exercise/control groups.

Descriptions of Variables

Table **20.30.1**: Description of Variables

Variable	Description
Group	0W-: sedentary group 0W+: sedentary group with one bout of exercise at least 2 hours before Visit 2 4W-: regularly exercising group 4W+: regularly exercising group with a bout of exercise at least 2 hours before Visit 2
Accuracy	The percentage of objects each group accurately identified as old or new when performing the novel object recognition task during each study visit
Difference score	Accuracy achieved by the subject in the novel object recognition task during Visit 2 minus accuracy during Visit 1, in percent
BDNF genotype	Whether a subject's BDNF genotype is Val/Val or Met carrier (Val/Met and Met/Met)

Links

How Exercise Affects the Brain: Age and Genetics Play a Role

BDNF

References

- Hopkins, M. E., Davis, F. C., Van Tieghem, M. R., Whalen, P. J., Bucci, D. J. (2012). Differential effects of acute and regular physical exercise on cognition and affect. Neuroscience, 215, 59-68.

This page titled 20.30: Exercise and Memory is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.30: Exercise and Memory** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.31: Parental Recognition of Child Obesity

Learning Objectives

- To study the parents' perception of their children's weight status

Research conducted by

Debra Etelson, Donald A. Brand, Patricia A. Patrick, and Anushree Shirali

Case study prepared by

Robert F. Houser and Georgette Baghdady

Overview

With increasing public awareness of child obesity as a major public health problem, studies are showing that it has not translated into an increased awareness of obesity in one's own child. Dietary patterns and weight status in childhood tend to carry into adolescence and adulthood, promoting the onset of chronic and other diseases. A key ingredient for combating childhood obesity is parental involvement and commitment. However, this is predicated on whether or not parents can recognize overweight and obesity in their children.

This study examined parents' perceptions of their children's weight status, their understanding of the health risks of obesity relative to other conditions they may perceive as health risks, and their knowledge of some healthy eating practices. Children's actual weight status was expressed as their body mass index (**BMI**) percentile, as determined by the CDC growth charts based on age and sex. According to the CDC growth charts for children, a child with a **BMI** percentile less than the 5th percentile is underweight; from the 5th to less than the 85th, a child is at a healthy weight; from the 85th to less than the 95th percentile, a child is overweight; and a **BMI** percentile equal to or greater than the 95th percentile, a child is considered to be obese.

A visual analog scale was used to measure parents' perceptions of their child's weight. The visual analog scale consisted simply of a 10-cm straight line anchored at the left end by the label "extremely underweight" and at the right end by the label "extremely overweight." A parent placed a mark along the line to indicate where they perceived their child's weight to be. The researchers interpreted the marks as percentiles in their analysis.

Questions to Answer

Do parents recognize when their children are overweight or obese? Do parents who make incorrect judgments about healthy food practices also make incorrect judgments about their child's weight status?

Design Issues

This study defines a parent's perception of their child's **BMI** percentile as "accurate" if their score on a visual analog scale fell within 30 points of the child's true **BMI** percentile. This wide range defining accuracy potentially allows for misclassification of a child's weight status among normal, overweight, and obese categories. For example, a parent who perceives their child's weight status as being at the 80th percentile, i.e., in the normal range, when in reality the child is obese with a **BMI** percentile of 98, the parent's assessment would be considered accurate by the operational definition used in this study. The authors explain that they chose this definition to give parents as much leeway as possible in assessing their child's weight on the visual analog scale.

Descriptions of Variables

Table 20.31.1: Description of Variables

Variable	Description
Sex	The sex of the participating parent's child
Overwt_Obese	Whether or not a child's body mass index (BMI) is equal to or greater than the 85th percentile for the child's age and sex, which means that the child is either overweight (85th to less than 95th percentile) or obese (95th percentile or above)
PA_overwt	Parental attitude expressing level of concern if their child were overweight, measured on a 4-point Likert Scale. In data analysis, the four categories were condensed into two categories: 0 = "not at all" or "a little" concerned 1 = "quite" or "extremely" concerned
PA-TV	Parental attitude expressing level of concern if their child watched >20 hours of TV per week, measured on a 4-point Likert Scale. In data analysis, the four categories were condensed into two categories: 0 = "not at all" or "a little" concerned 1 = "quite" or "extremely" concerned
Accurate	Whether or not the parent's perception of their child's weight status was accurate. Parent's perception was considered accurate if the BMI percentile it corresponded to fell within 30 points of the child's actual BMI percentile
Juice_boxes	The amount of juice that a parent thinks is healthy for their child to drink each day (a juice box contains eight ounces). We condensed the original four response categories into two categories: 0 = "1 or 2 juice boxes per day" 1 = "3 to 8 juice boxes per day"
Fast_food_meals	How often a parent feels it is okay to eat at fast-food restaurants. We condensed the original four response categories into two categories: 0 = "once a month"

Links

Etelson et al. article

BMI percentiles for children

References

- Debra Etelson, D., Brand, D. A., Patrick, P. A., Shirali, A. (2003). Childhood obesity: Do parents recognize this health risk? Obesity Research, 11, 1362-1368

This page titled 20.31: Parental Recognition of Child Obesity is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.31: Parental Recognition of Child Obesity** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

20.32: Educational Attainment and Racial, Ethnic, and Gender Disparity

Learning Objectives

- To study large disparities in educational attainment among various racial and ethnic groups

Research conducted by

United States Census Bureau

Case study prepared by

Robert F. Houser, Georgette Baghdady, and Jennifer E. Konick

Overview

The U.S. Census Bureau defines educational attainment as the highest level of education that a person has completed. Large disparities in educational attainment continue to exist among racial and ethnic groups. The gender gap in educational attainment, however, has been undergoing a dramatic social shift in recent decades. In Table **20.32.1** below, the U.S. Census Bureau tabulated these trends among Whites, Blacks, Asians and Pacific Islanders, and Hispanics between **1970** and **2010**. This case study focuses only on college graduates. The data for "College graduate or more" represent the percentage of adults aged **25** years and older that obtained a degree from regular four-year colleges and universities and graduate or professional schools in each racial and ethnic group.

The U.S. Census Bureau defines the racial and ethnic categories in the following manner:

- “White” refers to persons having origins in any of the original peoples of Europe, the Middle East, or North Africa.
- “Black” refers to persons having origins in any of the Black racial groups of Africa.
- “Asian” refers to persons having origins in any of the original peoples of the Far East, Southeast Asia, or the Indian subcontinent.
- “Pacific Islander” refers to persons having origins in any of the original peoples of the Pacific Islands, such as Hawaii, Guam, Samoa, and Tonga.
- “Hispanic” refers to an ethnic group comprised of persons of any race who are of Cuban, Mexican, Puerto Rican, South or Central American, or other Spanish culture or origin.

Table 230. Educational Attainment by Race, Hispanic Origin, and Sex: 1970 to 2010

[In percent. See Table 229 for headnote and totals for both sexes]

Year	All races [1]		White [2]		Black [2]		Asian and Pacific Islander [2]		Hispanic [3]	
	Male	Female	Male	Female	Male	Female	Male	Female	Male	Female
HIGH SCHOOL GRADUATE OR MORE [4]										
1970	51.9	52.8	54.0	55.0	30.1	32.5	61.3	63.1	37.9	34.2
1980	67.3	65.8	69.6	68.1	50.8	51.5	78.8	71.4	45.4	42.7
1990	77.7	77.5	79.1	79.0	65.8	66.5	84.0	77.2	50.3	51.3
1995	81.7	81.6	83.0	83.0	73.4	74.1	(NA)	(NA)	52.9	53.8
2000	84.2	84.0	84.8	85.0	78.7	78.3	88.2	83.4	56.6	57.5
2005	84.9	85.5	85.2	86.2	81.0	81.2	[5] 90.4	[5] 85.2	57.9	59.1
2007	85.0	86.4	85.3	87.1	81.9	82.6	89.8	85.9	58.2	62.5
2008	85.9	87.2	86.3	87.8	81.8	84.0	90.8	86.9	60.9	63.7
2009	86.2	87.1	86.5	87.7	84.0	84.1	90.4	86.2	60.6	63.3
2010	86.6	87.6	86.9	88.2	83.6	84.6	91.2	87.0	61.4	64.4
COLLEGE GRADUATE OR MORE [4]										
1970	13.5	8.1	14.4	8.4	4.2	4.6	23.5	17.3	7.8	4.3
1980	20.1	12.8	21.3	13.3	8.4	8.3	39.8	27.0	9.4	6.0
1990	24.4	18.4	25.3	19.0	11.9	10.8	44.9	35.4	9.8	8.7
1995	26.0	20.2	27.2	21.0	13.6	12.9	(NA)	(NA)	10.1	8.4
2000	27.8	23.6	28.5	23.9	16.3	16.7	47.6	40.7	10.7	10.6
2005	28.9	26.5	29.4	26.8	16.0	18.8	[5] 54.0	[5] 46.8	11.8	12.1
2007	29.5	28.0	29.9	28.3	18.0	19.0	55.2	49.3	11.8	13.7
2008	30.1	28.8	30.5	29.1	18.7	20.4	55.8	49.8	12.6	14.1
2009	30.1	29.1	30.6	29.3	17.8	20.6	55.7	49.3	12.5	14.0
2010	30.3	29.6	30.8	29.9	17.7	21.4	55.6	49.5	12.9	14.9

NA Not available. [1] Includes other races not shown separately. [2] Beginning 2005, for persons who selected this race group only. See footnote 2, Table 229. [3] Persons of Hispanic origin may be any race. [4] Through 1990, completed 4 years of high school or more and 4 years of college or more. [5] Starting in 2005, data are for Asians only, excludes Pacific Islanders.

Source: U.S. Census Bureau, U.S. Census of Population, 1970 and 1980, Vol. 1; Current Population Reports P20-550, and earlier reports; and "Educational Attainment," <http://www.census.gov/population/www/socdemo/educ-attn.html>.

Figure **20.32.1**: Educational Attainment by Race, Hispanic Origin and Sex

Educational attainment is strongly associated with future employment, income, and health status.

Questions to Answer

How has the percentage of college graduates changed over time between 1970 and 2010 among the racial and ethnic groups and between the genders within each group? How might we illustrate these changes graphically?

Design Issues

Beginning with the 2000 U.S. Census, respondents were given the option of selecting more than one race category to indicate their racial identities. Therefore, data on race from 2000 and beyond are not directly comparable with earlier censuses. The data in Table 20.32.1 represent persons who selected only one race category and exclude persons who selected more than one race.

In the 2005 U.S. Census and beyond, the "Asian and Pacific Islander" category was split into two separate categories, "Asian" and "Native Hawaiian or Other Pacific Islander." There were several reasons for the split. The combined category was not a homogeneous group because it put together peoples with few social or cultural similarities and who are dissimilar on important demographic characteristics. For example, in 1990, about 11 percent of Pacific Islanders aged 25 years and older obtained a bachelor's degree compared with about 40 percent of Asians. Since Pacific Islanders are numerically a smaller group than Asians (in 2010, there were about a half million Pacific Islanders versus about 14.6 million Asians), not including them in the data of Table 20.32.1 starting in 2005 biases the percentage of college graduates upwards somewhat, but not strongly.

Descriptions of Variables

Table 20.32.1: Description of Variables

Variable	Description
College graduate or more	Obtained a degree from regular four-year colleges and universities and graduate or professional schools
Year	Decade years from 1970 to 2010
White_M White_F	Percentage of college graduates in U.S. subpopulation of White males aged 25 years and over; likewise for White females
Black_M Black_F	Percentage of college graduates in U.S. subpopulation of Black males aged 25 years and over; likewise for Black females
AsnPac_M AsnPac_F	Percentage of college graduates in U.S. subpopulation of Asian and Pacific Islander males aged 25 years and over; likewise for Asian and Pacific Islander females
Hispan_M Hispan_F	Percentage of college graduates in U.S. subpopulation of Hispanic males aged 25 years and over; likewise for Hispanic females

Data Files

Educational_attainment.xls

Links

Overview of Race and Hispanic Origin: 2010

Latinos and Education: Explaining the Attainment Gap

Why Do Women Outnumber Men in College?

References

- U.S. Census Bureau, Statistical Abstract of the United States: 2012. Section 4. Education, 143-151
- Telfair, J., Shelton, T. L. (2012). Educational attainment as a social determinant of health. North Carolina Medical Journal, 73(5), 358-365

This page titled 20.32: Educational Attainment and Racial, Ethnic, and Gender Disparity is shared under a Public Domain license and was authored, remixed, and/or curated by David Lane via source content that was edited to the style and standards of the LibreTexts platform.

- **20.32: Educational Attainment and Racial, Ethnic, and Gender Disparity** by David Lane is licensed Public Domain. Original source: https://onlinestatbook.com.

00: Front Matter

This page was auto-generated because a user created a sub-page to this page.

Lake Tahoe Community College
Support Course for Elementary Statistics

Larry Green

This text is disseminated via the Open Education Resource (OER) LibreTexts Project (https://LibreTexts.org) and like the thousands of other texts available within this powerful platform, it is freely available for reading, printing, and "consuming."

The LibreTexts mission is to bring together students, faculty, and scholars in a collaborative effort to provide an accessible, and comprehensive platform that empowers our community to develop, curate, adapt, and adopt openly licensed resources and technologies; through these efforts we can reduce the financial burden born from traditional educational resource costs, ensuring education is more accessible for students and communities worldwide.

Most, but not all, pages in the library have licenses that may allow individuals to make changes, save, and print this book. Carefully consult the applicable license(s) before pursuing such effects. Instructors can adopt existing LibreTexts texts or Remix them to quickly build course-specific resources to meet the needs of their students. Unlike traditional textbooks, LibreTexts' web based origins allow powerful integration of advanced features and new technologies to support learning.



LibreTexts is the adaptable, user-friendly non-profit open education resource platform that educators trust for creating, customizing, and sharing accessible, interactive textbooks, adaptive homework, and ancillary materials. We collaborate with individuals and organizations to champion open education initiatives, support institutional publishing programs, drive curriculum development projects, and more.

The LibreTexts libraries are Powered by NICE CXone Expert and was supported by the Department of Education Open Textbook Pilot Project, the California Education Learning Lab, the UC Davis Office of the Provost, the UC Davis Library, the California State University Affordable Learning Solutions Program, and Merlot. This material is based upon work supported by the National Science Foundation under Grant No. 1246120, 1525057, and 1413739.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation nor the US Department of Education.

Have questions or comments? For information about adoptions or adaptions contact info@LibreTexts.org or visit our main website at https://LibreTexts.org.

This text was compiled on 04/19/2026

TABLE OF CONTENTS

00: Front Matter

- TitlePage
- InfoPage
- Table of Contents
- Licensing

21.1: Decimals Fractions and Percents

- 21.1.1: Comparing Fractions, Decimals, and Percents
- 21.1.2: Converting Between Fractions, Decimals and Percents
- 21.1.3: Decimals- Rounding and Scientific Notation
- 21.1.4: Using Fractions, Decimals and Percents to Describe Charts

21.2: The Number Line

- 21.2.1: Distance between Two Points on a Number Line
- 21.2.2: Plotting Points and Intervals on the Number Line
- 21.2.3: Represent an Inequality as an Interval on a Number Line
- 21.2.4: The Midpoint

21.3: Operations on Numbers

- 21.3.1: Area of a Rectangle
- 21.3.2: Factorials and Combination Notation
- 21.3.3: Order of Operations
- 21.3.4: Order of Operations in Expressions and Formulas
- 21.3.5: Perform Signed Number Arithmetic
- 21.3.6: Powers and Roots
- 21.3.7: Using Summation Notation

21.4: Sets

- 21.4.1: Set Notation
- 21.4.2: The Complement of a Set
- 21.4.3: The Union and Intersection of Two Sets
- 21.4.4: Venn Diagrams

21.5: Expressions, Equations and Inequalities

- 21.5.1: Evaluate Algebraic Expressions
- 21.5.2: Inequalities and Midpoints
- 21.5.3: Solve Equations with Roots
- 21.5.4: Solving Linear Equations in One Variable

21.6: Graphing Points and Lines in Two Dimensions

- 21.6.1: Finding Residuals
- 21.6.2: Find the Equation of a Line given its Graph
- 21.6.3: Find y given x and the Equation of a Line

- 21.6.4: Graph a Line given its Equation
- 21.6.5: Interpreting the Slope of a Line
- 21.6.6: Interpreting the y-intercept of a Line
- 21.6.7: Plot an Ordered Pair

Index

Glossary

Detailed Licensing

Licensing

*A detailed breakdown of this resource's licensing can be found in **Back Matter/Detailed Licensing**.*

SECTION OVERVIEW

21.1: Decimals Fractions and Percents

21.1.1: Comparing Fractions, Decimals, and Percents

21.1.2: Converting Between Fractions, Decimals and Percents

21.1.3: Decimals- Rounding and Scientific Notation

21.1.4: Using Fractions, Decimals and Percents to Describe Charts

---

This page titled 21.1: Decimals Fractions and Percents is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

21.1.1: Comparing Fractions, Decimals, and Percents

Learning Outcomes

1. Compare two fractions
2. Compare two numbers given in different forms

In this section, we will go over techniques to compare two numbers. These numbers could be presented as fractions, decimals or percents and may not be in the same form. For example, when we look at a histogram, we can compute the fraction of the group that occurs the most frequently. We might be interested in whether that fraction is greater than 25% of the population. By the end of this section we will know how to make this comparison.

Comparing Two Fractions

Whether you like fractions or not, they come up frequently in statistics. For example, a probability is defined as the number of ways a sought after event can occur over the total number of possible outcomes. It is commonly asked to compare two such probabilities to see if they are equal, and if not, which is larger. There are two main approaches to comparing fractions.

Approach 1: Change the fractions to equivalent fractions with a common denominator and then compare the numerators

The procedure of approach 1 is to first find the common denominator and then multiply the numerator and the denominator by the same whole number to make the denominators common.

Example 21.1.1.1

Compare: $\frac{2}{3}$ and $\frac{5}{7}$

Solution

A common denominator is the product of the two: $3 \times 7 = 21$. We convert:

$$\frac{2}{3}\frac{7}{7} = \frac{14}{21}$$

and

$$\frac{5}{7}\frac{3}{3} = \frac{15}{21}$$

Next we compare the numerators and see that $14 < 15$, hence

$\frac{2}{3} < \frac{5}{7}$

Example 21.1.1.2

In statistics, we say that two events are independent if the probability of the second occurring is equal to the probability of the second occurring given that the first occurs. The probability of rolling two dice and having the sum equal to 7 is $\frac{6}{36}$. If you know that the first die lands on a 4, then the probability that the sum of the two dice is a 7 is $\frac{1}{6}$. Are these events independent?

Solution

We need to compare $\frac{6}{36}$ and $\frac{1}{6}$. The common denominator is 36. We convert the second fraction to

$$\frac{1}{6}\frac{6}{6} = \frac{6}{36}$$

Now we can see that the two fractions are equal, so the events are independent.

Approach 2: Use a calculator or computer to convert the fractions to decimals and then compare the decimals

If it is easy to build up the fractions so that we have a common denominator, then Approach 1 works well, but often the fractions are not simple, so it is easier to make use of the calculator or computer.

Example 21.1.1.3

In computing probabilities for a uniform distribution, fractions come up. Given that the number of ounces in a medium sized drink is uniformly distributed between 15 and 26 ounces, the probability that a randomly selected medium sized drink is less than 22 ounces is $\frac{7}{11}$. Given that the weight of in a medium sized American is uniformly distributed between 155 and 212 pounds, the probability that a randomly selected medium sized American is less than 195 pounds is $\frac{40}{57}$. Is it more likely to select a medium sized drink that is less than 22 ounces or to select a medium sized American who is less than 195 pounds?

Solution

We could get a common denominator and build the fractions, but it is much easier to just turn both fractions into decimal numbers and then compare. We have:

$$\frac{7}{11} \approx 0.6364$$

and

$$\frac{40}{57} \approx 0.7018$$

Notice that

$$0.6364 < 0.7018$$

Hence, we can conclude that it is less likely to pick the medium sized 22 ounce or less drink than to pick the 195 pound or lighter medium sized person.

Exercise

If you guess on 10 true or false questions, the probability of getting at least 9 correct is $\frac{11}{1024}$. If you guess on six multiple choice questions with three choices each, then the probability of getting at least five of the six correct is $\frac{7}{729}$. Which of these is more likely?

Comparing Fractions, Decimals and Percents

When you want to compare a fraction to a decimal or a percent, it is usually easiest to convert to a decimal number first, and then compare the decimal numbers.

Example 21.1.1.4

Compare 0.52 and $\frac{7}{13}$.

Solution

We first convert $\frac{7}{13}$ to a decimal by dividing to get 0.5385. Now notice that

$$0.52 < 0.5385$$

Thus

$$0.52 < \frac{7}{13}$$

Example 21.1.1.5

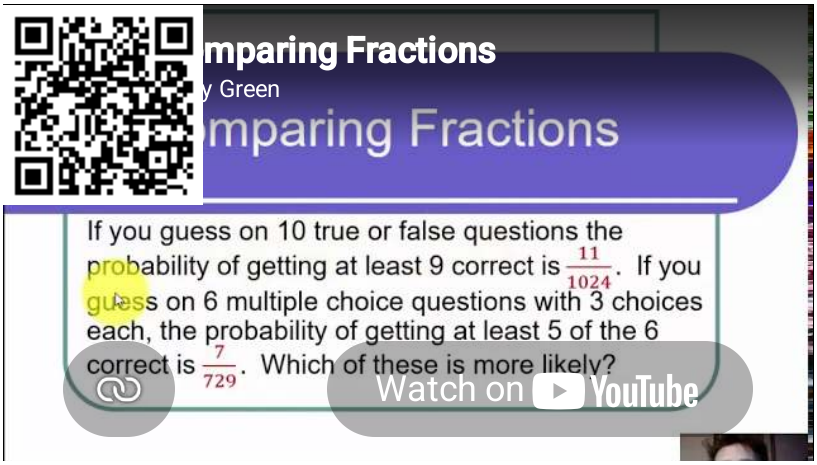
When we preform a hypothesis test in statistics, We have to compare a number called the p-value to another number called the level of significance. Suppose that the p-value is calculated as 0.0641 and the level of significance is 5%. Compare these two numbers.

Solution

We first convert the level of significance, 5%, to a decimal number. Recall that to convert a percent to a decimal, we move the decimal over two places to the right. This gives us 0.05. Now we can compare the two decimals:

$$0.0641 > 0.05$$

Therefore, the p-value is greater than the level of significance.



This is an application of comparing fractions to probability.

- Example: Comparing Fractions with Different Denominators using Inequality Symbols
- Ex: Compare Fractions and Decimals using Inequality Symbols
- https://youtu.be/lSzNkQjcfEU

This page titled 21.1.1: Comparing Fractions, Decimals, and Percents is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Comparing Fractions, Decimals, and Percents** by Larry Green is licensed CC BY 4.0.

21.1.2: Converting Between Fractions, Decimals and Percents

Learning Outcomes

1. Given a decimal, convert it to a percent
2. Given a percent, convert it to a decimal
3. Convert a fraction to a decimal and percent

In this section, we will convert from decimals to percents and back. We will also start with a fraction and convert it to a decimal and a percent. In statistics we are often given a number as a percent and have to do calculations on it. To do so, we must first convert it to a percent. Also, the computer or calculator shows numbers as decimals, but for presentations, percents are friendlier. It is also much easier to compare decimals than fractions, thus converting to a decimal is helpful.

For example, we often want to see if a probability is greater than 5%. A computer will display the probability as a decimal such as 0.04836. To make the comparison we will first change it to a percent and then compare it to 5%.

Transforming a Decimal to a Percent

We have all heard of percents before. "You only have a 20% chance of winning the game", "Just 38% of all Americans approve of Congress", and "I am 95% confident that my answer is correct" are just a few of the countless examples of percents as they come up in statistics.

Defintion: Percent

Percent means Parts Per Hundred

Thus if we are given a decimal and want to convert it to a percent, we multiply the decimal by 100. In practice, this means we move the decimal point two places to the right.

Example 21.1.2.1

Convert the number 0.1738 to a percent.

Solution

We move the decimal over two to the right as shown below.



We get: 17.38% for the answer.

Example 21.1.2.2

Convert 0.7 to a percent.

Solution

We want to move the decimal two places to the right, but there is only one digit to the right of the decimal place. The good news is that we can always add a 0 to the right of the last digit. We write:

$$0.7 = 0.70$$

Now move the decimal place two digits to the right to get 70%.

Example 21.1.2.3

In regression analysis, an important number that is calculated is called R-Squared. It helps us determine how helpful one variable is in predicting another variable. The computer and calculator always display it as a decimal, but it is more meaningful as a percent. Suppose that the R-Squared value that relates the amount of studying students do to prepare for a final exam and the score on the exam is: $r^2 = 0.8971$. Convert this to a percent rounded to the nearest whole number percent.

Solution

We move the decimal 0.8971 two places to the right to get 89.71%

Now round to the nearest whole number percent. Note that the digit to the left of the whole number is $7 \geq 5$. Thus we add 1 to the whole number, 89. This gives us 90%.

Exercise

A standard goal in statistics is to come up with a range of values that a population proportion is likely to lie. This range is called a confidence interval. Suppose that we want to interpret a confidence interval for the percent of patients who experience side effects from an experimental cancer treatment. The computer calculates it as the decimal range: [0.023,0.029]. What is the likely range for the percent of patients who experience side effects from the experimental cancer treatment?

Transforming a Percent to a Decimal

To convert a decimal to a percent, we multiply the decimal by 100 which is equivalent to moving the decimal two places to the right. Not surprisingly, to convert a percent to a decimal, we do exactly the opposite. We divide the number by 100 which is equivalent to moving the decimal two places to the left.

Example 21.1.2.4

Convert the percent 89.4% to a decimal.

Solution

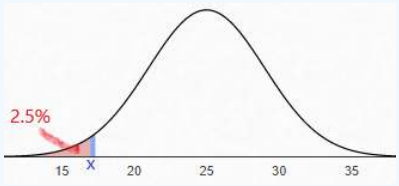
We move the decimal over two to the left as shown below.



We get: 0.894 for the answer.

Example 21.1.2.5

Suppose that you want to find the value of x such that 2.5% of the entire area under the Normal curve lies to the left of x. The first step will be to convert the 2.5% to a decimal. What decimal is equivalent to 2.5%?



Solution

We want to move the decimal 2.5 two places to the left, but since there is only one digit to the left of the decimal, we add a zero first: 02.5. Now move the decimal two places to the left to get 0.025.

Converting a Fraction to a Decimal and a Percent

Often in probability it is natural to represent probabilities as fractions, but it is easier to make comparisons as decimals. Thus, we need to be able to convert fractions to decimals. To do so we just divide.

Example 21.1.2.6

Convert the fraction $\frac{4}{7}$ to a decimal, rounding to the nearest hundredth.

Solution

We use long division:

$$\begin{array}{r} .571 \\ 7\overline{)4.000} \\ \underline{35} \\ 50 \\ \underline{49} \\ 10 \end{array} \qquad (21.1.2.1)$$

Next round to the nearest hundredth to get 0.57.

Although everyone's favorite thing to do is to perform long division by hand, in most statistics classes you will have a calculator or computer to use. Thus you just have to remember to perform the division with the calculator or computer and then round.

Example 21.1.2.7

In statistics we need to find basic probabilities and create a table for them. Suppose that you roll two six-sided dice, what percent of the time will the sum equal to a 4? Round to the nearest whole number percent.

Solution

First, notice that there are 36 total possibilities for rolling the dice, since there are 6 faces on the first die and for each value of the first die roll, there are 6 possibilities for the second die roll. Multiplying: 6 x 6 = 36. This will be the denominator. To find the numerator, we list all the possible outcome where the sum is 4:

(1,3), (2,2), and (3,1)

There are three possible outcomes with the sum equaling a 4. Thus:

$$P(\text{sum} = 4) = 3/36$$

Now we divide:

$$\frac{3}{36} = 0.08333...$$

Next to convert this decimal to a percent, we move the decimal two places to the right to get: 8.333...%

We are asked to round to the nearest whole number percent. The digit to the right or the whole number (8) is a 3. Since 3 < 5, we can just erase everything to the left of the 8 and leave the 8 unchanged to get 8%. Thus there is an 8% chance of getting a sum of 4 if you roll two six sided dice.

- Convert Percentages to Decimals
- Relating Fractions, Decimals, and Percents
- Statistics Application of Converting Decimals to Percents

This page titled 21.1.2: Converting Between Fractions, Decimals and Percents is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Converting Between Fractions, Decimals and Percents** by Larry Green is licensed CC BY 4.0.

21.1.3: Decimals- Rounding and Scientific Notation

Learning Outcomes

1. Understand what it means to have a number rounded to a certain number of decimal places.
2. Round a number to a fixed number of digits.
3. Convert from scientific notation to decimal notation and back.

In this section, we will go over how to round decimals to the nearest whole number, nearest tenth, nearest hundredth, etc. In most statistics applications that you will encounter, the numbers will not come out evenly, and you will need to round the decimal. We will also look at how to read scientific notation. A very common error that statistics students make is not noticing that the calculator is giving an answer in scientific notation.

For example, suppose that you used a calculator to find the probability that a randomly selected day in July will have a high temperature of over 90 degrees. Your calculator gives the answer: 0.4987230156. This is far too many digits for practical use, so it makes sense to round to just a few digits. By the end of this section you will be able to perform the rounding that is necessary to make unmanageable numbers manageable.

Brief Review of Decimal Language

Consider the decimal number: 62.5739. There is a defined way to refer to each of the digits.

- The digit 6 is in the "Tens Place"
- The digit 2 is in the "Ones Place"
- The digit 5 is in the "Tenths Place"
- The digit 7 is in the "Hundredths Place"
- The digit 3 is in the "Thousandths Place"
- The digit 9 is in the "Ten-thousandths Place"
- We also say that 62 is the "Whole Number" part.



Keeping this example in mind will help you when you are asked to round to a specific place value.

Example 21.1.3.1

It is reported that the mean number of classes that college students take each semester is 3.2541. Then the digit in the *hundredths place* is 5.

Rules of Rounding

Now that we have reviewed place values of numbers, we are ready to go over the process of rounding to a specified place value. When asked to round to a specified place value, the answer will erase all the digits after the specified digit. The process to deal with the other digits is best shown by examples.

Example 21.1.3.2: Case 1 - The Test Digit is Less Than 5

Round 3.741 to the nearest tenth.

Solution



Since the test digit (4) is less than 5, we just erase everything to the right of the tenths digit, 7. The answer is: 3.7.

Example 21.1.3.3: Case 2 - The Test Digit is 5 or Greater

Round 8.53792 to the nearest hundredth.

Solution



Since the test digit (6) is 5 or greater, we add one to the hundredths digit and erase everything to the right of the hundredths digit, 3. Thus the 3 becomes a 4. The answer is: 8.54.

Example 21.1.3.4: Case 3 - The Test Digit is 5 or Greater and the rounding position digit is a 9

Round 0.014952 to four decimal places.

Solution



The test digit is 5, so we must round up. The rounding position is a 9 and adding 1 gives 10, which is not a single digit number. Instead look at the two digits to the left of the test digit: 49. If we add 1 to 49, we get 50. Thus the answer is 0.0150.

Applications

Rounding is used in most areas of statistics, since the calculator or computer will produce numerical answers with far more digits than are useful. If you are not told how many decimal places to round to, then you often want to think about the smallest number of decimals to keep so that no important information is lost. For example suppose you conducted a sample to find the proportion of college students who receive financial aid and the calculator presented 0.568429314. You could turn this into a percent at 56.8429314%. There are no applications where keeping this many decimal places is useful. If, for example, you wanted to present this finding to the student government, you might want to round to the nearest whole number. In this case the ones digit is 6 and the test digit is 8. Since $8 \geq 5$, you add 1 to the ones digit. You can tell the student government that 57% of all college students receive financial aid.

Example 21.1.3.5

Suppose that you found out that the probability that a randomly selected person with who has misused prescription opioids will transition to heroin is 0.04998713. Round this number to four decimal places.

Solution

The first four decimal places are 0.0499 and the test digit is 8. Since $8 \geq 5$, we would like to add 1 to the fourth digit. Since this is a 9, we go to the next digit to the left. This is also a 9, so we go to the next one which is a 4. We can think of adding 0499 + 1 = 0500. Thus the answer is 0.0500. Note that we keep the last two 0's after the 5 to emphasize that this is accurate to the fourth decimal place.

Rounding and Arithmetic

Many times, we have to do arithmetic on numbers with several decimal places and want the answer rounded to a smaller number of decimal places. One question you might ask is should you round before you perform the arithmetic or after. For the most accurate result, you should always round after you preform the arithmetic if possible.

> *When asked to do arithmetic and present you answer rounded to a fixed number of decimal places, only round after performing the arithmetic.*

Example 21.1.3.6

Suppose you pick three cards from a 52 card deck with replacement and want to find the probability of the event, A, that none of the three cards will be a 2 through 7 of hearts. This probability is:

$$P(A) = (0.8846)^3$$

Round the answer to 2 decimal places.

Solution

Note that we have to first perform the arithmetic. With a computer or calculator we get:

$$0.8846^3 = 0.69221467973$$

Now we round to two decimal places. Notice that the hundredths digit is a 9 and the test digit is a 2. Thus the 9 remains unchanged and everything to the right of the 9 goes away. the result is

$$P(A) \approx 0.69$$

If we mistakenly rounded 0.8846 to two decimal places (0.88) and then cubed the answer we would have gotten 0.68 which is not the correct answer.

Scientific Notation

When a calculator presents a number in scientific notation, we must pay attention to what this represents. The standard way of writing a number in scientific notation is writing the number as a product of a number greater than or equal 1 but less than 10 followed by a power of 10. For example:

$$602,000,000,000,000,000,000,000 = 6.02 \times 10^{23}$$

The main purpose of scientific notation is to allow us to write very large numbers or numbers very close to 0 without having to use so many digits. Most calculators and computers use a different notation for scientific notation, most likely because the superscript is difficult to render on a screen. For example, with a calculator:

$$0.00000032 = 3.2E-7$$

Notice that to arrive at 3.2, the decimal needed to be moved 7 places to the right.

Example 21.1.3.7

A calculator displays:

$$2.0541E6$$

Write this number in decimal form.

Solution

Notice that the number following E is 6. This means move the decimal over 6 places to the right. The first 4 moves is natural, but for the last 2 moves, there are no numbers to move the decimal place past. We can always add extra zeros after the last number to the right of the decimal place:

$$2.0541E6 = 2.054100E6$$

Now we can move the decimal place to the right 6 places to get

$$2.0541E6 = 2.054100E6 = 2,054,100$$

Example 21.1.3.8

If you use a calculator or computer to find the probability of flipping a coin 27 times and getting all heads, then it will display:

$$7.45E-9$$

Write this number in decimal form.

Solution

Many students will forget to look for the "E" and just write that the probability is 7.45, but probabilities can never be bigger than 1. You can not have a 745% chance of it occurring. Notice that the number following E is −9. Since the power is negative, this means move the decimal to the left, and in particular 9 places to the left. There is only one digit to the left of the decimal place, so we need to insert 8 zeros:

$$7.45E-9 = 000000007.45E-9$$

Now we can move the decimal place to the right 9 places to the left to get

$$7.45E-9 = 000000007.45E-9 = 0.00000000745$$

- Application of Rounding Decimal Numbers
- Here is a video that explains rounding.

This page titled 21.1.3: Decimals- Rounding and Scientific Notation is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Decimals: Rounding and Scientific Notation** by Larry Green is licensed CC BY 4.0.

21.1.4: Using Fractions, Decimals and Percents to Describe Charts

Learning Outcomes

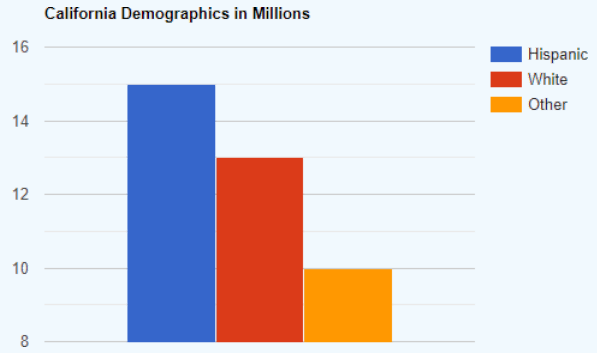
1. Interpret bar charts using fractions, decimals and percents
2. Interpret pie charts using fractions, decimals and percents

Charts, such as bar charts and pie charts are visual ways of presenting data. You can think of each slice of the pie or each bar as a part of the whole. The numerical versions of this are a list of fractions, decimals and percents. By the end of this section we will be able to look at one of these charts and produce the corresponding fractions, decimals, and percents.

Reading a Bar Chart

Bar charts occur frequently and it is definitely required to understand how to read them and interpret them in statistics. Often we want to convert the information of a bar chart to information shown numerically. We need fractions and/or percents to do this.

Example 21.1.4.1



The above bar chart shows the demographics of California in 2019 where the numbers represent millions of people. Here are some questions that might come up in a statistics class.

A. What fraction of Californians was Hispanic in 2019?
B. What proportion of all Californians was White in 2019? Write your answer as a decimal number rounded to four decimal places.
C. What percent of Californians who were neither Hispanic nor White in 2019? Round your answer to the nearest percent.

Solution

A. To find the fraction of California that was Hispanic in 2019, the numerator will be the total number of Hispanics and the denominator will be the total number of people in California in 2019. The height of the bar that represents Hispanics is 15. Therefore the numerator is 15. To find the total number of people in California, we add up the heights of the three bars:

$$15 + 13 + 10 = 38$$

Now we can just write down the fraction:

$$\frac{15}{38}$$

To find the proportion of Californians who were White in 2019, we start in the same way. The numerator will be the number of Whites: 13. The denominator will be the total number of Californians which we already computed as 38. Therefore the fraction of Californians who were White is:

$$\frac{13}{38}$$

To convert this to a decimal, we use a calculator to get:

$$\frac{13}{38} \approx 0.342105$$

Next round to four decimal places. Since the digit to the right of the fourth decimal place is $0 < 5$, we round down to:

$$0.3421$$

B. To find the percent of Californians who were neither Hispanic nor White in 2019, we first find the fraction who were neither. The numerator will be the number of "Other" which is: 10. The denominator will be the total which is 38. Thus the fraction is:

$$\frac{10}{38}$$

Next, use a calculator to divide these numbers to get:

$$\frac{10}{38} \approx 0.263158$$

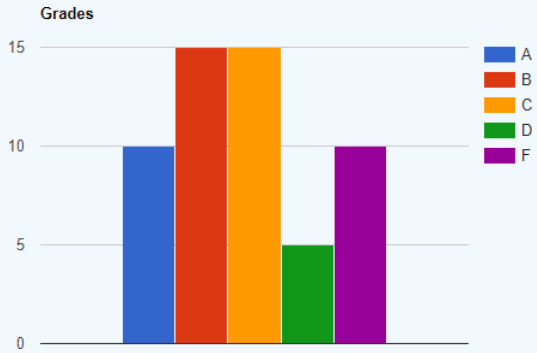
To convert this to a percent we multiply by 100% by moving the decimal two places to the right:

$$0.263158 \times 100\% = 26.3158\%$$

Finally we round to the nearest whole number. Noting that $3 < 5$, we round down to get: 26%

Exercise

The bar chart below shows the grade distribution for a math class.



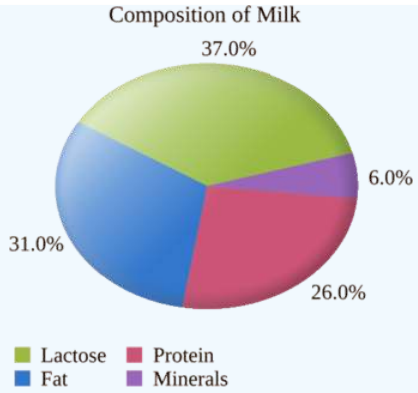
A. Find the fraction of students who received a "C" grade.
B. Find the proportion of grades below a "C". Write your answer as a decimal number rounded to the nearest hundredth.
C. What percent of the students received an "A" grade? Round your answer to the nearest whole number percent.

Reading a Pie Chart

Another important chart that is used to display the components of a whole is a pie chart. With a pie chart, it is very easy to determine the percent of each item.

Example 21.1.4.2

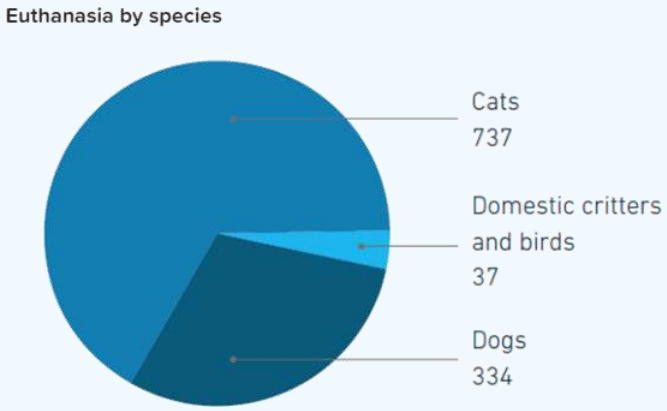
The pie chart below shows the makeup of milk. Write the proportion of fat contained in milk as a decimal.



Solution

We see that 31% of milk is fat. To convert a percent to a decimal, we just move the decimal over two places to the left. Thus, 31% becomes 0.31.

Example 21.1.4.3



The pie chart above shows the number of pets of each type that had to be euthanized by the humane society due to incurable illnesses.

A. What fraction of the euthanized pets were dogs?
B. What percent of the euthanized pets were cats? Round to the nearest whole number percent.

Solution

A. We take the number of dogs over the total. There were 334 euthanized dogs. To find the total we add:

$$737 + 37 + 334 = 1108$$

Therefore, the fraction of euthanized dogs is

$$\frac{334}{1108}$$

B. To find the percent of euthanized cats, we first find the fraction. There were 737 cats over a total of 1108 pets. The fraction is

$$\frac{737}{1108}$$

Next use a calculator to get the decimal number: 0.66516. Now multiply by 100% by moving the decimal place two digits to the right to get: 66.516%. Finally, we need to round to the nearest whole number percent. Since $5 \geq 5$, we round up.

> Thus the percent of euthanized cats is 67%.

- Finding Fractions, Decimals and Percents from a Bar Chart
- Ex: Find the a Percent of a Total Using an Amount in Pie Chart

---

This page titled 21.1.4: Using Fractions, Decimals and Percents to Describe Charts is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Using Fractions, Decimals and Percents to Describe Charts** by Larry Green is licensed CC BY 4.0.

SECTION OVERVIEW

21.2: The Number Line

21.2.1: Distance between Two Points on a Number Line

21.2.2: Plotting Points and Intervals on the Number Line

21.2.3: Represent an Inequality as an Interval on a Number Line

21.2.4: The Midpoint

This page titled 21.2: The Number Line is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

21.2.1: Distance between Two Points on a Number Line

Learning Outcomes

1. Calculate the distance between two points on a number line when both are non-negative.
2. Calculate the distance between two points on a number line when at least one is negative.

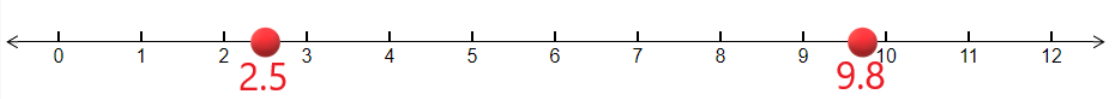
The number line is the main visual base in statistics and we often want to look at two points on the number line and determine the distance between them. This is used to find the base of a rectangle or another figure that lies above the number line. By the end of this section, you will be able to determine the distance between any two points on a number line that comes from a statistics application.

Finding the Distance Between Two Points with Positive Coordinates on a Number Line

The key to finding the distance between two points is to remember that the geometric definition of subtraction is the distance between the two numbers as long as we subtract the smaller number from the larger.

Example 21.2.1.1

Find the distance between the points 2.5 and 9.8 as shown below on the number line.



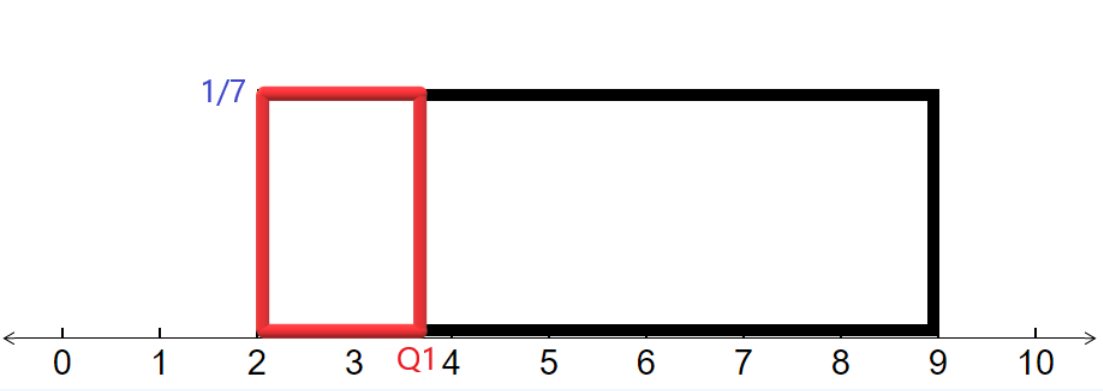
Solution

To find the distance, we just subtract:

$$9.8 - 2.5 = 7.3$$

Example 21.2.1.2

When finding probabilities involving a uniform distribution, we have to find the base of a rectangle that lies on a number line. Find the base of the rectangle shown below that represents a uniform distribution from 2 to 9.



Solution

We just subtract:

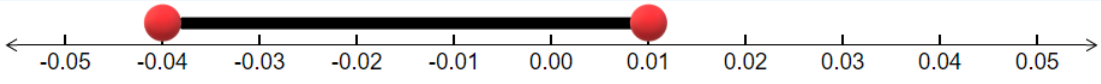
$$9 - 2 = 7$$

Finding the Distance Between Two Points on a Number Line When the Coordinates Are Not Both Positive

In statistics, it is common to have points on a number line where the points are not both positive and we need to find the distance between them.

Example 21.2.1.3

The diagram below shows the confidence interval for the difference between the proportion of men who are planning on going into the health care profession and the proportion of women. What is the width of the confidence interval?



Solution

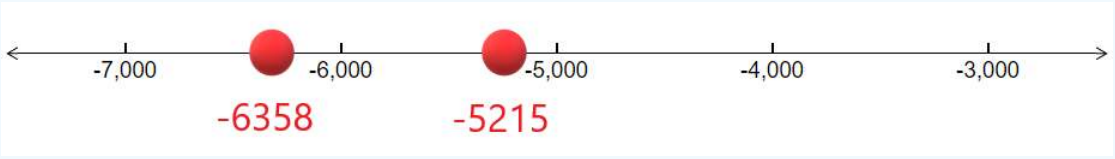
Whenever we want want to find the distance between two numbers, we always subtract. Recall that subtracting a negative number is adding.

$$0.01 - (-0.04) = 0.01 + 0.04 = 0.05$$

Therefore the width of the confidence interval is 0.05.

Example 21.2.1.4

The mean value of credit card accounts is -6358 dollars. A study was done of recent college graduates and found their mean value for their credit card accounts was -5215 dollars. The number line below shows this situation. How far apart are these values?



Solution

We subtract the two numbers and recall that when we subtract two negative numbers when we are looking at the right minus the left, we make them positive and subtract the positive numbers.

$$-5215 - (-6358) = 6358 - 5215 = 1143$$

Thus the mean credit card balances are $1143 apart.

Exercise

In statistics, we are asked to find a z-score, which tells us how unusual an event is. The first step in finding a z-score is to calculate the distance a value is from the mean. The number line below depicts the mean of 18.56 and the value of 20.43. Find the distance between these two points.



- Finding the Distance Between Points on a Number Line
- Integer Subtracton Using the Number Line

This page titled 21.2.1: Distance between Two Points on a Number Line is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Distance between Two Points on a Number Line** by Larry Green is licensed CC BY 4.0.

21.2.2: Plotting Points and Intervals on the Number Line

Learning Outcomes

1. Plot a point on the number line
2. Plot an interval on the number line

The number line is of fundamental importance and is used repeatedly in statistics. It is a tool to visualize all of the possible outcomes of a study and to organize the results of the study. Often a diagram is placed above the number line to provide us with a picture of the results. By the end of this section, you will be able to plot points and intervals on a number line and use these plots to understand the possible outcomes and actual outcomes of studies.

Drawing Points on a Number Line

A number line is just a horizontal line that is used to display all the possible outcomes. It is similar to a ruler in that it helps us describe and compare numbers. Similar to a ruler that can be marked with many different scales such as inches or centimeters, we get to choose the scale of the number line and where the center is.

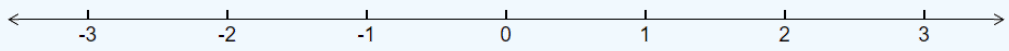
Example 21.2.2.1

The standard normal distribution is plotted above a number line. The most important values are the integers between -3 and 3. The number 0 is both the mean (average) and median (center).

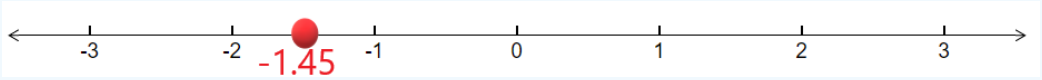
1. Plot the number line that best displays this information.
2. Plot the value -1.45 on this number line.

Solution

1. We sketch a line, mark 0 as the center, and label the numbers -3, -2, -1, 0, 1, 2, 3 from left to right.



2. To plot the point -1.45, we first have to understand that this number is between -1 and -2. It is close to half way between -1 and -2. We put a circle on the number line that is close to halfway between these values as shown below.

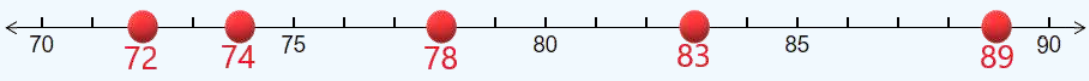


Example 21.2.2.2

When working with box plots, we need to first set up a number line that labels what is called the five point summary: Minimum, First Quartile, Median, Third Quartile, and Maximum. Suppose the five point summary for height in inches for a basketball team is: 72,74,78,83,89. Plot these points on a number line

Solution

When plotting points on a number line, we first have to decide what range of the line we want to show in order to best display the points that appear. Technically all numbers are on every number line, but that does not mean we show all numbers. In this example, the numbers are all between 70 and 90, so we certainly don't need to display the number 0. A good idea is to let 70 be on the far left and 90 be on the far right and then plot the points between them. We also have to decide on the spacing of the tick marks. Since the range from 70 to 90 is 20, this may be too many numbers to display. Instead we might want to count by 5's. Below is the number line that shows the numbers 70 to 90 and counts by 5's. The five point summary is plotted on this line.



Exercise

A histogram will be drawn to display the annual income that experienced registered nurses make. The boundaries of the bars of the histogram are: $81,000, $108,000, $135,000, $162,000, and $189,000. Plot these points on a number line.

Plotting an Interval on a Number Line

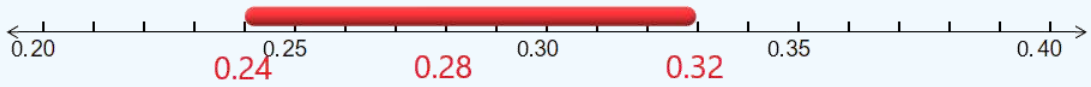
Often in statistics, instead of just having to plot a few points on a number line, we need to instead plot a whole interval on the number line. This is especially useful when we want to exhibit a range of values between two numbers, to the left of a number or to the right of a number.

Example 21.2.2.3

A 95% confidence interval for the proportion of Americans who work on weekends is found to be 0.24 to 0.32, with the center at 0.28. Use a number line to display this information.

Solution

We just draw a number line, include the three key numbers: 0.24, 0.32, and 0.28 and highlight the part of the interval between 0.23 and 0.31.

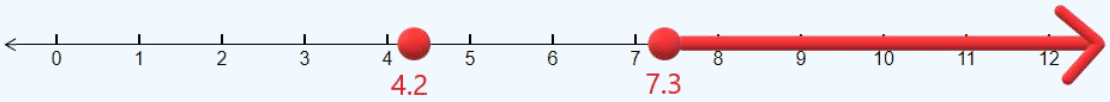


Example 21.2.2.4: rejection region

In Hypothesis testing, we sketch something called the rejection region which is an interval that goes off to infinity or to negative infinity. Suppose that the mean number of hours to work on the week's homework is 4.2. The rejection region for the hypothesis test is all numbers larger than 7.3 hours. Plot the mean and sketch the rejection region on a number line.

Solution

We plot the point 4.2 on the number line and shade everything to the right of 7.3 on the number line.



- Plot Integers on the Number Line
- Intervals: Given an Inequality, Graph the Interval and State Using Interval Notation
- Plotting Points on a Number Line Application

This page titled 21.2.2: Plotting Points and Intervals on the Number Line is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Plotting Points and Intervals on the Number Line** by Larry Green is licensed CC BY 4.0.

21.2.3: Represent an Inequality as an Interval on a Number Line

Learning Outcomes

1. Graph and inequality on a number line.
2. Graph the complement on a number line for both continuous and discrete variables.

Inequalities come up frequently in statistics and it is often helpful to plot the inequality on the number line in order to visualize the inequality. This helps both for inequalities that involve real numbers and for inequalities that refer to just integer values. As an extension of this idea, we often want to look at the complement of an inequality, that is all numbers that make the inequality false. In this section we will look at examples that accomplish this task.

Sketching an Inequality on a number line where the possible values are real numbers.

There are four different inequalities: $<$, $\leq$, $>$, $\geq$. What makes this the most challenging is when they are expressed in words. Here are some of the words that are used for each:

- $<$: "Less Than", "Smaller", "Lower", "Younger"
- $\leq$: "Less Than or Equal to", "At Most", "No More Than", "Not to Exceed"
- $>$: "Greater Than", "Larger", "Higher", "Bigger", "Older", "More Than"
- $\geq$: "Greater Than or Equal to", "At Least", "No Less than"

These are the most common words that correspond to the inequalities, but there are others that come up less frequently.

Example 21.2.3.1

Graph the inequality: $3 < x \leq 5$ on a number line

Solution

First notice that the interval does not include the number 3, but does include the number 5. We can represent not including a number with an open circle and including a number with a closed circle. The number line representation of the inequality is shown below.

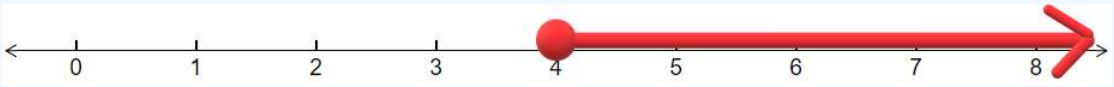


Example 21.2.3.2

In statistics, we often want to find probabilities of an event being at least as large or no more than a given value. It helps to first plot the interval on a number line. Suppose you want to find the probability that you will have to wait in line for at least 4minutes. Sketch this inequality on a number line.

Solution

First, notice that "At Least" has the symbol $\geq$. Thus, we have a closed circle on the number 4. There is no upper bound, so we draw a long arrow from 4 to the right of 4. The solution is shown below



Example 21.2.3.3

Another main topic that comes up in statistics is confidence intervals. For example in recent poll to see the percent of Americans who think that Congress is doing a good job found that a 95% confidence interval had lower bound of 0.18 and an upper bound of 0.24. This can be written as [0.18,0,24]. Sketch this interval on the number line.

Solution

The first thing we need to do is decide on the tick marks to put on the number line. If we counted by 1's, then the interval of interest would be too small to stand out. Instead we will count by 0.1's. The number line is shown below.

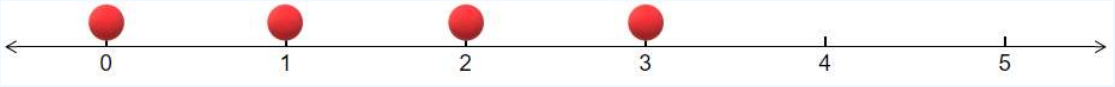


Example 21.2.3.4

Often in statistics, we deal with discrete variables. Most of the time this will mean that only whole number values can occur. For example, you want to find out the probability that a college student is taking at most three classes. Graph this on a number line.

Solution

First note that the outcomes can only be whole numbers. Second, note that "at most" means $\leq$. Thus the possible outcomes are: 0, 1, 2, and 3. The number line below displays these outcomes.



Graphing the Complement

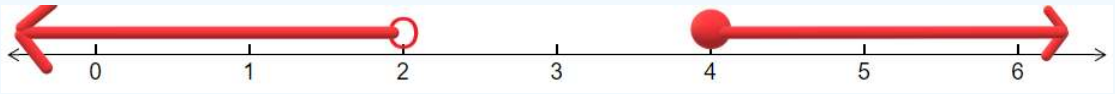
In statistics, we often want to graph the complement of an interval. The complement means everything that is not in the interval.

Example 21.2.3.5

Graph the complement of the interval [2,4).

Solution

Notice that the complement of numbers inside the interval between 2 and 4 is the numbers outside that interval. This will consist of the numbers to the left of 2 and to the right of 4. Since the number 2 is included in the original interval, it will not be included in the complement. Since the number 4 is not included in the original interval, it will be included in the complement. The complement is shown on the number line below.

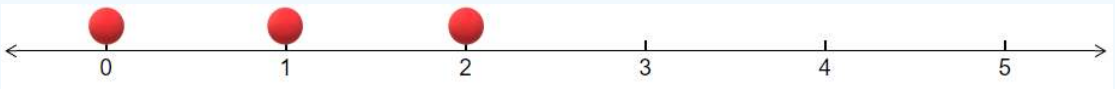


Example 21.2.3.6

Some calculators can only find probabilities for values less than a certain number. If we want the probability of an interval greater than a number, we need to use the complement. Suppose that you want to find the probability that a person will have traveled to more than two foreign countries in the last twelve months. Find the complement of this and graph it on a number line.

Solution

First notice that only whole numbers are possible since it does not make sense to go to a fractional number of countries. Second note that the lowest number that is more than 2 is 3. If 3 is included in the original list, then 3 will not be included in the complement. Thus, the highest number that is in the complement of "more than 2" is 2. The number line below shows the complement of more than 2.



Exercise

Suppose you want to find the probability that at least 4 people in your class have a last name that contains the letter "W". To make this calculation you will need to first find the complement of "at least 4". Sketch this complement on the number line.

- Intervals: Given an Inequality, Graph the Interval and State Using Interval Notation
- Express Inequalities as a Graph and Interval Notation
- Sketching the Complement of an Interval on a Number Line

This page titled 21.2.3: Represent an Inequality as an Interval on a Number Line is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Represent an Inequality as an Interval on a Number Line** by Larry Green is licensed CC BY 4.0.

21.2.4: The Midpoint

Learning Outcomes

1. Find the midpoint between two numbers.
2. Sketch the midpoint of two numbers on a number line.

As the word sounds, "midpoint" means "the point in the middle". Finding a midpoint is not too difficult and has applications in many areas of statistics, from confidence intervals to sketching distributions, to means.

Finding the Midpoint Between Two Numbers

If we are given two numbers, then the midpoint is just the average of the two numbers. To calculate the midpoint, we add them up and then divide the result by 2. The formula is as follows:

Definition: the Midpoint

Let a and b be two numbers. Then the midpoint, M of these two numbers is

$$M = \frac{a+b}{2} \tag{21.2.4.1}$$

Example 21.2.4.1

Find the midpoint of the numbers **3.5** and **7.2**.

Solution

The most important thing about finding the midpoint is that the addition of the two numbers must occur before the division by 2. We can either do this one step at a time in our calculator or we can enclose the sum in parentheses. In this example we will perform the addition first:

$$3.5 + 7.2 = 10.7$$

Now we are ready to divide by 2:

$$\frac{10.7}{2} = 5.35$$

Thus the midpoint of 3.5 and 7.2 is 5.35.

Example 21.2.4.2

A major topic in statistics is the confidence interval which tells us the most likely interval that the mean or the proportion will lie in. Often the lower and upper bound of the confidence interval are given, but the midpoint of these two numbers is the best guess for what we are looking for. Suppose a 95% confidence interval for the difference between two means is -1.34 and 2.79. Find the midpoint of these numbers, which is the best guess for the difference between the two means.

Solution

We use the formula for the midpoint (Equation 21.2.4.1):

$$M = \frac{a+b}{2} = \frac{-1.34 + 2.79}{2}$$

Now let's use a calculator. We will need parentheses around the numerator:

$$(-1.34 + 2.79) \div 2 = 0.725$$

Thus, the midpoint of the numbers -1.34 and 2.79 is 0.725.

Sketching the Midpoint on a Number Line

Visualizing the midpoint can often reveal it much better than just writing down its value. The diagrams are of fundamental importance in statistics.

Example 21.2.4.3

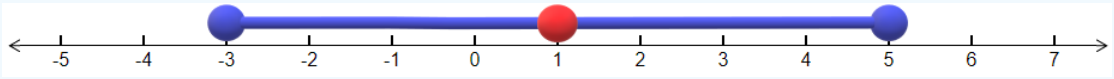
Sketch the points -3, 5 and the midpoint of these two numbers on a number line.

Solution

We start by finding the midpoint using the midpoint formula (Equation 21.2.4.1):

$$M = \frac{-3+5}{2} = (-3+5) \div 2 = 1$$

Now we sketch these three points on the number line:

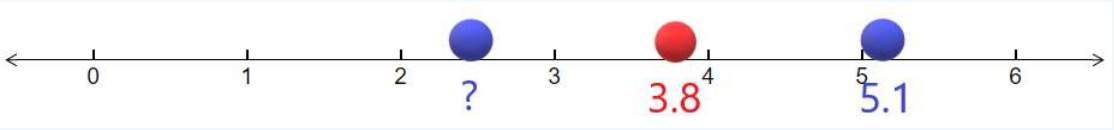


Example 21.2.4.4: hypothesis testing

Another application of the midpoint involves hypothesis testing. Sometimes we are given the hypothesized mean, which is the midpoint. We are also given the sample mean, which is either the left or right endpoint. The goal is to find the other endpoint. Suppose that the midpoint (hypothesized mean) is at 3.8 and the right endpoint (sample mean) is at 5.1. Find the value of the left endpoint.

Solution

It helps to sketch the diagram on the number line as shown below.



Now since 3.8 is the midpoint, the distance from the left endpoint to the midpoint is equal to the distance from 3.8 to 5.1. The distance from 3.8 to 5.1 is:

$$5.1 - 3.8 = 1.3$$

Therefore the left endpoint is 1.3 to the left of 3.8. This can be found by subtracting the two numbers:

$$3.8 - 1.3 = 2.5$$

Therefore the left endpoint is at 2.5.

Exercise

Suppose that the midpoint (hypothesized proportion) is at 0.31 and the left endpoint (sample proportion) is at 0.28. Find the value of the right endpoint.

- Midpoint on the Number line
- Finding the Right Endpoint Given the Left Endpoint and Midpoint

---

This page titled 21.2.4: The Midpoint is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **The Midpoint** by Larry Green is licensed CC BY 4.0.

SECTION OVERVIEW

21.3: Operations on Numbers

21.3.1: Area of a Rectangle

21.3.2: Factorials and Combination Notation

21.3.3: Order of Operations

21.3.4: Order of Operations in Expressions and Formulas

21.3.5: Perform Signed Number Arithmetic

21.3.6: Powers and Roots

21.3.7: Using Summation Notation

This page titled 21.3: Operations on Numbers is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

21.3.1: Area of a Rectangle

Learning Outcomes

- Find the area of a rectangle.
- Find the height of a rectangle given that the area is equal to 1.

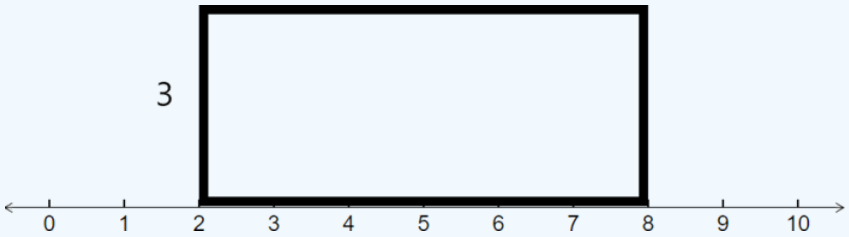
Rectangles are of fundamental importance in the portion of statistics that involves the uniform distribution. Every rectangle has a base and a height and an area. The formula for the area of a rectangle is:

$$\text{Area} = \text{Base} \times \text{Height} \tag{21.3.1.1}$$

When working with the uniform distribution, the area represents the probability of an event being within the bounds of the base.

Example 21.3.1.1

Consider the rectangle shown below.



Find the area of this rectangle.

Solution

We use the Area formula (Equation 21.3.1.1). To find the base, we notice that it runs from 2 to 8, so we subtract these numbers to get the base:

$$Base = 8 - 2 = 6$$

Next multiply by the height, 3, to get

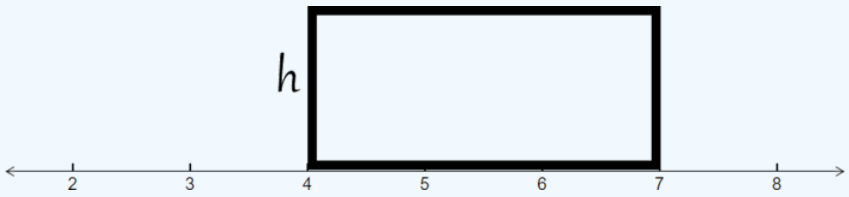
$$Area = Base \times Height = 6 \times 3 = 18$$

Example 21.3.1.2

It turns out that the area of the rectangles that equal to 1 will occur the most often for a uniform distribution. Suppose that we know that the area of a rectangle that depicts a uniform distribution is equal to 1 and that the base of the rectangle goes from 4 to 7. Find the height of the rectangle.

Solution

First sketch the rectangle below, labeling the height as h.



Next, find the base of the rectangle that goes from 4 to 7 by subtracting:

$$Base = 7 - 4 = 3$$

Next, plug in what we know into the area equation:

$$1 = Area = Base \times Height = 3 \times h$$

This tell us that 3 times a number is equal to 1. To find out what the number is, we just divide both sides by 3 to get:

$$h = \frac{1}{3}$$

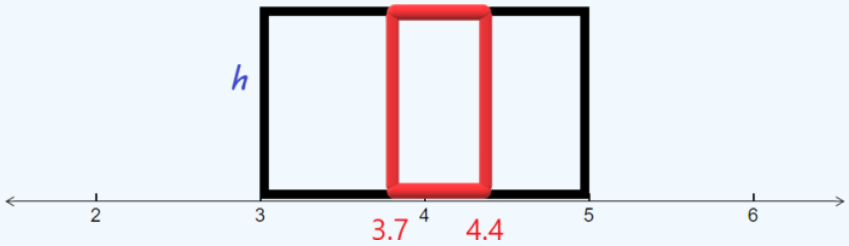
Therefore the height of an area 1 rectangle with base from 4 to 7 is $\frac{1}{3}$.

Example 21.3.1.3

Suppose that we know that the area of a rectangle that depicts a uniform distribution is equal to 1 and that the base of the rectangle goes from 3 to 5. There is a smaller rectangle within the larger one with the same height, but whose base goes from 3.7 to 4.4. Find the area of the smaller rectangle.

Solution

First, sketch the larger rectangle with the smaller rectangle shaded in.



Next, we find the height of the rectangle. We know that the area of the larger rectangle is 1. The base goes from 3 to 5, so the base is $5 - 3 = 2$ Hence:

$$1 = Area = Base \times Height = 2h$$

Dividing by 2, gives us that the height is $\frac{1}{2}$ or 0.5. Now we are ready to find the area of the smaller rectangle. We first find the base by subtracting:

$$\text{Base} = 4.4 - 3.7 = 0.7$$

Next, use the area formula:

$$Area = Base \times Height = 0.7 \times 0.5 = 0.35$$

Exercise 21.3.1.1

Suppose that elementary students' ages are uniformly distributed from 5 to 11 years old. The rectangle that depicts this has base from 5 to 11 and area 1. The rectangle that depicts the probability that a randomly selected child will be between 6.5 and 8.6 years old has base from 6.5 to 8.6 and the same height as the larger rectangle. Find the area of the smaller rectangle

- Ex: Determine the Area of a Rectangle Involving Whole Numbers
- Area of a Rectangle and the Uniform Distribution

This page titled 21.3.1: Area of a Rectangle is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Area of a Rectangle** by Larry Green is licensed CC BY 4.0.

21.3.2: Factorials and Combination Notation

Learning Outcomes

1. Evaluate a factorial.
2. Use combination notation for statistics applications.

When we need to compute probabilities, we often need to multiple descending numbers. For example, if there is a deck of 52 cards and we want to pick five of them without replacement, then there are 52 choices for the first pick, 51 choices for the second pick since one card has already been picked, 50 choices for the third, 49 choices for the fourth, and 48 for the fifth. If we want to find out how many different outcomes there are, we can use what we call the multiplication principle and multiple them: $52 \times 51 \times 50 \times 49 \times 48$. If we wanted to pick all 52 of the cards one at a time, then this list would be excessively long. Instead there is a notation that describes multiplying all the way down to 1, called the factorial. It must be exciting, since we use the symbol "!" for the factorial.

Example 21.3.2.1

Calculate $4!$

Solution

We use the definition which says start at 4 and multiply until we get to 1:

$$4! = 4 \times 3 \times 2 \times 1 = 24$$

Example 21.3.2.2

If we pick 5 cards from a 52 card deck without replacement and the same two sets of 5 cards, but in different orders, are considered different, how many sets of 5 cards are there?

Solution

From the introduction, the number of sets is just:

$$52 \times 51 \times 50 \times 49 \times 48$$

This is not quite a factorial since it stops at 48; however, we can think of this as $52!$ with $47!$ removed from it. In other words we need to find

$$\frac{52!}{47!}$$

We could just multiply the numbers from the original list, but it is a good idea to practice with your calculator or computer to find this using the ! symbol. When you do use technology, you should get:

$$\frac{52!}{47!} = 311,875,200$$

Combinations

One of the most important applications of factorials is combinations which count the number of ways of selecting a smaller collection from a larger collection when order is not important. For example if there are 12 people in a room and you want to select a team of 4 of them, then the number of possibilities uses combinations. Here is the definition:

Definition: Combinations

The number of ways of selecting k items without replacement from a collection of n items when order does not matter is:

$$\binom{n}{r} = {}_nC_r = \frac{n!}{r!\,(n-r)!} \qquad (21.3.2.1)$$

Notice that there are a few notations. The first is more of a mathematical notation while the second is the notation that a calculator uses. For example, in the TI 84+ calculator, the notation for the number of combinations when selecting 4 from a collection of 12 is:

$$12\ {}_nC_r\ 4$$

There are many internet sites that will perform combinations. For example the math is fun site asks you to put in n and r and also state whether order is important and repetition is allowed. If you click to make both "no" then you will get the combinations.

Example 21.3.2.3

Calculate

$$\binom{15}{11} = {}_{15}\ C_{11}$$

Solution

Whether you use a hand calculator or a computer you should get the number: **1365**

Example 21.3.2.4

The probability of winning the Powerball lottery if you buy one ticket is:

$$P(win) = \frac{1}{{}_{69}C_5 \times 26}$$

Calculate this probability.

Solution

First, let's calculate ${}_{69}C_5$. Using a calculator or computer, you should get 11,238,513. Next, multiply by 26 to get

$$11,238,513 \times 26 = 292,201,338$$

Thus, there is a one in 292,201,338 chance of winning the Powerball lottery if you buy a ticket. We can also write this as a decimal by dividing:

$$P\,(win) = \frac{1}{292,201,338} = 0.000000003422$$

As you can see, your chances of winning the Powerball are very small.

Exercise

A classroom is full of 28 students and there will be one president of the class and a "Congress" of 4 others selected. The number of different leadership group possibilities is:

$$28 \times {}_{27}\ C_4$$

Calculate this number to find out how many different leadership group possibilities there are.

Ex 1: Simplify Expressions with Factorials

Combinations

Combinations

This page titled 21.3.2: Factorials and Combination Notation is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Factorials and Combination Notation** by Larry Green is licensed CC BY 4.0.

21.3.3: Order of Operations

Learning Outcomes

1. Use the order of operations to correctly perform multi-step arithmetic
2. Apply the order of operations to statistics related complex questions.

When we are given multiple arithmetic operations within a calculation, there is a, established order that we must do them in based on how the expression is written. Understanding these rules is especially important when using a calculator, since calculators are programmed to strictly follow the order of operations. This comes up in every topic in statistics, so knowing the order of operations is an essential skill for all successful statistics students to have.

PEMDAS

The order of operations are as follows:

1. **P**arentheses
2. **E**xponents
3. **M**ultiplication and **D**ivision
4. **A**ddition and **S**ubtraction

When there is a tie, the rule is to go from left to right.

Notice that Multiplication and division are listed together as item 3. If you see multiplication and division in the same expression the rule is to go from left to right. Similarly, if you see addition and subtraction in the same expression the rule is to from go left to right. The same goes for two of the same arithmetic operators.

Example 21.3.3.1

Evaluate: $20 - 6 \div 3 + (2 \times 3^2)$

Solution

We start with what is inside the parentheses: $2 + 3^2$. Since exponents comes before addition, we find $3^2 = 9$ first. We now have

$$20 - 6 \div 3 + (2 \times 9)$$

We continue inside the parentheses and perform the multiplication: $2 \times 9 = 18$.

This gives

$$20 - 6 \div 3 + 18$$

Since division comes before addition and subtraction, we next calculate $6 \div 3 = 2$ to get

$$20 - 2 + 18$$

Since subtraction and addition are tied, we go from left to right. We calculate: $20 - 2 = 18$ to get

$$18 + 18 \ = 36$$

The key to arriving at the correct answer is to go slow and write down each step in the arithmetic.

Hidden Parentheses

You may think that since you always have a calculator or computer at hand, that you don't need to worry about order of operations. Unfortunately, the way that expressions are written is not the same as the way that they are entered into a computer or calculator. In particular, exponents need to be treated with care as do fractions bars.

Example 21.3.3.3

Evaluate 2.1^{6-2}

Solution

First, note that we use the symbol "^" to tell a computer or calculator to exponentiate. If you were to enter 2.1^6-2 into a computer, it would give you the answer of 83.766121 which is not correct, since the computer will first expontiate and then subtract. Since the subtraction is within the exponent, it must be performed first. To tell a calculator or computer to perform the subtraction first, we use parentheses:

2.1^(6 - 2) = 19.4481

Example 21.3.3.4: z-scores

The "z-score" is defined by:

$$z = \frac{x - \mu}{\sigma}$$

Find the z-score rounded to one decimal place if:

$$x = 2.323,\ \mu = 1.297,\ \sigma = 0.241$$

Solution

Once again, if we put these numbers into the z-score formula and use a computer or calculator by entering $3.323 - 1.297 \div 0.241$ we will get -0.259 which is the wrong answer. Instead, we need to know that the fraction bar separates the numerator and the denominator, so the subtraction must be done first. We compute

$$\frac{2.323 - 1.297}{0.241} = (2.323 - 1.297) \div 0.241 = 4.25726141$$

Now round to one decimal place to get 4.3. Notice that if you rounded before you did the arithmetic, you would get exactly 5 which is very different. 4.3 is more accurate.

Exercise

Suppose the equation of the regression line for the number of pairs of socks a person owns, y, based on the number of pairs of shoes, x, the person owns is

$$\hat{y} = 6 + 2x$$

Use this regression line to predict the number of pairs of socks a person owns for a person who owns 4 pairs of shoes.

- Order of Operations - The Basics
- Order of Operations

This page titled 21.3.3: Order of Operations is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Order of Operations** by Larry Green is licensed CC BY 4.0.

21.3.4: Order of Operations in Expressions and Formulas

Learning Outcomes

- Use Order of Operations in Statistics Formulas.

We have already encountered the order of operations: Parentheses, Exponents, Multiplication and Division, Addition and Subtraction. In this section, we will give some additional examples where the order of operations must be used properly to evaluate statistics.

Example 21.3.4.1

The sample standard deviation asks us to add up the squared deviations, take the square root and divide by one less than the sample size. For example, suppose that there are three data values: 3, 5, 10. The mean of these values is 6. Then the standard deviation is:

$$s = \sqrt{\frac{(3-6)^2 + (5-6)^2 + (10-6)^2}{3-1}}$$

Evaluate this number rounded to the nearest hundredth.

Solution

The first thing in the order of operations is to do what is in the parentheses. We must subtract:

$$3-6=-3, \quad 5-6 = -1, \quad 10-6=4$$

We can substitute the numbers in to get:

$$= \sqrt{\frac{(-3)^2 + (-1)^2 + (4)^2}{3-1}}$$

Next, we exponentiate:

$$(-3)^2 = 9, \quad (-1)^2 = 1, \quad 4^2 = 16$$

Substitute these in to get:

$$\sqrt{\frac{9+1+16}{3-1}}$$

We can now perform the addition inside the square root to get:

$$\sqrt{\frac{26}{3-1}}$$

Next, perform the subtraction of the denominator to get:

$$\sqrt{\frac{26}{2}}$$

We can divide to get:

$$\sqrt{13}$$

We don't want to do this by hand, so in a calculator or computer type in:

$$13^{0.5} = 3.61$$

Example 21.3.4.2

When calculating the probability that a value will be less than 4.6 if the value is taken randomly from a uniform distribution between 3 and 7, we have to calculate:

$$(4.6-3)\times\frac{1}{7-3}$$

Find this probability.

Solution

We can use a calculator or computer, but we must be very careful about the order of operations. Notice that there are implied parentheses due to the fraction bar. The answer is:

$$\frac{(4.6-3)\times 1}{7-3}$$

Using technology, we get:

$$(4.6-3)\times\frac{1}{7-3}=0.4$$

Exercise

When finding the upper bound, U, of a confidence interval given the lower bound, L, and the margin of error, E, we use the formula

$$U=L+2E$$

Find the upper bound of the confidence interval for the proportion of babies that are born preterm if the lower bound is 0.085 and the margin of error is 0.03.

- Ex: Evaluate an Expression Using the Order of Operations
- Order of Operations and Confidence Intervals

This page titled 21.3.4: Order of Operations in Expressions and Formulas is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Order of Operations in Expressions and Formulas** by Larry Green is licensed CC BY 4.0.

21.3.5: Perform Signed Number Arithmetic

Learning Outcomes

1. Add signed numbers.
2. Subtract signed numbers.
3. Multiply signed numbers.
4. Divide signed numbers.

Even though negative numbers seem not that common in the real world, they do come up often when doing comparisons. For example, a common question is how much bigger is one number than another, which involves subtraction. In statistics we don't know the means until we collect the data and do the calculations. This often results in subtracting a larger number from a smaller number which yields a negative number. Because of this and for many other reasons, we need to be able to perform arithmetic on both positive and negative numbers.

Adding Signed Numbers

We will assume that you are very familiar with adding positive numbers, but when there are negative numbers involved, there are some rules to follow:

1. When adding two negative numbers, ignore the negative signs, add the positive numbers and then make the result negative.
2. When adding two numbers such that one is positive and the other is negative, ignore the sign, subtract the smaller from the larger. If the larger of the positive numbers was originally negative, then make the result negative. Otherwise keep the result positive.

Example 21.3.5.1

Add:

$$-4+(-3)$$

Solution

First we ignore the signs and add the positive numbers.

$$4+3=7$$

Next we make the result negative.

$$-4+(-3)=-7$$

Example 21.3.5.2

Add:

$$-2+5$$

Solution

Since one of the numbers is positive and the other is negative, we subtract:

$$5-2=3$$

Of the two numbers, 2 and 5, 5 is the larger one and started positive. Hence we keep the result positive:

$$-2+5=3$$

Subtracting Numbers

Subtraction comes up often when we want to find the width of an interval in statistics. Here are the cases for subtracting: $a-b$:

1. If $a \geq b \geq 0$, then this is just ordinary subtraction.

2. If $b \geq a \geq 0$, then find $b - a$ and make the result negative.
3. If $a < 0$, $b \geq 0$, then make both positive, add the two positive numbers and make the result negative.
4. If $b < 0$ then you use the rule that subtracting a negative number is the same as adding the positive number.

Example 21.3.5.3

Evaluate $5 - 9$

Solution

Since 9 is bigger than 5, we subtract:

$$9 - 5 = 4$$

Next, we make the result negative to get:

$$5 - 9 = -4$$

Example 21.3.5.4

Evaluate $-9 - 4$

Solution

We are in the case $a < 0$, $b \geq 0$. Therefore, we first make both positive and add the positive numbers.

$$9 + 4 = 13$$

The final step is to make the answer negative to get

$$-9 - 4 = -13$$

Example 21.3.5.5: Uniform distributions

In statistics, we call a *distribution Uniform* if an event is just as likely to be in any given interval within the bounds as any other interval within the bounds as long as the intervals are both of the same width. Finding the width of a given interval is usually the first step in solving a question involving uniform distributions. Suppose that the temperature on a winter day has a Uniform distribution on [-8,4]. Find the width of this interval

Solution

To find the width of an interval, we subtract the left endpoint from the right endpoint:

$$4 - (-8)$$

Since we are subtracting a negative number, the "-" signs become addition:

$$4 - (-8) = 4 + 8 = 12$$

Thus the width of the interval is 12.

Multiplying and Dividing Signed Numbers

When we have a multiplication or division problem, we just remember that two negatives make a positive. So if there are an even number of negative numbers that are multiplied or divided, the result is negative. If there are an odd number of negative numbers that are multiplied or divided, the result is positive.

Example 21.3.5.6

Perform the arithmetic:

$$\frac{(-6)(-10)}{(-4)(-5)}$$

Solution

First, just ignore all of the negative signs and multiply the numerator and denominator separately:

$$\frac{(6)(10)}{(4)(5)} = \frac{60}{20}$$

Now divide:

$$\frac{60}{20} = \frac{6}{2} = 3$$

Finally, notice that there are four negative numbers in the original multiplication and division problem. Four is an even number, so the answer is positive:

$$\frac{(-6)(-10)}{(-4)(-5)} = 3$$

Example 21.3.5.7

A confidence interval for the population mean difference in books read per year by men and women was was found to be [-4,1]. Find the midpoint of this interval.

Solution

First recall that to find the midpoint of two numbers, we add then and then divide by 2. Hence, our first step is to add -4 and 1. Since 1 is positive and -4 is negative, we first subtract the two numbers:

$$4 - 1 = 3$$

Of the two numbers, 4 and 1, 4 is the larger one and started negative. Hence we change the sign to negative::

$$-4 + 1 = -3$$

The final step in finding the midpoint is to divide by 2. First we divide them as positive numbers:

$$\frac{3}{2} = 1.5$$

Since the original quotient has a single negative number (an odd number of negative numbers), the answer is negative. Thus the midpoint of -4 and 1 is -1.5.

Exercise

The difference between the observed value and the expected value in linear regression is called the residual. Suppose that the three observed values are: -4, 2, and 5. The expected values are -3, 7, and -1. First find the residuals and then find the sum of the residuals.

- Signed Number Operations (L1.4)
- signed arithmetic

This page titled 21.3.5: Perform Signed Number Arithmetic is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Perform Signed Number Arithmetic** by Larry Green is licensed CC BY 4.0.

21.3.6: Powers and Roots

Learning Outcomes

1. Raise a number to a power using technology.
2. Take the square root of a number using technology.
3. Apply the order of operations when there is root or a power.

It can be a challenge when we first try to use technology to raise a number to a power or take a square root of a number. In this section, we will go over some pointers on how to successfully take powers and roots of a number. We will also continue our practice with the order of operations, remembering that as long as there are no parentheses, exponents always come before all other operations. We will see that taking a power of a number comes up in probability and taking a root comes up in finding standard deviations.

Powers

Just about every calculator, computer, and smartphone can take powers of a number. We just need to remember that the symbol "^" is used to mean "to the power of". We also need to remember to use parentheses if we need to force other arithmetic to come before the exponentiation.

Example 21.3.6.1

Evaluate: 1.04^5 and round to two decimal places.

Solution

This definitely calls for the use of technology. Most calculators, whether hand calculators or computer calculators, use the symbol "^" (shift 6 on the keyboard) for exponentiation. We type in:

$$1.04^5 = 1.2166529$$

We are asked to round to two decimal places. Since the third decimal place is a 6 which is 5 or greater, we round up to get:

$$1.04^5 \approx 1.22$$

Example 21.3.6.2

Evaluate: $2.8^{5.3\times 0.17}$ and round to two decimal places.

Solution

First note that on a computer we use "*" (shift 8) to represent multiplication. If we were to put in 2.8 ^ 5.3 * 0.17 into the calculator, we would get the wrong answer, since it will perform the exponentiation before the multiplication. Since the original question has the multiplication inside the exponent, we have to force the calculator to perform the multiplication first. We can ensure that multiplication occurs first by including parentheses:

$$2.8^{5.3\times 0.17} = 2.52865$$

Now round to decimal places to get:

$$2.8^{5.3\times 0.17} \approx 2.53$$

Example 21.3.6.3

If we want to find the probability that if we toss a six sided die five times that the first two rolls will each be a 1 or a 2 and the last three die rolls will be even, then the probability is:

$$\left(\frac{1}{3}\right)^2 \times \left(\frac{1}{2}\right)^3$$

What is this probability rounded to three decimal places?

Solution

We find:

$$(1/3)^2(1/2)^3 \approx 0.013888889$$

Now round to three decimal places to get

$$\left(\frac{1}{3}\right)^2 \times \left(\frac{1}{2}\right)^3 \approx 0.014$$

Square Roots

Square roots come up often in statistics, especially when we are looking at standard deviations. We need to be able to use a calculator or computer to compute a square root of a number. There are two approaches that usually work. The first approach is to use the $\sqrt{}$ symbol on the calculator if there is one. For a computer, using sqrt() usually works. For example if you put 10*sqrt(2) in the Google search bar, it will show you 14.1421356. A second way that works for pretty much any calculator, whether it is a hand held calculator or a computer calculator, is to realize that the square root of a number is the same thing as the number to the 1/2 power. In order to not have to wrap 1/2 in parentheses, it is easier to type in the number to the 0.5 power.

Example 21.3.6.3

Evaluate $\sqrt{42}$ and round your answer to two decimal places.

Solution

Depending on the technology you are using you will either enter the square root symbol and then the number 42 and then close the parentheses if they are presented and then hit enter. If you are using a computer, you can use sqrt(42). The third way that will work for both is to enter:

$$42^{0.5} \approx 6.4807407$$

You must then round to two decimal places. Since 0 is less than 5, we round down to get:

$$\sqrt{42} \approx 6.48$$

Example 21.3.6.4

The "z-score" is for the value of 28 for a sampling distribution with sample size 60 coming from a population with mean 28.3 and standard deviation 5 is defined by:

$$z = \frac{28 - 28.3}{\frac{5}{\sqrt{60}}}$$

Find the z-score rounded to two decimal places.

Solution

We have to be careful about the order of operations when putting it into the calculator. We enter:

$$(28 - 28.3)/(5/60\text{^}0.5) = -0.464758$$

Finally, we round to 2 decimal places. Since 4 is smaller than 5, we round down to get:

$$z = \frac{28 - 28.3}{\frac{5}{\sqrt{60}}} = -0.46$$

Exercise

The standard error, which is an average of how far sample means are from the population mean is defined by:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$$

where $\sigma_{\bar{x}}$ is the standard error, σ is the standard deviation, and n is the sample size. Find the standard error if the population standard deviation, σ, is 14 and the sample size, n, is 11.

- Square Root on the TI-83plus and TI-84 family of Calculators
- Square Roots with a Computer

This page titled 21.3.6: Powers and Roots is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Powers and Roots** by Larry Green is licensed CC BY 4.0.

21.3.7: Using Summation Notation

Learning Outcomes

1. Evaluate an expression that includes summation notation.
2. Apply summation notation to calculate statistics.

This notation is called summation notation and appears as:

$$\sum_{i=1}^{n} a_i$$

In this notation, the a_i is an expression that contains the index i and you plug in 1 and then 2 and then 3 all the way to the last number n and then add up all of the results.

Example 21.3.7.1

Calculate

$$\sum_{i=1}^{4} 3i$$

Solution

First notice that i = 1, then 2, then 3 and finally 4. We are supposed to multiply each of these by 3 and add them up:

$$\begin{aligned}\sum_{i=1}^{4} 3i &= 3(1) + 3(2) + 3(3) + 3(4) \\ &= 3 + 6 + 9 + 12 \;=\; 30\end{aligned}$$

Example 21.3.7.2

The formula for the sample mean, sometimes called the average, is

$$\bar{x} \;=\; \frac{\sum_{i-1}^{n} x_i}{n}$$

A survey was conducted asking 8 older adults how many sexual partners they have had in their lifetime. Their answers were {4,12,1,3,4,9,24,7}. Use the formula to find the sample mean.

Solution

Notice that the numerator of the formula just tells us to add the numbers up. Computing the numerator first gives:

$$\sum_{i=1}^{8} x_i = 4 + 12 + 1 + 3 + 4 + 9 + 24 + 7 \;= 64$$

Now that we have the numerator calculated, the formula tells us to divide by *n*, which is just 8. We have:

$$\bar{x} \;= \frac{64}{8} = 8$$

Thus, the sample mean number of sexual partners this group had in their lifetimes is 8.

Example 21.3.7.3

The next most important statistic is the standard deviation. The formula for the sample standard deviation is:

$$s = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}}$$

Let's consider the data in the previous example. Find the standard deviation.

Solution

The formula is quite complicated, but if tackle it one piece at a time using the order of operations properly, we can succeed in finding the sample standard deviation for the data. Notice that there are parentheses, so based on the order of operations, we must do the subtraction within the parentheses first. Since this is all part of the sum, we have eight different subtractions to do. From our calculations in the previous example, the sample mean was $\bar{x} = 8$. We compute the 8 subtractions:

$$4 - 8 = -4,\ 12 - 8 = 4,\ 1 - 8 = -7,\ 3 - 8 = -5,$$

$$4 - 8 = -4,\ 9 - 8 = 1,\ 24 - 8 = 16,\ 7 - 8 = -1$$

The next arithmetic to do is to square each of the differences to get:

$$(-4)^2 = 16,\ (4)^2 = 16, (-7)^2 = 49,\ (-5)^2 = 25,$$

$$(-4)^2 = 16,\ 1^2 = 1,\ 16^2 = 256,\ (-1)^2 = 1$$

Now we have all the entries in the summation, so we add them all up:

$$16 + 16 + 49 + 25 + 16 + 1 + 256 + 1 = 380$$

Now we can write

$$s = \sqrt{\frac{380}{8-1}} = \sqrt{\frac{380}{7}}$$

We can put this into the calculator or computer to get:

$$s = \sqrt{\frac{380}{7}} = 7.3679$$

Exercise: expected value

The expected value, EV, is defined by the formula

$$EV = \sum_{i=1}^{n} x_i\, P(x_i)$$

Where x_i are the possible outcomes and $P(x_i)$ are the probabilities of the outcomes occurring. Suppose the table below shows the number of eggs in a bald eagle clutch and the probabilities of that number occurring.

Probability Distribution Table with Outcomes, x, and probabilities, P(x)

x	1	2	3	4
P(x)	0.2	0.4	0.3	0.1

Find the expected value.

Ex 1: Find a Sum Written in Summation / Sigma Notation

Summation Notation and Expected Value

This page titled 21.3.7: Using Summation Notation is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Using Summation Notation** by Larry Green is licensed CC BY 4.0.

SECTION OVERVIEW

21.4: Sets

21.4.1: Set Notation

21.4.2: The Complement of a Set

21.4.3: The Union and Intersection of Two Sets

21.4.4: Venn Diagrams

This page titled 21.4: Sets is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

21.4.1: Set Notation

Learning Outcomes

1. Read set notation.
2. Describe sets using set notation.

A set is just a collection of items and there are different ways of representing a set. We want to be able to both read the various ways and be able to write down the representation ourselves in order to best display the set. We have already seen how to represent a set on a number line, but that can be cumbersome, especially if we want to just use a keyboard. Imagine how difficult it would be to text a friend about a cool set if the only way to do this was with a number line. Fortunately, mathematicians have agreed on notation to describe a set.

Example 21.4.1.1

If we just have a few items to list, we enclose them in curly brackets "{" and "}" and separate the items with commas. For example,

$$\{\text{Miguel, Kristin, Leo, Shanice}\}$$

means the set the contains these four names.

Example 21.4.1.2

If we just have a long collection of numbers that have a clear pattern, we use the "..." notation to mean "start here, keep going, and end there". For example,

$$\{3, 6, 9, 12, ..., 90\}$$

This set contains more than just the five numbers that are shown. It is clear that the numbers are separated by three each. After the 12, even though it is not explicitly shown, is a 15 which is part of this set. It also contains 18, 21 and keeps going including all the multiples of 3 until it gets to its largest number 90.

Example 21.4.1.3

If we just have a collection of numbers that have a clear pattern, but never ends, we use the "..." without a number at the end. For example,

$$\left\{\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \cdots\right\}$$

This set contains an infinite number of fractions, since there is no number followed by the "...".

Example 21.4.1.4

Sometimes we have a set that it best described by stating a rule. For example, if you want to describe the set of all people who are over 18 years old but not 30 years old, you announce the conditions by putting them to the left of a vertical line segment. We read the line segment as "such that".

$$\{x \mid x > 18 \text{ and } x \neq 30\}$$

This can be read as "the set of all numbers x such that x is greater than 18 and x is not equal to 30".

Exercise

Describe using set notation the collection of all positive even whole numbers that are not equal to 20 or 50.

- Set-Builder Notation

- https://youtu.be/VGphtczN0-c

This page titled 21.4.1: Set Notation is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Set Notation** by Larry Green is licensed CC BY 4.0.

21.4.2: The Complement of a Set

Learning Outcomes

1. Determine the complement of a set.
2. Write the complement of a set using set notation.

We saw in the section "Represent an Inequality as an Interval on a Number Line" how to graph the complement for a set defined by an inequality. Complements come up very often in statistics, so it is worth revisiting this, but instead of graphically we will focus on set notation. Recall that the complement of a set is everything that is not in that set. Sometimes it is much easier to find the probability of a complement than of the original set, and there is an easy relationship between the probability of an event happening and the probability of the complement of that event happening.

$$P(A) = 1 - P(not\ A)$$

Example 21.4.2.1

Find the complement of the set:

$$A = \{x \mid x < 4\}$$

Solution

The complement of the set of all numbers that are less than 4 is the set of all numbers that are at least as big as 4. Notice that the number 4 is not in the set A, since the inequality is strict (does not have an "="). Therefore the number 4 is in the complement of the set A. In set notation:

$$A^c = \{x \mid x \geq 4\}$$

Example 21.4.2.2

When computing probabilities the complement is sometimes much easier than the original set. For example suppose you roll a die 6 times and want to find the probability that the number 3 comes up at least once. Find the complement of this event.

Solution

First note that the event of at least once means that there could be one 3, two 3's, three 3's, four 3's, five 3's, or six 3's. It turns out that this would be a burden to deal with each of these possibilities. However the complement is quite easy. The complement of getting at least one 3 is that you go no 3's.

Example 21.4.2.3

Suppose that we want to find the probability that at least 20 people in the class have done their homework. Find the complement of this event.

Solution

Sometimes it is easiest to list nearby outcomes and then determine the outcomes that satisfy the event. Finally, to find the complement, you select the rest. First list numbers near 20:

$$..., 17, 18, 19, 20, 21, 22, ...$$

Now, the ones that are at least 20 are all the ones including 20 and to the right of 20:

$$20, 21, 22, ...$$

These are the large numbers. The complement includes all the small numbers.

$$..., 17, 18, 19$$

We can write this in set notation as:

$$\{x \mid x \leq 19\}$$

or equivalently

$$\{x \mid x < 20\}$$

Example 21.4.2.4

Suppose a number is picked at random from the whole numbers from 1 to 10. Let A be the event that a number is both even and less than 8. Find the complement of A.

Solution

First, the set of numbers that are both even and less than 8 is:

$$A = \{2, 4, 6\}$$

The complement of this set is all the numbers from 1 to 10 that are not in A:

$$A^c = \{1, 3, 5, 7, 8, 9, 10\}$$

Exercise

Suppose that two six sided dice are rolled. Let the A be the event that either the first die is even or the sum of the dice is greater than 5 or both have occurred. Find the complement of A.

- Ex: Find the Intersection of a Set and A Complement Using a Venn Diagram
- https://youtu.be/ek3QwY2gw4w

This page titled 21.4.2: The Complement of a Set is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **The Complement of a Set** by Larry Green is licensed CC BY 4.0.

21.4.3: The Union and Intersection of Two Sets

Learning Outcomes

1. Find the union of two sets.
2. Find the intersection of two sets.
3. Combine unions intersections and complements.

All statistics classes include questions about probabilities involving the union and intersections of sets. In English, we use the words "Or", and "And" to describe these concepts. For example, "Find the probability that a student is taking a mathematics class or a science class." That is expressing the union of the two sets in words. "What is the probability that a nurse has a bachelor's degree and more than five years of experience working in a hospital." That is expressing the intersection of two sets. In this section we will learn how to decipher these types of sentences and will learn about the meaning of unions and intersections.

Unions

An element is in the union of two sets if it is in the first set, the second set, or both. The symbol we use for the union is ∪. The word that you will often see that indicates a union is "or".

Example 21.4.3.1: Union of Two sets

Let:

$$A = \{2, 5, 7, 8\}$$

and

$$B = \{1, 4, 5, 7, 9\}$$

Find $A \cup B$

Solution

We include in the union every number that is in A or is in B:

$$A \cup B = \{1, 2, 4, 5, 7, 8, 9\}$$

Example 21.4.3.2: Union of Two sets

Consider the following sentence, "Find the probability that a household has fewer than 6 windows or has a dozen windows." Write this in set notation as the union of two sets and then write out this union.

Solution

First, let A be the set of the number of windows that represents "fewer than 6 windows". This set includes all the numbers from 0 through 5:

$$A = \{0, 1, 2, 3, 4, 5\}$$

Next, let B be the set of the number of windows that represents "has a dozen windows". This is just the set that contains the single number 12:

$$B = \{12\}$$

We can now find the union of these two sets:

$$A \cup B = \{0, 1, 2, 3, 4, 5, 12\}$$

Intersections

An element is in the intersection of two sets if it is in the first set and it is in the second set. The symbol we use for the intersection is $\cap$. The word that you will often see that indicates an intersection is "and".

Example 21.4.3.3: Intersection of Two sets

Let:

$$A = \{3, 4, 5, 8, 9, 10, 11, 12\}$$

and

$$B = \{5, 6, 7, 8, 9\}$$

Find $A \cap B$.

Solution

We only include in the intersection that numbers that are in both A and B:

$$A \cap B = \{5, 8, 9\}$$

Example 21.4.3.4: Intersection of Two sets

Consider the following sentence, "Find the probability that the number of units that a student is taking is more than 12 units and less than 18 units." Assuming that students only take a whole number of units, write this in set notation as the intersection of two sets and then write out this intersection.

Solution

First, let A be the set of numbers of units that represents "more than 12 units". This set includes all the numbers starting at 13 and continuing forever:

$$A = \{13,\ 14,\ 15,\ ...\}$$

Next, let B be the set of the number of units that represents "less than 18 units". This is the set that contains the numbers from 1 through 17:

$$B = \{1,\ 2,\ 3,\ ...,\ 17\}$$

We can now find the intersection of these two sets:

$$A \cap B = \{13,\ 14,\ 15,\ 16,\ 17\}$$

Combining Unions, Intersections, and Complements

One of the biggest challenges in statistics is deciphering a sentence and turning it into symbols. This can be particularly difficult when there is a sentence that does not have the words "union", "intersection", or "complement", but it does implicitly refer to these words. The best way to become proficient in this skill is to practice, practice, and practice more.

Example 21.4.3.5

Consider the following sentence, "If you roll a six sided die, find the probability that it is not even and it is not a 3." Write this in set notation.

Solution

First, let A be the set of even numbers and B be the set that contains just 3. We can write:

$$A = \{2, 4, 6\},\ \ B = \{3\}$$

Next, since we want "not even" we need to consider the complement of A:

$$A^c = \{1, 3, 5\}$$

Similarly since we want "not a 3", we need to consider the complement of B:

$$B^c = \{1, 2, 4, 5, 6\}$$

Finally, we notice the key word "and". Thus, we are asked to find:

$$A^c \cap B^c = \{1, 3, 5\} \cap \{1, 2, 4, 5, 6\} = \{1, 5\}$$

Example 21.4.3.6

Consider the following sentence, "If you randomly select a person, find the probability that the person is older than 8 or is both younger than 6 and is not younger than 3." Write this in set notation.

Solution

First, let A be the set of people older than 8, B be the set of people younger than 6, and C be the set of people younger than 3. We can write:

$$A = \{x \mid x > 8\}, \quad B = \{x \mid x < 6\}, \, C = \{x \mid x < 3\}$$

We are asked to find

$$A \cup (B \cap C^c)$$

Notice that the complement of "<" is "≥". Thus:

$$C^c = \{x \mid x \geq 3\}$$

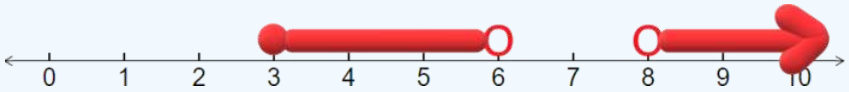
Next we find:

$$B \cap C^c = \{x \mid x < 6\} \cap \{x \mid x \geq 3\} = \{x \mid 3 \leq x < 6\}$$

Finally, we find:

$$A \cup (B \cap C^c) = \{x \mid x > 8\} \cup \{x \mid 3 \leq x < 6\}$$

The clearest way to display this union is on a number line. The number line below displays the answer:



Exercise

Suppose that we pick a person at random and are interested in finding the probability that the person's birth month came after July and did not come after September. Write this event using set notation.

- Ex: Find the Intersection of a Set and A Complement Using a Venn Diagram
- Intersection and Complements of Sets

This page titled 21.4.3: The Union and Intersection of Two Sets is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **The Union and Intersection of Two Sets** by Larry Green is licensed CC BY 4.0.

21.4.4: Venn Diagrams

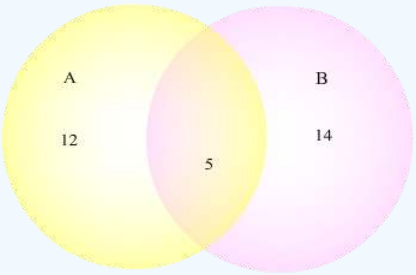
Learning Outcomes

1. Read a Venn Diagram to extract information.
2. Draw a Venn Diagram.

Venn Diagrams are a simple way of visualizing how sets interact. Many times we will see a long wordy sentence that describes a numerical situation, but it is a challenge to understand. As the saying goes, "A picture is worth a thousand words." In particular, a Venn Diagram describes how many elements are in each set displayed and how many elements are in their intersections and complements.

Example 21.4.4.1

Consider the Venn Diagram shown below.



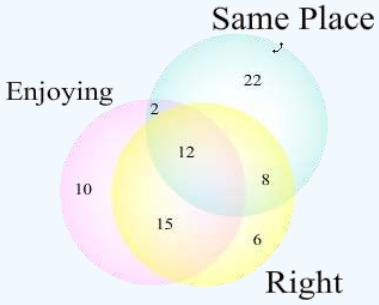
Describe how many elements are in each of the sets.

Solution

Once we understand how to read the Venn Diagram we can use it in many applications. For the Venn Diagram above, there are 12 from A that are not in B, there are 5 in both A and B, and there are 14 in B that are not in A. If we wanted to find the total in A, we would just add 12 and 5 to get 17 total in A. Similarly, there are 19 total in B.

Example 21.4.4.2

Consider the Venn Diagram below that shows the results of a study asking students whether their first college class was at the same place they are at now, whether they are right handed, and whether they are enjoying their experience at their college.



Determine how many students are:

1. Right handed and enjoy college.
2. At the same place but not right handed.
3. Enjoy college.

Solution

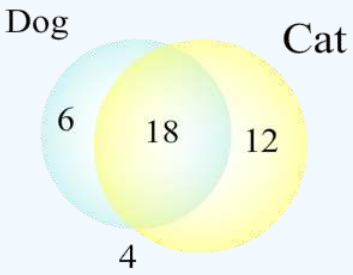
1. To be right handed and enjoy college they must be in both the Right circle and the Enjoying circle. Notice that the numbers 12 and 15 are in both these circles. Thus, there are 12 + 15 = 27 total students who are right handed and enjoy college.
2. To be in the same place and not be right handed, the number must be in the same place circle but not in the right circle. We see that 2 and 22 are the numbers in the same place circle but not in the right circle. Adding these gives 2 + 22 = 24 total students who are at the same place but not right handed.
3. We must count all the numbers in the Enjoying circle. These are 2, 10, 12, and 15. Adding these up gives: 2 + 10 + 12 + 15 = 39. Thus, 39 students enjoy college.

Example 21.4.4.3

Suppose that a group of 40 households was looked at. 24 of them housed dogs, 30 of them housed cats, and 18 of them housed both cats and dogs. Sketch a Venn Diagram that displays this information.

Solution

To get ready to sketch the Venn Diagram, we first plan on what it will look like. There are two main groups here: houses with dogs and houses with cats. Therefore we will have two circles. The intersection will have the number 18. Since there are 24 houses with dogs and 18 also have cats, we subtract 24 - 18 = 6 to find the houses with dogs but no cats. Similarly, we subtract 30 - 18 = 12 houses with cats and no dogs. If we add 18 + 6 + 12 = 36, we find the total number of houses with a dog, cat or both. Therefore there are 40 - 36 = 4 houses without any pets. Now we are ready to put in the numbers into the Venn Diagram. It is shown below.



Exercise

Suppose that a group of 55 businesses was researched. 29 of them were open on the weekends, 25 of them paid more than minimum wage for everyone , 17 of them were both open on the weekends and paid more than minimum wage for everyone, and 4 of them were government consulting businesses. None of the government consulting businesses were open on the weekend nor did they pay more than minimum wage for everyone. Sketch a Venn Diagram that displays this information.

- **Solving Problems with Venn Diagrams**
- https://youtu.be/t67RMAWGMdY

This page titled 21.4.4: Venn Diagrams is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Venn Diagrams** by Larry Green is licensed CC BY 4.0.

SECTION OVERVIEW

21.5: Expressions, Equations and Inequalities

21.5.1: Evaluate Algebraic Expressions

21.5.2: Inequalities and Midpoints

21.5.3: Solve Equations with Roots

21.5.4: Solving Linear Equations in One Variable

This page titled 21.5: Expressions, Equations and Inequalities is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

21.5.1: Evaluate Algebraic Expressions

Learning Outcomes

1. Evaluate an algebraic expression given values for the variables.
2. Recognize given values in a word problem and evaluate an expression using these values.

There are many formulas that are encountered in a statistics class and the values of each variable will be given. It will be your task to carefully evaluate the expression after plugging in each of the given values into the formula. In order to be successful you should not rush through the process and you need to be aware of the order of operations and use parentheses when necessary.

Example 21.5.1.1

Suppose that equation of the regression line for the number of days a week, x, a person exercises and the number of days, $\hat{y}$, a year a person is sick is:

$$\hat{y} = 12.5 - 1.6x$$

We use $\hat{y}$ instead of y since this is a prediction instead of an actual data value's y-coordinate. Use this regression line to predict the number of times a person who exercises 4 days a week will be sick this year.

Solution

The first step is always to identify the variable or variables that are given. In this case, we have 4 days of exercise a week, so:

$$x = 4$$

Next, we plug in to get:

$$\hat{y} = 12.5 - 1.6(4) = 6.1$$

Since we are predicting the number of days a year being sick, it is a good idea to round to the nearest whole number. We get that the best prediction for the number of sick days for a person who exercises 4 days per week is that they will be sick 6 days this year.

Example 21.5.1.2

For a yes/no question, a sample size is considered large enough to use a Normal distribution if

$np > 5$ and $nq > 5$

where n is the sample size, p is the proportion of Yes answers, and q is the proportion of No answers. A survey was given to 59 American adults asking them if they were food insecure today. 6.8% of them said they were food insecure today. Was the sample size large enough to use the Normal distribution?

Solution

Our first task is to list out each of the needed variables. Let's start with n, the sample size. We are given that 59 Americans were surveyed. Thus

$$n = 59$$

Next, we will find p, the proportion of Yes answers. We are given that 6.8% said Yes. Since this is a percent and not a proportion, we must convert the percent to a proportion by moving the decimal place two places to the right. It helps to place a 0 to the left of the 6, so that the decimal point has a place to go. A common error is to rush through this and wrongly write down 0.68. Instead, the proportion is:

$$p = 0.068$$

Our next task is to find q, the proportion of No answers. For a Yes/No question, the proportion of Yes answers and the proportion of No answers must always add up to 1. Thus:

$$q = 1 - 0.068 = 0.932$$

Now we are ready to plug into the two inequalities:

$$np = 59 \times 0.068 = 4.012$$

and

$$nq = 59 \times 0.932 = 54.988$$

Although $nq = 54.988 > 5$, we have $np = 4.012 < 5$, so the sample size was not large enough to use the Normal distribution.

Example 21.5.1.3

For a quantitative study, the sample size, n, needed in order to produce a confidence interval with a margin of error no more than $\pm E$, is

$$n = \left(\frac{z\sigma}{E}\right)^2$$

where z is a value that is determined from the confidence level and σ is the population standard deviation. You want to conduct a survey to estimate the population mean amount of years it takes psychologists to get through college and you require a margin of error of no more than ± 0.1 years. Suppose that you know that the population standard deviation is 1.3 years. If you want a 95% confidence interval that comes with a $z = 1.96$, at least how many psychologists must you survey? Round your answer up.

Solution

We start out by identifying the given values for each variable. Since we want a margin of error of no more than ± 0.1, we have:

$$E = 0.1$$

We are told that the population standard is 1.3, so:

$$\sigma = 1.3$$

We are also given the value of z:

$$z = 1.96$$

Now put this into the formula to get:

$$n = \left(\frac{1.96 \times 1.3}{0.1}\right)^2$$

We put this into a calculator or computer to get:

$$(1.96 \times 1.3 \div 0.1)^2 = 649.2304$$

We round up and can conclude that we need to survey 650 psychologists.

Example 21.5.1.4

Based on the Central Limit Theorem, the standard deviation of the sampling distribution when samples of size n are taken from a population with standard deviation, σ, is given by:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$$

If the population standard deviation for the number of customers who walk into a fast food restaurant is 12, what is the standard deviation of the sampling distribution for samples of size 35? Round your answer to two decimal places.

Solution

First we identify each of the given variables. Since the population standard deviation was 12, we have:

$$\sigma = 12$$

We are told that the sample size is 35, so:

$$n = 35$$

Now we put these numbers into the formula for the standard deviation of the sampling distribution to get:

$$\sigma_{\bar{x}} = \frac{12}{\sqrt{35}}$$

We are now ready to put this into our calculator or computer. We put in:

$$\sigma_{\bar{x}} = \frac{12}{\sqrt{35}} = 12 \div (35\text{^}0.5) = 2.02837$$

Rounded to two decimal places, we can say that the standard deviation of the sampling distribution is 2.03.

Example 21.5.1.5: Z score

The z-score for a given sample mean $\bar{x}$ for a sampling distribution with population mean μ, population standard deviation σ, and sample size n is given by:

$$z = \frac{\bar{x} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

An environmental scientist collected data on the amount of glacier retreat. She measured 45 glaciers. The population mean retreat is 22 meters and the population standard deviation is 16 meters. The sample mean for her data was 27 meters and the sample standard deviation for her data was 18 meters. What was the z-score?

Solution

First we identify each of the given variables. Since the sample mean was 27, we have:

$$\bar{x} = 27$$

We are told that the population mean is 22 meters, so:

$$\mu = 22$$

We are also given that the population standard deviation is 16 meters, hence:

$$\sigma = 16$$

Finally, since she measured 45 glaciers, we have:

$$n = 45$$

Now we put the numbers into the formula for the z-score to get:

$$z = \frac{27 - 22}{\frac{16}{\sqrt{45}}}$$

We are now ready to put this into our calculator or computer. We must pay attention to the order of operations and put parentheses around the numerator, since the subtraction happens for this expression before the division. We also must put parentheses around the denominator. We put in:

$$z = (27 - 22) \div \left(16 \div \sqrt{45}\right) = 2.0963$$

Exercise

You want to come up with a 90% confidence interval for the proportion of people in your community who are obese and require a margin of error of no more than $\pm 3\%$. According to the Journal of the American Medical Association (JAMA) 34% of all Americans are obese. The equation to find the sample size, n, needed in order to come up with a confidence interval is:

$$n = p(1-p)\left(\frac{z}{E}\right)^2$$

where p is the preliminary estimate for the population proportion. Based on calculations, $z = 1.645$. How many people in your community must you survey?

Evaluating Algebraic Expressions (L2.1)

https://youtu.be/HLjUT8Kvc5U

This page titled 21.5.1: Evaluate Algebraic Expressions is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Evaluate Algebraic Expressions** by Larry Green is licensed CC BY 4.0.

21.5.2: Inequalities and Midpoints

Learning Objectives

- Write out an inequality from words.
- Go from a midpoint and error to an inequality.
- Go from inequality to a midpoint and error.

Inequalities are an essential component of statistics. One very important use of inequalities is when we have found a mean or proportion from a sample and want to write out an inequality that gives where the population mean or proportion is likely to lie. Another application is in probability where we want to find the probability of a value being more than a number, less than a number, or between two numbers.

Converting Words to Inequalities

Example 21.5.2.1

You want to find the probability that it will a patient will "take at least three hours to wake up after surgery". Write an inequality for this situation.

Solution

The key words here are "at least". These words can be written symbolically as "≤≤". Therefore we can write "take at least three hours to wake up after surgery" as:

$$x \leq 3$$

Example 21.5.2.2

Suppose you want to find the probability that a relationship will last "more than 1 week and at most 8 weeks". Write an inequality for this situation.

Solution

Let's first translate the words "more than". This is equivalent to ">". Next translate the words "at most". This is equivalent to "≤". Now we can put this together to get:

$$1 < x \leq 8$$

Midpoints and Inequalities

There are two ways of thinking about an interval. The first is that x is greater than the lower bound and less than the upper bound. The second is that the center or midpoint of the interval is a given value and the interval goes no more than a certain distance from that value. In statistics, this is important when we look at confidence intervals. Both ways of presenting the interval are commonly used, so we need to be able to go from one way to the other.

Example 21.5.2.3

A researcher observed 45 startup companies to find a 95% confidence interval for the population mean amount of time it takes to make a profit. The sample mean was 14 months and the margin of error was plus or minus 8 months. In symbols the confidence interval can be written as:

$$14 \pm 8$$

Express this as a trilinear inequality.

Solution

We first find the lower bound by subtracting:

$$14 - 8 = 6$$

Next, we find the upper bound by adding:

$$14 + 8 = 22$$

We can now put this together as a trilinear inequality:

$$6 \leq x \leq 22$$

Example 21.5.2.4

A researcher interviewed 1000 Americans to asking them if they thought abortion should be against the law. The following 95% confidence interval was given for the population proportion of all Americans who are against abortion:

$$(0.41, 0.47)$$

Find the midpoint and the margin or error. That is write this interval in the form:

$$a \pm b \tag{21.5.2.1}$$

Solution

Let's first find the midpoint. This is the average of the left and right endpoints:

$$a = \frac{0.41 + 0.47}{2} = 0.44$$

Next, find the distance from the midpoint to either boundary:

$$b = 0.47 - 0.44 = 0.3$$

Finally we can put these two together to get:

$$0.44 \pm 0.03$$

Exercise 21.5.2.1

A study was done to see how many years longer it takes low income students to finish college compared to high income students. The confidence interval for the population mean difference was found to be:

$$[0.67, 0.84]$$

Find the midpoint and the margin of error. That is write this interval as in the form:

$$a \pm b$$

Converting an Inequality from Interval Notation to Midpoint and Error Notation (Links to an external site.)

Writing Equations and Inequalities for Scenarios

21.5.2: Inequalities and Midpoints is shared under a not declared license and was authored, remixed, and/or curated by LibreTexts.

- **Inequalities and Midpoints** has no license indicated.

21.5.3: Solve Equations with Roots

Learning Outcomes

- Solve equations that include square roots.

Square roots occur frequently in a statistics course, especially when dealing with standard deviations and sample sizes. In this section we will learn how to solve for a variable when that variable lies under the square root sign. The key thing to remember is that the square of a square root is what lies inside. In other words, squaring a square root cancels the square root.

Example 21.5.3.1

Solve the following equation for x.

$$2+\sqrt{x-3} = 6$$

Solution

What makes this a challenge is the square root. The strategy for solving is to isolate the square root on the left side of the equation and then square both sides. First subtract 2 from both sides:

$$\sqrt{x-3}=4$$

Now that the square root is isolated, we can square both sides of the equation:

$$\left(\sqrt{x-3}\right)^2=4^2$$

Since the square and the square root cancel we get:

$$x-3=16$$

Finally add 3 to both sides to arrive at:

$$x=19$$

It's always a good idea to check your work. We do this by plugging the answer back in and seeing if it works. We plug in $x = 19$ to get

$$\begin{aligned} 2+\sqrt{19-3} &= 2+\sqrt{16} \\ &= 2+4 \\ &= 6 \end{aligned}$$

Yes, the solution is correct.

Example 21.5.3.2

The standard deviation, $\sigma_{\hat{p}}$, of the sampling distribution for a proportion follows the formula:

$$\sigma_{\hat{p}}=\sqrt{\frac{p(1-p)}{n}}$$

Where p is the population proportion and n is the sample size. If the population proportion is 0.24 and you need the standard deviation of the sampling distribution to be 0.03, how large a sample do you need?

Solution

We are given that $p = 0.24$ and $\sigma_{\hat{p}} = 0.03$

Plug in to get:

$$0.03=\sqrt{\frac{0.24(1-0.24)}{n}}$$

We want to solve for n, so we want n on the left hand side of the equation. Just switch to get:

$$\sqrt{\frac{0.24\,(1-0.24)}{n}} = 0.03$$

Next, we subtract:

$$1 - 0.24 = 0.76$$

And them multiply:

$$0.24\,(0.76) = 0.1824$$

This gives us

$$\sqrt{\frac{0.1824}{n}} = 0.03$$

To get rid of the square root, square both sides:

$$\left(\sqrt{\frac{0.1824}{n}}\right)^2 = 0.03^2$$

The square cancels the square root, and squaring the right hand side gives:

$$\frac{0.1824}{n} = 0.0009$$

We can write:

$$\frac{0.1824}{n} = \frac{0.0009}{1}$$

Cross multiply to get:

$$0.0009\,n = 0.1824$$

Finally, divide both sides by 0.0009:

$$n = \frac{0.1824}{0.0009} = 202.66667$$

Round up and we can conclude that we need a sample size of 203 to get a standard error that is 0.03. We can check to see if this is reasonable by plugging $n = 203$ back into the equation. We use a calculator to get:

$$\sqrt{\frac{0.24\,(1-0.24)}{203}} = 0.029975$$

Since this is very close to 0.03, the answer is reasonable.

Exercise

The standard deviation, $\sigma_{\bar{x}}$, of the sampling distribution for a mean follows the formula:

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$$

Where σ is the population standard deviation and n is the sample size. If the population standard deviation is 3.8 and you need the standard deviation of the sampling distribution to be 0.5, how large a sample do you need?

- Ex 1: Solve a Basic Radical Equation - Square Roots
- https://youtu.be/u1aGMkJIlMI

This page titled 21.5.3: Solve Equations with Roots is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Solve Equations with Roots** by Larry Green is licensed CC BY 4.0.

21.5.4: Solving Linear Equations in One Variable

Learning Outcomes

- Solve linear equations for the variable.

It is a common task in algebra to solve an equation for a variable. The goal will be to get the variable on one side of the equation all by itself and have the other side of the equation just be a number. The process will involve identifying the operations that are done on the variable and apply the inverse operation to both sides of the equation. This will be managed in the reverse of the order of operations.

Example 21.5.4.1

Solve the following equation for x.

$$3x + 4 = 11 \tag{21.5.4.1}$$

Solution

We begin by looking at the operations that are done to x, keeping track the order. The first operation is "multiply by 3" and the second is "add 4". We now do everything backwards. Since the last operation is "add 4", our first step is to subtract 4 from both sides of Equation 21.5.4.1.

$$3x \cancel{+4} \cancel{-4} = 11 - 4$$

which simplifies the equation

$$3x = 7$$

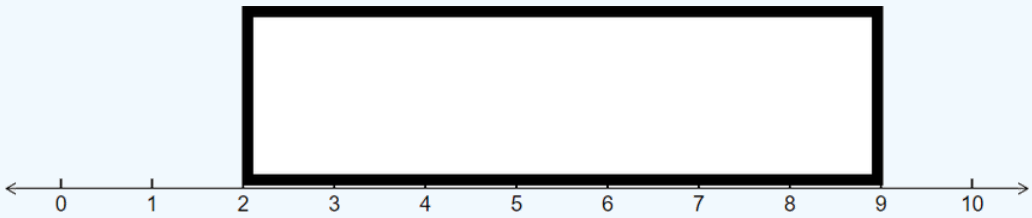
Next, the way to undo "multiply by 3" is to divide both sides by 3. We get

$$\frac{\cancel{3}x}{\cancel{3}} = \frac{7}{3}$$

or

$$x = \frac{7}{3}$$

Example 21.5.4.2



The rectangle above is a diagram for a uniform distribution from 2 to 9 that asks for the first quartile. The area of the smaller red rectangle that has base from 2 to Q1 and height 1/7 is 1/4. Find Q1.

Solution

We start by using the area formula for a rectangle:

$$\text{Area} = \text{Base} \times \text{Height} \tag{21.5.4.2}$$

We have:

- Area = $\frac{1}{4}$
- Base = $Q1 - 2$

- Height = $\frac{1}{7}$

Plug this into Equation 21.5.4.2 to get:

$$\frac{1}{4} = (Q1 - 2)\left(\frac{1}{7}\right) \tag{21.5.4.3}$$

We need to solve for $Q1$. First multiple both sides of Equation 21.5.4.3 by 7 to get:

$$7\left(\frac{1}{4}\right) = \cancel{7}\,(Q1 - 2)\,\cancel{\left(\frac{1}{7}\right)}$$

$$\frac{7}{4} = Q1 - 2 \tag{21.5.4.4}$$

Now add 2 to both sides of Equation 21.5.4.4 to get:

$$\frac{7}{4} + 2 = Q1 \cancel{-2} \cancel{+2}$$

$$\frac{7}{4} + 2 = Q1$$

or

$$Q1 = \frac{7}{4} + 2$$

Putting this into a calculator gives:

$$Q1 = 3.75$$

Example 21.5.4.3: z-score

The z-score for a given value x for a distribution with population mean μ and population standard deviation σ is given by:

$$z = \frac{x - \mu}{\sigma}$$

An online retailer has found that the population mean sales per day is \$2,841 and the population standard deviation is \$895. A value of x is considered an outlier if the z-score is less than -2 or greater than 2. How many sales must be made to have a z-score of 2?

Solution

First we identify each of the given variables. Since the population mean is 2,841, we have:

$$\mu = 2841$$

We are told that the population standard deviation is 895 meters, so:

$$\sigma = 895$$

We are also given that the z-score is 2, hence:

$$z = 2$$

Now we put the numbers into the formula for the z-score to get:

$$2 = \frac{x - 2841}{895}$$

We can next switch the order of the equation so that the x is on the left hand side of the equation:

$$\frac{x - 2841}{895} = 2$$

Next, we solve for x. First multiply both sides of the equation by 895 to get

$$x - 2841 = 2\,(895) = 1790$$

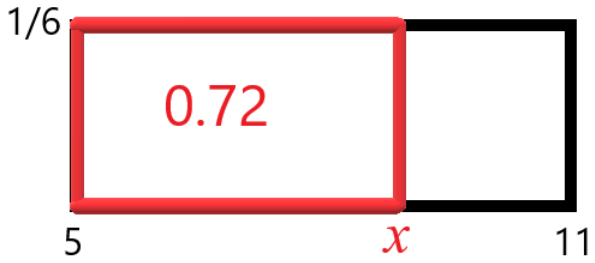
Finally, we can add 2841 to both sides of the equation to get x by itself:

$$x = 1790 + 2841 = 4631$$

We can conclude that if the day's sales is at $4631, the z-score is 2.

Exercise

The rectangle below is a diagram for a uniform distribution from 5 to 11 that asks for the 72nd percentile. The area of the smaller red rectangle that has base from 5 to the 72nd percentile, x, and height 1/6 is 0.72. Find x.



- Solving Two Step Equations: The Basics
- Solving Linear Equations

This page titled 21.5.4: Solving Linear Equations in One Variable is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Solving Linear Equations in One Variable** by Larry Green is licensed CC BY 4.0.

SECTION OVERVIEW

21.6: Graphing Points and Lines in Two Dimensions

21.6.1: Finding Residuals

21.6.2: Find the Equation of a Line given its Graph

21.6.3: Find y given x and the Equation of a Line

21.6.4: Graph a Line given its Equation

21.6.5: Interpreting the Slope of a Line

21.6.6: Interpreting the y-intercept of a Line

21.6.7: Plot an Ordered Pair

This page titled 21.6: Graphing Points and Lines in Two Dimensions is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

21.6.1: Finding Residuals

Learning Outcomes

- Given a Regression line and a data point, find the residual

In the linear regression part of statistics we are often asked to find the residuals. Given a data point and the regression line, the residual is defined by the vertical difference between the observed value of y and the computed value of $\hat{y}$ based on the equation of the regression line:

$$\text{Residual} = y - \hat{y}$$

Example 21.6.1.1

A study was conducted asking female college students how tall they are and how tall their mother is. The results are show in the table below:

Table of Mother and Daughter Heights

Mother's Height	63	67	64	60	65	67	59	60
Daughter's Height	58	64	65	61	65	67	61	64

The equation of the regression line is

$$\hat{y} = 30.28 + 0.52x$$

Find the residual for the mother who is 59 inches tall.

Solution

First note that the Daughter's Height associated with the mother who is 59 inches tall is 61 inches. This is y. Next we use the equation of the regression line to find $\hat{y}$. Since $x = 59$, we have

$$\hat{y} = 30.28 + 0.52(59)$$

We can use a calculator to get:

$$\hat{y} = 60.96$$

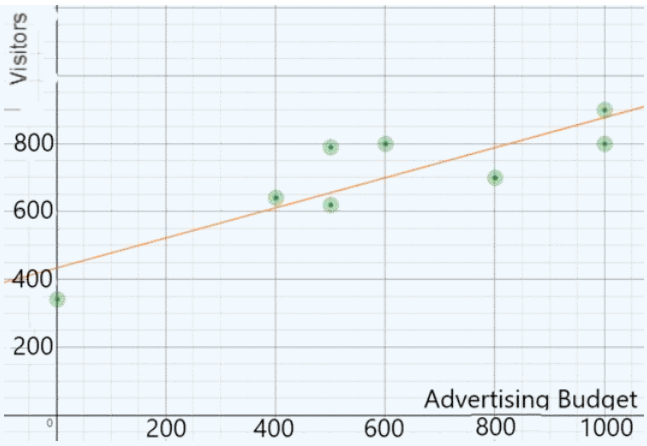
Now we are ready to put the values into the residual formula:

$$\text{Residual} = y - \hat{y} = 61 - 60.96 = 0.04$$

Therefore the residual for the 59 inch tall mother is 0.04. Since this residual is very close to 0, this means that the regression line was an accurate predictor of the daughter's height.

Example 21.6.1.2

An online retailer wanted to see how much bang for the buck was obtained from online advertising. The retailer experimented with different weekly advertising budgets and logged the number of visitors who came to the retailer's online site. The regression line for this is shown below.



Find the residual for the week when the retailer spent $600 on advertising.

Solution

First notice that the point of the scatterplot with x-coordinate of 600 has y-coordinate 800. Thus $y = 800$. Next note that the point on the line with x-coordinate 600 has y-coordinate 700. Thus $\hat{y} = 700$. Now we are ready to put the values into the residual formula:

$$\text{Residual} = y - \hat{y} = 800 - 700 = 100$$

Therefore the residual for the $600 advertising budget is -100.

Exercise

Data was taken from the recent Olympics on the GDP in trillions of dollars of 8 of the countries that competed and the number of gold medals that they won. The equation of the regression line is:

$$\hat{y} = 7.55 + 1.57x$$

The table below shows the data:

GDP	21	1.6	16	1.8	4	5.4	3.1	2.3
Medals	46	8	26	19	17	12	10	9

Find the residual for the country with a GDP of 4 trillion dollars.

- Calculating residual example | Exploring bivariate numerical data | AP Statistics | Khan Academy
- Finding a Residual

This page titled 21.6.1: Finding Residuals is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Finding Residuals** by Larry Green is licensed CC BY 4.0.

21.6.2: Find the Equation of a Line given its Graph

Learning Outcomes

1. Find the slope of a line given its graph.
2. Find the y-intercept of a line given its graph.
3. Find the equation of a line given its graph.

There are two main ways of representing a line: the first is with its graph, and the second is with its equation. In this section, we will practice how to find the equation of the line if we are given the graph of the line. The two key numbers in the equation of a line are the slope and the y-intercept. Thus the main steps in finding the equation of a line are finding the slope and finding the y-intercept. In statistics we are often presented with a **scatterplot** where we can eyeball the line. Once we have the graph of the line, getting the equation is helpful for making predictions based on the line.

Finding the Slope of a Line Given Its Graph

The steps to follow to fine the slope of the line given its graph are the following.

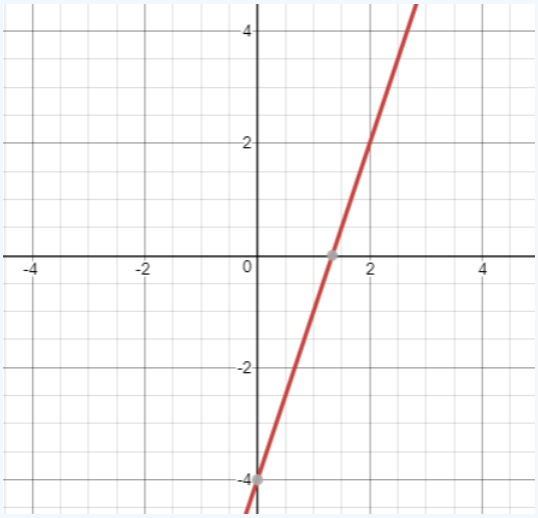
Step 1: Identify two points on the line. Any two points will do, but it is recommended to find points with nice x and y coordinates.

Step 2: The slope is the rise over the run. Thus if the points have coordinates (x_1, y_1) and (x_2, y_2), then the slope is:

$$Slope = \frac{Rise}{Run} = \frac{y_2 - y_1}{x_2 - x_1}$$

Example 21.6.2.1

Find the slope of the line shown below.



Solution

First, we locate points on the line that are as easy as possible to work with. The points with integer coordinates are (0,-4) and (2,2).

Next, we use the rise over run formula to find the slope of the line.

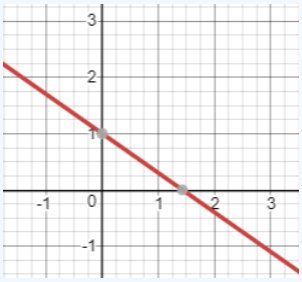
$$Slope = \frac{y_2 - y_1}{x_2 - x_1} = \frac{2 - (-4)}{2 - 0} = \frac{6}{2} = 3$$

Finding the y-intercept from the graph

If the portion of the graph that is in view includes the y-axis, then the y-intercept is very easy to spot. You just see where it crosses the y-axis. On the other hand, if the portion of the graph in view does not contain the y-axis, then it is best to first find the equation of the line and then use the equation to find the y-intercept.

Example 21.6.2.2

Find the y-intercept of the line shown below.



Solution

We just look at the line and notice that it crosses the y-axis at $y = 1$. Therefore, the y-intercept is 1 or (0,1).

Finding the equation of the line given its graph

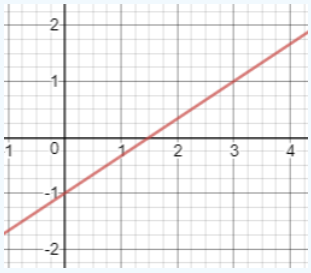
If you are given the graph of a line and want to find its equation, then you first find the slope as in Example **21.6.2.1**. Then you use one of the points you found (x_1, y_1) when you computed the slope, m, and put it into the **point slope equation**:

$$y - y_1 = m(x - x_1)$$

Then you multiply the slope through and add y_1 to both sides to get y by itself.

Example 21.6.2.3

Find the equation of the line shown below.



Solution

First we find the slope by identifying two nice points. Notice that the line passes through (0,-1) and (3,1). Now compute the slope using the rise over run formula:

$$Slope = \frac{rise}{run} = \frac{1-(-1)}{3-0} = \frac{2}{3}$$

Next use the point slope equation with the point (0,-1).

$$y - (-1) = \frac{2}{3}(x - 0)$$

Now simplify:

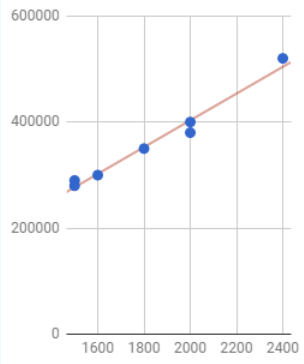
$$y + 1 = \frac{2}{3}x$$

Finally subtract 1 from both sides to get:

$$y = \frac{2}{3}x - 1$$

Example 21.6.2.4

A study was done to look at the relationship between the square footage of a house and the price of the house. The scatter plot and regression line are shown below. Find the equation of the regression line.



Solution

First we find the slope by identifying two nice points. You will have to eyeball it and notice that the line passes through (1600, 300000) and (2000,400000). Now compute the slope using the rise over run formula:

$$\frac{rise}{run} = \frac{400000 - 300000}{2000 - 1600} = \frac{100000}{400} = 250$$

Next use the point slope equation with the point (2000,400000).

$$y - (400000) = 250\,(x - 2000)$$

Now simplify:

$$y - 400000 = 250x - 500000$$

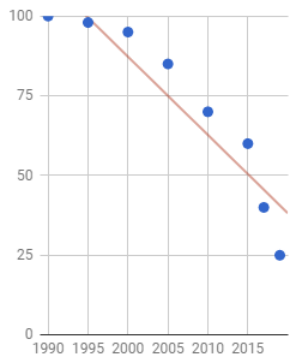
Finally add 400000 to both sides to get:

$$y = 250x - 100000$$

Notice that although the y-intercept is not visible from the graph of the line, we can see from the equation of the line that the y-intercept is -100000 or (0,-100000).

Exercise

The regression line and scatterplot below show the result of surveys that were taken in multiple years to find out the percent of households that had a landline telephone.



Find the equation of this regression line.

Ex 1: Find the Equation of a Line in Slope Intercept Form Given the Graph of a Line

Finding the Equation of a Line Given Its Graph

This page titled 21.6.2: Find the Equation of a Line given its Graph is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Find the Equation of a Line given its Graph** by Larry Green is licensed CC BY 4.0.

21.6.3: Find y given x and the Equation of a Line

Learning Outcomes

1. Find the value of y given x and the equation of a line.
2. Use a line to make predictions.

A line can be thought of as a function, which means that if a value of x is given, the equation of the line produces exactly one value of y; This is particularly useful in regression analysis where the line is used to make a prediction of one variable given the value of the other variable.

Example 21.6.3.1

Consider the line with equation:

$$y = 3x - 4$$

Find the value of y when x is 5.

Solution

Just replace the variable x with the number 5 in the equation and perform the arithmetic:

$$y = 3(5) - 4 = 15 - 4 = 11$$

Example 21.6.3.2

A survey was done to look at the relationship between a woman's height, x and the woman's weight, y. The equation of the regression line was found to be:

$$y = -220 + 5.5x$$

Use this equation to estimate the weight in pounds of a woman who is 5' 2" (62 inches) tall.

Solution

Just replace the variable x with the number 62 in the equation and perform the arithmetic:

$$y = -220 + 5.5(62)$$

We can put this into a calculator or computer to get:

$$y = 121$$

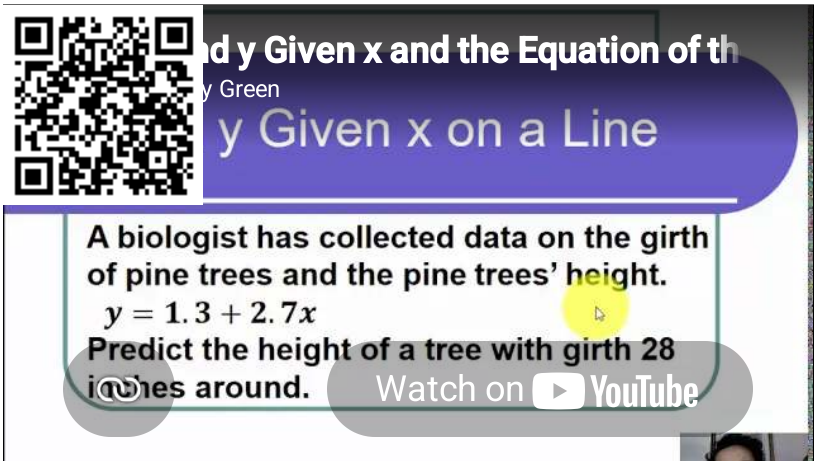
Therefore, our best prediction for the weight of a woman who is 5' 2" tall is that she is 121 lbs.

Exercise

A biologist has collected data on the girth (how far around) of pine trees and the pine tree's height. She found the equation of the regression line to be:

$$y = 1.3 + 2.7x$$

Where the girth, x, is measured in inches and the height, y, is measured in feet. Use the regression line to predict the height of a tree with girth 28 inches.



https://youtu.be/cS95PlUKZ6I

This page titled 21.6.3: Find y given x and the Equation of a Line is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Find y given x and the Equation of a Line** by Larry Green is licensed CC BY 4.0.

21.6.4: Graph a Line given its Equation

Learning Outcomes

1. Identify the slope and y-intercept from the equation of a line.
2. Plot the y-intercept of a line given its equation.
3. Plot a second point on a line given the y-intercept and the slope.
4. Graph a line given its equation in slope y-intercept form.

Often we are given an equation of a line and we want to visualize it. For this reason, it is important to be able to graph a line given its equation. We will look at lines that are in slope intercept form: $y = a + bx$ where a is the y-intercept of the line and b is the slope of the line. The y-intercept is the value of y where the line crosses the y-axis. The slope is the rise over run. If we write the slope as a fraction, then the numerator tells us how far to move up (or down if it is negative) and the denominator tells us how far to the right we need to go. the main application to statistics is in regression analysis which is the study of how to use a line to make a prediction about one variable based on the value of the other variable.

Example 21.6.4.1

Graph the line given by the equation:

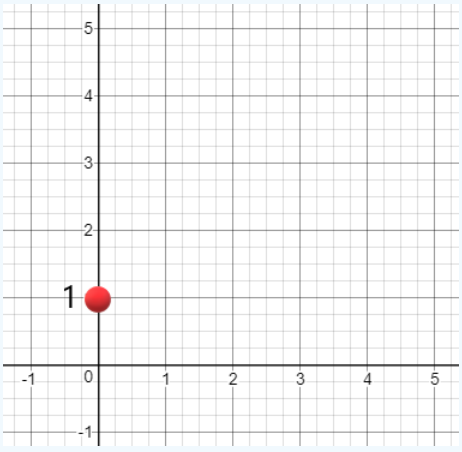
$$y = 1 + \frac{3}{2}x$$

Solution

We follow the three step process:

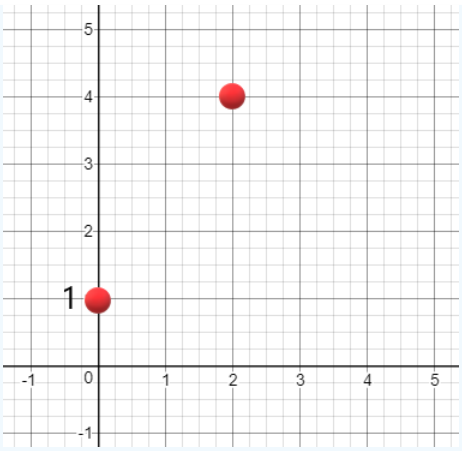
Step 1: Plot the y-intercept

The y-intercept is the number that is not associated with the x. For this example, it is 1. The x-coordinate of the y-intercept is always 0. So the coordinates of the y-intercept are (0,1). Thus start at the origin and move up 1:



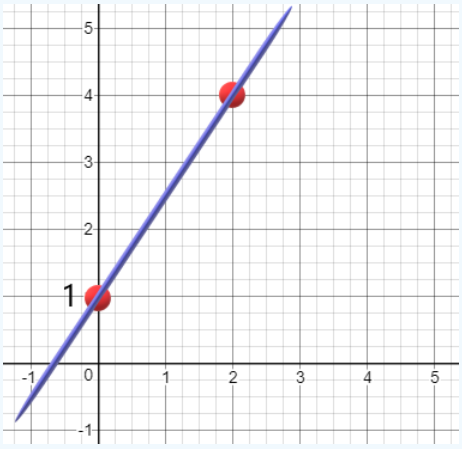
Step 2: Plot the Slope.

The slope of a line is the coefficient of the x term. Here it is $\frac{3}{2}$. What this means is that we rise 3 and run to the right 2. Rising 3 from an original y-coordinate of 1 gives a new y-coordinate of 4. Running 2 to the right from an initial x-coordinate of 0gives a new x-coordinate of 2. Thus we next plot the point (2,4).



Step 3: Connect the Dots

The last thing we need to do is connect the dots with a line:



Example 21.6.4.2

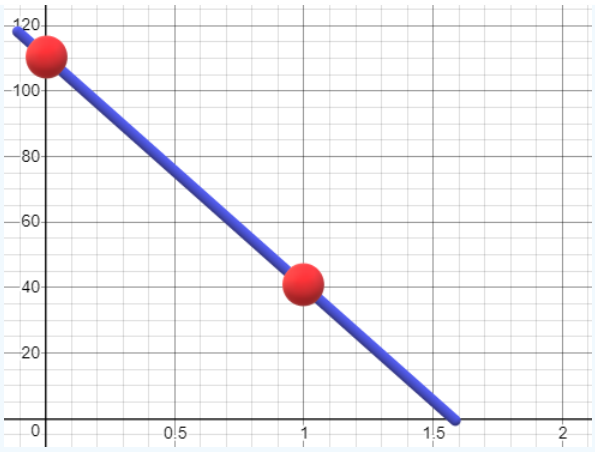
A study was done to look at the relationship between the weight of a car, x, in tons and its gas mileage in mpg, y. The equation of the regression line was found to be:

$$y = 110 - 70x \tag{21.6.4.1}$$

Graph this line.

Solution

The fist step is to note that the y-intercept is 110, hence the graph goes through the point (0,110). The next step is to see that the slope is -70. We can always put a number over 1 in order to make it a fraction. The slope of $-\frac{70}{1}$ tells us that y goes down by 70 if x goes up by 1. We use this to find the second point. The y-coordinate is: $110 - 70 = 40$. The x-coordinate is 1. Thus, a second point is (1,40). We can now plot the two points and connect the dots with a line.



Exercise

The regression line that relates the ounces of beer consumed just before a test, x, and the score on the test, y, is given by

$y = 93 - 1.2x$

Graph this line.

Graphing a Line in Slope-Intercept Form

https://youtu.be/z3rM-ZidXaw

This page titled 21.6.4: Graph a Line given its Equation is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Graph a Line given its Equation** by Larry Green is licensed CC BY 4.0.

21.6.5: Interpreting the Slope of a Line

Learning Outcomes

1. Interpret the slope of a line as the change in y when x changes by 1.

Template for Interpreting the Slope of a Line

For every increase in the x-variable by 1, the y-variable tends to change by (xxx the slope).

A common issue when we learn about the equation of a line in algebra is to state the slope as a number, but have no idea what it represents in the real world. The slope of a line is the rise over the run. If the slope is given by an integer or decimal value we can always put it over the number 1. In this case, the line rises by the slope when it runs 1. "Runs 1" means that the x value increases by 1 unit. Therefore the slope represents how much the y value changes when the x value changes by 1 unit. In statistics, especially regression analysis, the x value has real life meaning and so does the y value.

Example 21.6.5.1

A study was done to see the relationship between the time it takes, x, to complete a college degree and the student loan debt incurred, y. The equation of the regression line was found to be:

$$y = 25142 + 14329x \tag{21.6.5.1}$$

Interpret the slope of the regression line in the context of the study.

Solution

First, note that the slope is the coefficient in front of the x. Thus, the slope is 14,329. Next, the slope is the rise over the run, so it helps to write the slope as a fraction:

$$Slope = \frac{rise}{run} = \frac{14,329}{1} \tag{21.6.5.2}$$

The rise is the change in y and y represents student loan debt. Thus, the numerator represents an increase of $14,329 of student loan debt. The run is the change in x and x represents the time it takes to complete a college degree. Thus, the denominator represents an increase of 1 year to complete a college degree. We can put this all together and interpret the slope as telling us that

For every additional year it takes to complete a college degree, on average the student loan debt tends to increase by $14,329.

Example 21.6.5.2

Suppose that a research group tested the cholesterol level of a sample of 40 year old women and then waited many years to see the relationship between a woman's HDL cholesterol level in mg/dl, x, and her age of death, y. The equation of the regression line was found to be:

$$y = 103 - 0.3x \tag{21.6.5.3}$$

Interpret the slope of the regression line in the context of the study.

Solution

The slope of the regression line is -0.3. The slope as a fraction is:

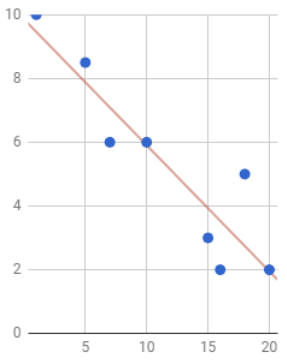
$Slope = \frac{rise}{run} = \frac{-0.3}{1}"width = "233$

The rise is the change in y and y represents age of death. Since the slope is negative, the numerator indicates a decrease in lifespan. Thus, the numerator represents a decrease in lifespan of 0.3 years. The run is the change in x and x represents the HDL cholesterol level. Thus, the denominator represents an HDL cholesterol level increase of 1 mg/dl. Now, put this all together and interpret the slope as telling us that

For every additional 1 mg/dl of HDL cholesterol, on average women are predicted to die 0.3 years younger.

Example 21.6.5.3

A researcher asked several employees who worked overtime "How many hours of overtime did you work last week?" and "On a scale from 1 to 10 how satisfied are you with your job?". The scatterplot and the regression line from this study are shown below.



Interpret the slope of the regression line in the context of the study.

Solution

We first need to determine the slope of the regression line. To find the slope, we get two points that have as nice coordinates as possible. From the graph, we see that the line goes through the points (10,6) and (15,4). The slope of the regression line can now be found using the rise over the run formula:

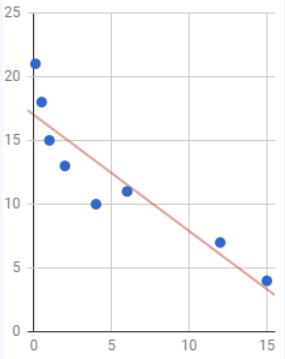
$$Slope = \frac{rise}{run} = \frac{4-6}{15-10} = \frac{-2}{5} \tag{21.6.5.4}$$

The rise is the change in y and y represents job satisfaction rating. Since the slope is negative, the numerator indicates a decrease in job satisfaction. Thus, the numerator represents a decrease in job satisfaction of 2 on the scale from 1 to 10. The run is the change in x and x represents the overtime work hours. Thus, the denominator represents an increase of 5 hours of overtime work. Now, put this all together and interpret the slope as telling us that

For every additional 5 hours of overtime work that employees are asked to do, their job satisfaction tends to go down an average of 2 points.

Exercise

The scatterplot and regression line below are from a study that collected data on the population (in hundred thousands) of cities and the average number of hours per week the city's residents spend outdoors.



Interpret the slope of this regression line in the context of the study.

This page titled 21.6.5: Interpreting the Slope of a Line is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Interpreting the Slope of a Line** by Larry Green is licensed CC BY 4.0.

21.6.6: Interpreting the y-intercept of a Line

Learning Outcomes

1. Interpret the y-intercept of a line as the value of y when x equals to 0.
2. Determine whether the y-intercept is useful for interpreting the relationship between x and y

Just like the slope of a line, many algebra classes go over the y-intercept of a line without explaining how to use it in the real world. The y-intercept of a line is the value of y where the line crosses the y-axis. In other words, it is the value of y when the value of x is equal to 0. Sometimes this has true meaning for the model that the line provides, but other times it is meaningless. We will encounter examples of both types in this section.

Template for the y-Intercept Interpretation

When the value for the x-variable is 0, the best prediction for the value of the y-variable is (xxx the y-intercept).

Example 21.6.6.1

A study was done to see the relationship between the ounces of meat, x, that people eat each day on average and the hours per week, y they watch sports. The equation of the regression line was found to be:

$$y = 1.3 + 0.4x$$

Interpret the y-intercept of the regression line in the context of the study or explain why it has no practical meaning.

Solution

First, note that the y-intercept is the number that is not in front of the x. Thus, the y-intercept is 1.3. Next, the y-intercept is the value of y when x equals zero. For this example, x represents the ounces of meat consumed each day.

When the consumption of meat is 0, the best prediction for the value of the hours of sports each week is 1.3.

If x is equal to 0, this means the person does not consume any meat. Since there are people, called vegetarians, who consume no meat, it is meaningful to have an x-value of 0. The y-value of 1.3 represents the hours of sports the person watches. Putting this all together we can state:

A vegetarian is predicted to watch 1.3 hours of sports each week.

Example 21.6.6.2

A neonatal nurse at Children's Hospital has collected data on the birth weight, x, in pounds the number of days, y, that the newborns stay in the hospital. The equation of the regression line was found to be

$$y = 45 - 3.9x$$

Interpret the y-intercept of the regression line in the context of the study or explain why it has no practical meaning.

Solution

Again, we note that the y-intercept is the number that is not in front of the x. Thus, the y-intercept is 45. Next, the y-intercept is the value of y when x equals zero.

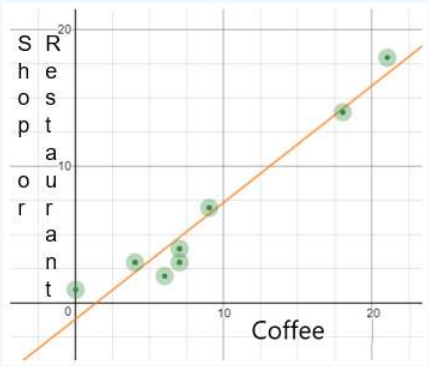
When the birth weight in pounds is 0, the best prediction for the value of the number of days the newborn is predicted to stay in the hospital is 45 days.

For this example, x represents the new born baby's birth weight in pounds. If x is equal to 0, this means the baby was born with a weight of 0 pounds. Since it makes no sense for a baby

to weigh 0 pounds, we can say that the y-intercept of this regression line has no practical meaning.

Example 21.6.6.3

A researcher asked several people "How many cups of coffee did you drink last week?" and "How many times did you go to a shop or restaurant for a meal or a drink last week?" The scatterplot and the regression line from this study are shown below.



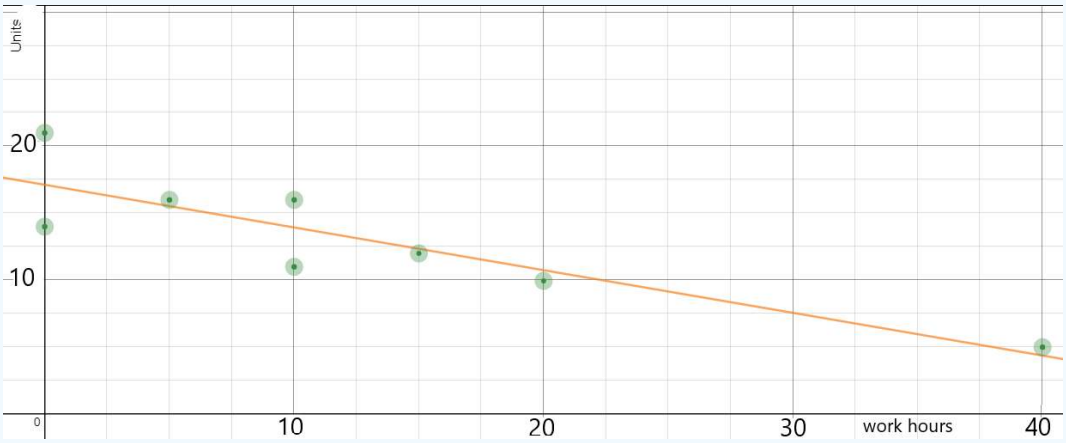
Interpret the y-intercept of the regression line in the context of the study or explain why it has no practical meaning.

Solution

The y-intercept of a line is where it crosses the y-axis. In this case, the line crosses at around y = -1. The value of x, by definition is 0 and the x-axis represents the number of cups of coffee a person drank last week. Since there are people who don't drink coffee, it does male sense to have an x-value of 0. The y-axis represents the number of times the person went to a shop or restaurant last week to purchase a meal or a drink. It makes no sense to say that a person went -1 times to a shop or restaurant last week to purchase a meal or a drink. Therefore the y-intercept of this regression line has no practical meaning.

Exercise

The scatterplot and regression line below are from a study that collected data from a group of college students on the number of hours per week during the school year they work at a paid job and the number of units they are taking. Interpret the y-intercept of the regression line or explain why it has no practical meaning.



- Interpret the Meaning of the y-intercept Given a Linear Equation
- Interpreting the y-Intercept

This page titled 21.6.6: Interpreting the y-intercept of a Line is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Interpreting the y-intercept of a Line** by Larry Green is licensed CC BY 4.0.

21.6.7: Plot an Ordered Pair

Learning Outcomes

1. Draw x and y axes.
2. Plot a point in the xy-plane

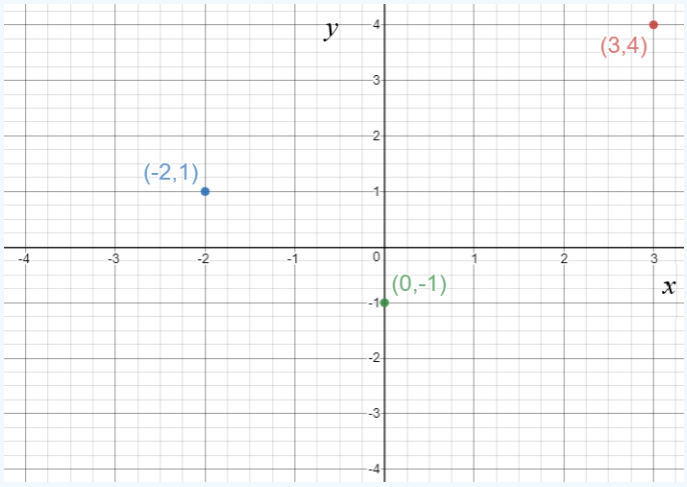
We have already gone into detail about how to plot points on a number line, and that is very useful for single variable presentations. Now we will move to questions that involve comparing two variables. Working with two variables is frequently encountered in statistical studies and we would like to be able to display the results graphically. This is best done by plotting points in the xy-plane.

Example 21.6.7.1

Plot the points: $(3,4)$,$(-2,1)$, and $(0,-1)$

Solution

The first thing to do when plotting points is to sketch the x-axis and y-axis and decide on the tick marks. Here the numbers are all less than 5, so it is reasonable to count by 1's. Next, we plot the first point, $(3,4)$. This means to start at the origin, where the axes intersect. Then move 3 units to the right and 4 units up. After arriving there, we just draw a dot. For the next point, $(-2,1)$, we start at the origin, move 2 units to the left and 1 unit up and draw the dot. For the third point, $(0,-1)$, we don't move left or right at all since the x-coordinate is 0, but we do move 1 unit down and draw the dot. The plot is shown below.



Example 21.6.7.2

A survey was done to look at the relationship between a person's age and their income. The first three answers are shown in the table below:

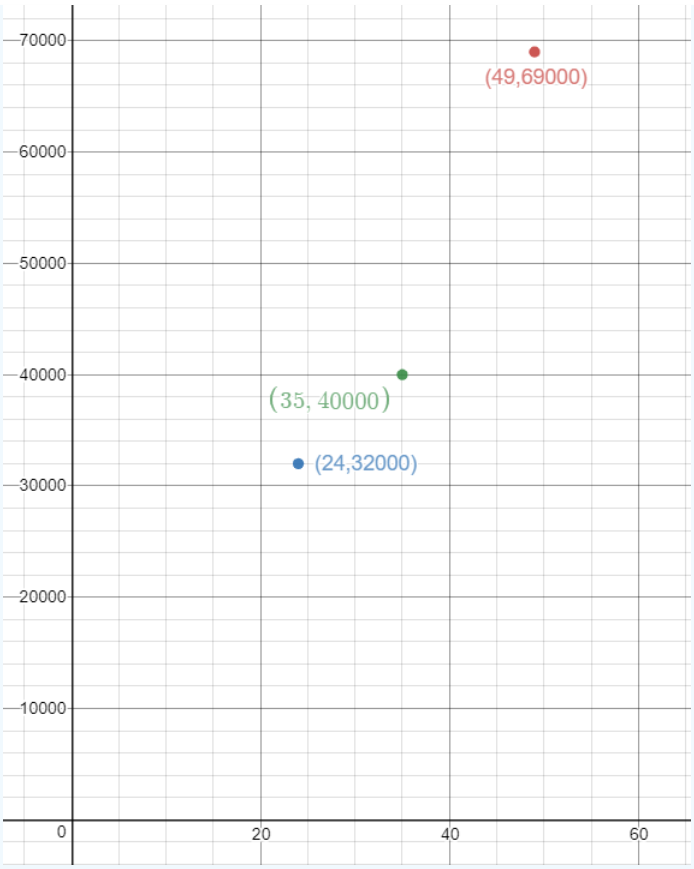
Table of ages and income

Age	49	24	35
Income	69,000	32,000	40,000

Graph the three points on the xy-plane.

Solution

Notice that the numbers are all relatively large. Therefore counting by 1's would not make sense. Instead, it makes better sense to count the Age axis, x, by 10's and the Income axis, y, by 1000's. The points are plotted below.



Exercise

A hotel manager was interested in seeing the relationship between the price per night, x, that the hotel charged and the number of occupied rooms, y. The results were (75,83), (100,60), (110,55), and (125,40). Plot these points in the xy-plane.

Ex: Plotting Points on the Coordinate Plane

Plotting Points

This page titled 21.6.7: Plot an Ordered Pair is shared under a CC BY 4.0 license and was authored, remixed, and/or curated by Larry Green.

- **Plot an Ordered Pair** by Larry Green is licensed CC BY 4.0.

CHAPTER OVERVIEW

Back Matter

Index

Glossary

Detailed Licensing

Index

A

Arithmetic

21.3.5: Perform Signed Number Arithmetic

B

bar graph

21.1.4: Using Fractions, Decimals and Percents to Describe Charts

C

Comparing numbers

21.1.1: Comparing Fractions, Decimals, and Percents

complement

21.4.2: The Complement of a Set

E

expected value

21.3.7: Using Summation Notation

F

Factorials

21.3.2: Factorials and Combination Notation

I

inequality

21.2.3: Represent an Inequality as an Interval on a Number Line

INTERSECTIONS

21.4.3: The Union and Intersection of Two Sets

M

midpoint

21.2.4: The Midpoint

N

Number Line

21.2.2: Plotting Points and Intervals on the Number Line

21.2.3: Represent an Inequality as an Interval on a Number Line

O

order of operations

21.3.3: Order of Operations

21.3.4: Order of Operations in Expressions and Formulas

P

PEMDAS

21.3.3: Order of Operations

pie chart

21.1.4: Using Fractions, Decimals and Percents to Describe Charts

powers

21.3.6: Powers and Roots

R

residuals

21.6.1: Finding Residuals

roots

21.3.6: Powers and Roots

rounding

21.1.3: Decimals- Rounding and Scientific Notation

S

set

21.4.1: Set Notation

set notation

21.4.1: Set Notation

square root

21.5.3: Solve Equations with Roots

summation notation

21.3.7: Using Summation Notation

U

unions

21.4.3: The Union and Intersection of Two Sets

V

Venn diagram

21.4.4: Venn Diagrams

Glossary

Sample Word 1 | Sample Definition 1

Detailed Licensing

Overview

Title: 21: Math Review for Introductory Statistics

Webpages: 46

All licenses found:

- CC BY 4.0: 78.3% (36 pages)
- Undeclared: 21.7% (10 pages)

By Page

- 21: Math Review for Introductory Statistics - *CC BY 4.0*
  - 00: Front Matter - *Undeclared*
    - TitlePage - *Undeclared*
    - InfoPage - *Undeclared*
    - Table of Contents - *Undeclared*
    - Licensing - *Undeclared*
  - 21.1: Decimals Fractions and Percents - *CC BY 4.0*
    - 21.1.1: Comparing Fractions, Decimals, and Percents - *CC BY 4.0*
    - 21.1.2: Converting Between Fractions, Decimals and Percents - *CC BY 4.0*
    - 21.1.3: Decimals- Rounding and Scientific Notation - *CC BY 4.0*
    - 21.1.4: Using Fractions, Decimals and Percents to Describe Charts - *CC BY 4.0*
  - 21.2: The Number Line - *CC BY 4.0*
    - 21.2.1: Distance between Two Points on a Number Line - *CC BY 4.0*
    - 21.2.2: Plotting Points and Intervals on the Number Line - *CC BY 4.0*
    - 21.2.3: Represent an Inequality as an Interval on a Number Line - *CC BY 4.0*
    - 21.2.4: The Midpoint - *CC BY 4.0*
  - 21.3: Operations on Numbers - *CC BY 4.0*
    - 21.3.1: Area of a Rectangle - *CC BY 4.0*
    - 21.3.2: Factorials and Combination Notation - *CC BY 4.0*
    - 21.3.3: Order of Operations - *CC BY 4.0*
    - 21.3.4: Order of Operations in Expressions and Formulas - *CC BY 4.0*
    - 21.3.5: Perform Signed Number Arithmetic - *CC BY 4.0*
    - 21.3.6: Powers and Roots - *CC BY 4.0*
    - 21.3.7: Using Summation Notation - *CC BY 4.0*
  - 21.4: Sets - *CC BY 4.0*
    - 21.4.1: Set Notation - *CC BY 4.0*
    - 21.4.2: The Complement of a Set - *CC BY 4.0*
    - 21.4.3: The Union and Intersection of Two Sets - *CC BY 4.0*
    - 21.4.4: Venn Diagrams - *CC BY 4.0*
  - 21.5: Expressions, Equations and Inequalities - *CC BY 4.0*
    - 21.5.1: Evaluate Algebraic Expressions - *CC BY 4.0*
    - 21.5.2: Inequalities and Midpoints - *Undeclared*
    - 21.5.3: Solve Equations with Roots - *CC BY 4.0*
    - 21.5.4: Solving Linear Equations in One Variable - *CC BY 4.0*
  - 21.6: Graphing Points and Lines in Two Dimensions - *CC BY 4.0*
    - 21.6.1: Finding Residuals - *CC BY 4.0*
    - 21.6.2: Find the Equation of a Line given its Graph - *CC BY 4.0*
    - 21.6.3: Find y given x and the Equation of a Line - *CC BY 4.0*
    - 21.6.4: Graph a Line given its Equation - *CC BY 4.0*
    - 21.6.5: Interpreting the Slope of a Line - *CC BY 4.0*
    - 21.6.6: Interpreting the y-intercept of a Line - *CC BY 4.0*
    - 21.6.7: Plot an Ordered Pair - *CC BY 4.0*
  - Back Matter - *Undeclared*
    - Index - *Undeclared*
    - Glossary - *Undeclared*
    - Detailed Licensing - *Undeclared*

Index

A

Arithmetic
- 21.3.5: Perform Signed Number Arithmetic

arithmetic mean
- 5.3: Measures of Central Tendency

B

bar graph
- 21.1.4: Using Fractions, Decimals and Percents to Describe Charts

bar graphs
- 3.1: Qualitative Data

C

causation
- 16.6: Causation

central tendency
- 5.1: Central Tendency
- 5.2: What is Central Tendency

cluster sample
- 1.2.6: Observational Studies and Sampling Strategies

Comparing numbers
- 21.1.1: Comparing Fractions, Decimals, and Percents

complement
- 21.4.2: The Complement of a Set

confidence interval
- 8.7: Statistical Literacy
- 8.E: Estimation (Exercises)

confounding variable
- 1.2.6: Observational Studies and Sampling Strategies
- 1.2.8: How Not to Do Statistics

Construct Validity
- 16.2: Measurement

correlation
- 13.2: Line Fitting, Residuals, and Correlation

E

expected value
- 21.3.7: Using Summation Notation

extrapolation
- 13.3: Fitting a Line by Least Squares Regression

F

Face Validity
- 16.2: Measurement

Factorials
- 21.3.2: Factorials and Combination Notation

frequency distribution
- 3.2: Quantitative Data

full model
- 14.2: Model Selection

G

generalized linear model
- 14.4: Introduction to Logistic Regression

geometric mean
- 5.6: Additional Measures

H

hidden bias
- 1.2.8: How Not to Do Statistics

histogram
- 3.2: Quantitative Data

I

inequality
- 21.2.3: Represent an Inequality as an Interval on a Number Line

influential point
- 13.4: Types of Outliers in Linear Regression

INTERSECTIONS
- 21.4.3: The Union and Intersection of Two Sets

K

kurtosis
- 5.10: Shapes of Distributions
- 6.3: Skew and Kurtosis

L

law of large numbers
- 8.2: The Law of Large Numbers

least squares criterion
- 13.3: Fitting a Line by Least Squares Regression

least squares line
- 13.3: Fitting a Line by Least Squares Regression

leptokurtic
- 6.3: Skew and Kurtosis

leverage
- 13.4: Types of Outliers in Linear Regression

Line Fitting
- 13.2: Line Fitting, Residuals, and Correlation

LINEAR REGRESSION MODEL
- 13: Introduction to Linear Regression

logistic regression
- 14: Multiple and Logistic Regression
- 14.4: Introduction to Logistic Regression

logit transformation
- 14.4: Introduction to Logistic Regression

lurking variable
- 1.2.8: How Not to Do Statistics

M

mean
- 5.3: Measures of Central Tendency
- 5.4: Median and Mean

median
- 5.3: Measures of Central Tendency
- 5.4: Median and Mean
- 5.5: Measures of the Location of the Data

midpoint
- 21.2.4: The Midpoint

Misconceptions
- 9.10: Misconceptions of Hypothesis Testing

mode
- 5.3: Measures of Central Tendency

Model Selection
- 14.2: Model Selection

multiple regression
- 14: Multiple and Logistic Regression
- 14.1: Introduction to Multiple Regression
- 14.3: Checking Model Assumptions using Graphs

N

natural splines
- 14.4: Introduction to Logistic Regression

Normal probability plot
- 14.3: Checking Model Assumptions using Graphs

Number Line
- 21.2.2: Plotting Points and Intervals on the Number Line
- 21.2.3: Represent an Inequality as an Interval on a Number Line

O

order of operations
- 21.3.3: Order of Operations
- 21.3.4: Order of Operations in Expressions and Formulas

outliers
- 5.5: Measures of the Location of the Data
- 13.4: Types of Outliers in Linear Regression

overgeneralization
- 1.2.8: How Not to Do Statistics

P

Pareto charts
- 3.1: Qualitative Data

Pearson's measure of skew
- 5.10: Shapes of Distributions

PEMDAS
- 21.3.3: Order of Operations

pie chart
- 21.1.4: Using Fractions, Decimals and Percents to Describe Charts

pie charts
- 3.1: Qualitative Data

placebo
- 1.2.5: Overview of Data Collection Principles

powers
- 21.3.6: Powers and Roots

Predictive Validity
- 16.2: Measurement

prospective study
- 1.2.6: Observational Studies and Sampling Strategies

Q

quartiles
- 5.5: Measures of the Location of the Data

R

range
- 5.9: Measures of Variability

reliability
- 16.2: Measurement

research design
- 16: Research Design

residuals
- 13.2: Line Fitting, Residuals, and Correlation
- 21.6.1: Finding Residuals

Retrospective studies
- 1.2.6: Observational Studies and Sampling Strategies

roots
- 21.3.6: Powers and Roots

rounding
21.1.3: Decimals- Rounding and Scientific Notation

S

sampling bias
16.4: Sampling Bias
scientific method
16.1: Scientific Method
set
21.4.1: Set Notation
set notation
21.4.1: Set Notation
simple random sampling
1.2.6: Observational Studies and Sampling Strategies
skew
5.10: Shapes of Distributions
6.3: Skew and Kurtosis
skewness
6.3: Skew and Kurtosis
square root
21.5.3: Solve Equations with Roots
standard error
16.2: Measurement
stratified sampling
1.2.6: Observational Studies and Sampling Strategies
summation notation
21.3.7: Using Summation Notation
Survivorship Bias
16.4: Sampling Bias

T

trimean
5.6: Additional Measures
trimmed mean
5.6: Additional Measures

U

Undercoverage Bias
16.4: Sampling Bias
unions
21.4.3: The Union and Intersection of Two Sets

V

validity
16.2: Measurement
variance
5.9: Measures of Variability
Variance Sum Law
5.12: Variance Sum Law I - Uncorrelated Variables
Venn diagram
21.4.4: Venn Diagrams

W

weapons effect
20.9: Weapons and Aggression

Glossary

Sample Word 1 | Sample Definition 1

Detailed Licensing

Overview

Title: Introduction to Statistics with R

Webpages: 310

Applicable Restrictions: Noncommercial

All licenses found:

- CC BY-SA 4.0: 44.5% (138 pages)
- Public Domain: 22.9% (71 pages)
- CC BY 4.0: 12.3% (38 pages)
- Undeclared: 7.7% (24 pages)
- CC BY-NC 2.0: 6.8% (21 pages)
- CC BY-SA 3.0: 5.8% (18 pages)

By Page

- Introduction to Statistics with R - *Undeclared*
  - Front Matter - *Undeclared*
    - Note to Students and Instructors - *Undeclared*
    - TitlePage - *Undeclared*
    - InfoPage - *Undeclared*
    - Table of Contents - *Undeclared*
    - Licensing - *Undeclared*
  - 1: Basics - *Undeclared*
    - 1.1: Introduction - *CC BY-NC 2.0*
      - 1.1.1: What Is Statistical Thinking? - *CC BY-NC 2.0*
      - 1.1.2: Dealing with Statistics Anxiety - *CC BY-NC 2.0*
      - 1.1.3: What Can Statistics Do for Us? - *CC BY-NC 2.0*
      - 1.1.4: The Big Ideas of Statistics - *CC BY-NC 2.0*
      - 1.1.5: Causality and Statistics - *CC BY-NC 2.0*
    - 1.2: Working with Data - *CC BY-NC 2.0*
      - 1.2.1: What Are Data? - *CC BY-NC 2.0*
      - 1.2.2: Data Basics - *CC BY-SA 3.0*
      - 1.2.3: Scales of Measurement - *CC BY-NC 2.0*
      - 1.2.4: What Makes a Good Measurement? - *CC BY-NC 2.0*
      - 1.2.5: Overview of Data Collection Principles - *CC BY-SA 3.0*
      - 1.2.6: Observational Studies and Sampling Strategies - *CC BY-SA 3.0*
      - 1.2.7: Experiments - *CC BY-SA 3.0*
      - 1.2.8: How Not to Do Statistics - *CC BY-SA 4.0*
      - 1.2.9: Exercises - *Public Domain*
  - 2: Introduction to R - *CC BY-NC 2.0*
    - 2.1: Why Programming Is Hard to Learn - *CC BY-NC 2.0*
    - 2.2: Using RStudio - *CC BY-NC 2.0*
    - 2.3: Installing R - *CC BY-SA 4.0*
    - 2.4: Getting Started with R - *CC BY-NC 2.0*
    - 2.5: Variables - *CC BY-NC 2.0*
    - 2.6: Functions - *CC BY-NC 2.0*
    - 2.7: Letting RStudio Help You with Your Commands - *CC BY-SA 4.0*
    - 2.8: Vectors - *CC BY-NC 2.0*
    - 2.9: Math with Vectors - *CC BY-NC 2.0*
    - 2.10: Data Frames - *CC BY-NC 2.0*
    - 2.11: Using R Libraries - *CC BY-NC 2.0*
    - 2.12: Installing and Loading Packages - *CC BY-SA 4.0*
    - 2.13: Using Comments - *CC BY-SA 4.0*
    - 2.14: Navigating the File System - *CC BY-SA 4.0*
    - 2.15: Loading and Saving Data - *CC BY-SA 4.0*
    - 2.16: Useful Things to Know about Variables - *CC BY-SA 4.0*
    - 2.17: Factors - *CC BY-SA 4.0*
    - 2.18: Data frames - *CC BY-SA 4.0*
    - 2.19: Suggested Readings and Videos - *CC BY-NC 2.0*
  - 3: Summarizing Data Visually - *Undeclared*
    - 3.1: Qualitative Data - *CC BY-SA 4.0*
    - 3.2: Quantitative Data - *CC BY-SA 4.0*
    - 3.3: Other Graphical Representations of Data - *CC BY-SA 4.0*
    - 3.4: Statistical Literacy - *Public Domain*
  - 4: Summarizing Data Visually Using R - *CC BY-SA 4.0*

- 4.1: An Overview of R Graphics - *CC BY-SA 4.0*
- 4.2: An Introduction to Plotting - *CC BY-SA 4.0*
- 4.3: Histograms - *CC BY-SA 4.0*
- 4.4: Stem and Leaf Plots - *CC BY-SA 4.0*
- 4.5: Scatterplots - *CC BY-SA 4.0*
- 4.6: Bar Graphs - *CC BY-SA 4.0*
- 4.7: Saving Image Files Using R and Rstudio - *CC BY-SA 4.0*
- 4.8: Summary - *CC BY-SA 4.0*

5: Summarizing Data With Numbers - *Public Domain*

- 5.1: Central Tendency - *Public Domain*
- 5.2: What is Central Tendency - *Public Domain*
- 5.3: Measures of Central Tendency - *Public Domain*
- 5.4: Median and Mean - *Public Domain*
- 5.5: Measures of the Location of the Data - *CC BY 4.0*
- 5.6: Additional Measures - *Public Domain*
- 5.7: Comparing Measures - *Public Domain*
- 5.8: Variability - *Public Domain*
- 5.9: Measures of Variability - *Public Domain*
- 5.10: Shapes of Distributions - *Public Domain*
- 5.11: Effects of Linear Transformations - *Public Domain*
- 5.12: Variance Sum Law I - Uncorrelated Variables - *Public Domain*
- 5.13: Statistical Literacy - *Public Domain*
- 5.14: Case Study- Using Stents to Prevent Strokes - *CC BY-SA 3.0*
- 5.15: Measures of the Location of the Data (Exercises) - *CC BY 4.0*
- 5.E: Summarizing Distributions (Exercises) - *Public Domain*

6: Describing Data With Numbers Using R - *CC BY-SA 4.0*

- 6.1: Measures of Central Tendency - *CC BY-SA 4.0*
- 6.2: Measures of Variability - *CC BY-SA 4.0*
- 6.3: Skew and Kurtosis - *CC BY-SA 4.0*
- 6.4: Getting an Overall Summary of a Variable - *CC BY-SA 4.0*
- 6.5: Descriptive Statistics Separately for each Group - *CC BY-SA 4.0*
- 6.6: Standard Scores - *CC BY-SA 4.0*
- 6.7: Epilogue- Good Descriptive Statistics Are Descriptive! - *CC BY-SA 4.0*

7: Introduction to Probability - *CC BY-SA 4.0*

- 7.1: How are Probability and Statistics Different? - *CC BY-SA 4.0*
- 7.2: What Does Probability Mean? - *CC BY-SA 4.0*
- 7.3: Basic Probability Theory - *CC BY-SA 4.0*
- 7.4: The Binomial Distribution - *CC BY-SA 4.0*
- 7.5: The Normal Distribution - *CC BY-SA 4.0*
- 7.6: Other Useful Distributions - *CC BY-SA 4.0*
- 7.7: Summary - *CC BY-SA 4.0*
- 7.8: Statistical Literacy - *Public Domain*
- 7.E: Probability (Exercises) - *Public Domain*

8: Estimating Unknown Quantities from a Sample - *CC BY-SA 4.0*

- 8.1: Samples, Populations and Sampling - *CC BY-SA 4.0*
- 8.2: The Law of Large Numbers - *CC BY-SA 4.0*
- 8.3: Sampling Distributions and the Central Limit Theorem - *CC BY-SA 4.0*
- 8.4: Estimating Population Parameters - *CC BY-SA 4.0*
- 8.5: Estimating a Confidence Interval - *CC BY-SA 4.0*
- 8.6: Summary - *CC BY-SA 4.0*
- 8.7: Statistical Literacy - *Public Domain*
- 8.E: Estimation (Exercises) - *Public Domain*

9: Hypothesis Testing - *CC BY-SA 4.0*

- 9.1: A Menagerie of Hypotheses - *CC BY-SA 4.0*
- 9.2: Two Types of Errors - *CC BY-SA 4.0*
- 9.3: Test Statistics and Sampling Distributions - *CC BY-SA 4.0*
- 9.4: Making Decisions - *CC BY-SA 4.0*
- 9.5: The p value of a test - *CC BY-SA 4.0*
- 9.6: Reporting the Results of a Hypothesis Test - *CC BY-SA 4.0*
- 9.7: Running the Hypothesis Test in Practice - *CC BY-SA 4.0*
- 9.8: Effect Size, Sample Size and Power - *CC BY-SA 4.0*
- 9.9: Some Issues to Consider - *CC BY-SA 4.0*
- 9.10: Misconceptions of Hypothesis Testing - *Public Domain*
- 9.11: Summary - *CC BY-SA 4.0*
- 9.12: Statistical Literacy - *Public Domain*
- 9.13: Logic of Hypothesis Testing (Exercises) - *Public Domain*

10: Categorical Data Analysis - *CC BY-SA 4.0*

- 10.1: The $\chi 2$ Goodness-of-fit Test - *CC BY-SA 4.0*
- 10.2: The $\chi 2$ test of independence (or association) - *CC BY-SA 4.0*
- 10.3: The Continuity Correction - *CC BY-SA 4.0*
- 10.4: Effect Size - *CC BY-SA 4.0*
- 10.5: Assumptions of the Test(s) - *CC BY-SA 4.0*
- 10.6: The Most Typical Way to Do Chi-square Tests in R - *CC BY-SA 4.0*
- 10.7: The Fisher Exact Test - *CC BY-SA 4.0*

- 10.8: The McNemar Test - *CC BY-SA 4.0*
- 10.9: What's the Difference Between McNemar and Independence? - *CC BY-SA 4.0*
- 10.10: Summary - *CC BY-SA 4.0*
- 10.11: Statistical Literacy - *Public Domain*
- 10.12: Chi Square (Exercises) - *Public Domain*

- 11: Comparing Two Means - *CC BY-SA 4.0*
  - 11.1: The one-sample z-test - *CC BY-SA 4.0*
  - 11.2: The One-sample t-test - *CC BY-SA 4.0*
  - 11.3: The Independent Samples t-test (Student Test) - *CC BY-SA 4.0*
  - 11.4: The Independent Samples t-test (Welch Test) - *CC BY-SA 4.0*
  - 11.5: The Paired-samples t-test - *CC BY-SA 4.0*
  - 11.6: One Sided Tests - *CC BY-SA 4.0*
  - 11.7: Using the t.test() Function - *CC BY-SA 4.0*
  - 11.8: Effect Size - *CC BY-SA 4.0*
  - 11.9: Checking the Normality of a Sample - *CC BY-SA 4.0*
  - 11.10: Testing Non-normal Data with Wilcoxon Tests - *CC BY-SA 4.0*
  - 11.11: Summary - *CC BY-SA 4.0*
  - 11.12: Statistical Literacy - *Public Domain*
  - 11.E: Tests of Means (Exercises) - *Public Domain*
- 12: Comparing Several Means (One-way ANOVA) - *CC BY-SA 4.0*
  - 12.1: Summary - *CC BY-SA 4.0*
  - 12.2: An Illustrative Data Set - *CC BY-SA 4.0*
  - 12.3: How ANOVA Works - *CC BY-SA 4.0*
  - 12.4: Running an ANOVA in R - *CC BY-SA 4.0*
  - 12.5: Effect Size - *CC BY-SA 4.0*
  - 12.6: Multiple Comparisons and Post Hoc Tests - *CC BY-SA 4.0*
  - 12.7: Assumptions of One-way ANOVA - *CC BY-SA 4.0*
  - 12.8: Checking the Homogeneity of Variance Assumption - *CC BY-SA 4.0*
  - 12.9: Removing the Homogeneity of Variance Assumption - *CC BY-SA 4.0*
  - 12.10: Checking the Normality Assumption - *CC BY-SA 4.0*
  - 12.11: Removing the Normality Assumption - *CC BY-SA 4.0*
  - 12.12: On the Relationship Between ANOVA and the Student t Test - *CC BY-SA 4.0*
- 13: Introduction to Linear Regression - *CC BY-SA 3.0*
  - 13.1: Prelude to Linear Regression - *CC BY-SA 3.0*
  - 13.2: Line Fitting, Residuals, and Correlation - *CC BY-SA 3.0*
  - 13.3: Fitting a Line by Least Squares Regression - *CC BY-SA 3.0*
  - 13.4: Types of Outliers in Linear Regression - *CC BY-SA 3.0*
  - 13.5: Inference for Linear Regression - *CC BY-SA 3.0*
  - 13.6: Exercises - *CC BY-SA 3.0*
- 14: Multiple and Logistic Regression - *CC BY-SA 3.0*
  - 14.1: Introduction to Multiple Regression - *CC BY-SA 3.0*
  - 14.2: Model Selection - *CC BY-SA 3.0*
  - 14.3: Checking Model Assumptions using Graphs - *CC BY-SA 3.0*
  - 14.4: Introduction to Logistic Regression - *CC BY-SA 3.0*
  - 14.5: Exercises - *CC BY-SA 3.0*
  - 14.6: Statistical Literacy - *Public Domain*
  - 14.E: Regression (Exercises) - *Public Domain*
- 15: Regression in R - *CC BY-SA 4.0*
  - 15.1: What Is a Linear Regression Model? - *CC BY-SA 4.0*
  - 15.2: Estimating a Linear Regression Model - *CC BY-SA 4.0*
  - 15.3: Multiple Linear Regression - *CC BY-SA 4.0*
  - 15.4: Quantifying the Fit of the Regression Model - *CC BY-SA 4.0*
  - 15.5: Hypothesis Tests for Regression Models - *CC BY-SA 4.0*
  - 15.6: Correlations - *CC BY-SA 4.0*
  - 15.7: Handling Missing Values - *CC BY-SA 4.0*
  - 15.8: Testing the Significance of a Correlation - *CC BY-SA 4.0*
  - 15.9: Regarding Regression Coefficients - *CC BY-SA 4.0*
  - 15.10: Assumptions of Regression - *CC BY-SA 4.0*
  - 15.11: Model Checking - *CC BY-SA 4.0*
  - 15.12: Model Selection - *CC BY-SA 4.0*
  - 15.13: Summary - *CC BY-SA 4.0*
- 16: Research Design - *Public Domain*
  - 16.1: Scientific Method - *Public Domain*
  - 16.2: Measurement - *Public Domain*
  - 16.3: Data Collection - *Public Domain*
  - 16.4: Sampling Bias - *Public Domain*
  - 16.5: Experimental Designs - *Public Domain*
  - 16.6: Causation - *Public Domain*
  - 16.7: Statistical Literacy - *Public Domain*
  - 16.E: Research Design (Exercises) - *Public Domain*
- 17: Preparing Datasets and Other Pragmatic Matters - *CC BY-SA 4.0*

- 17.1: Tabulating and Cross-tabulating Data - *CC BY-SA 4.0*
- 17.2: Transforming and Recoding a Variable - *CC BY-SA 4.0*
- 17.3: A few More Mathematical Functions and Operations - *CC BY-SA 4.0*
- 17.4: Extracting a Subset of a Vector - *CC BY-SA 4.0*
- 17.5: Extracting a Subset of a Data Frame - *CC BY-SA 4.0*
- 17.6: Sorting, Flipping and Merging Data - *CC BY-SA 4.0*
- 17.7: Reshaping a Data Frame - *CC BY-SA 4.0*
- 17.8: Working with Text - *CC BY-SA 4.0*
- 17.9: Reading Unusual Data Files - *CC BY-SA 4.0*
- 17.10: Coercing Data from One Class to Another - *CC BY-SA 4.0*
- 17.11: Other Useful Data Structures - *CC BY-SA 4.0*
- 17.12: Miscellaneous Topics - *CC BY-SA 4.0*
- 17.13: Summary - *CC BY-SA 4.0*

18: Basic Programming - *CC BY-SA 4.0*

- 18.1: Scripts - *CC BY-SA 4.0*
- 18.2: Loops - *CC BY-SA 4.0*
- 18.3: Conditional Statements - *CC BY-SA 4.0*
- 18.4: Writing Functions - *CC BY-SA 4.0*
- 18.5: Implicit Loops - *CC BY-SA 4.0*
- 18.6: Summary - *CC BY-SA 4.0*

19: Bayesian Statistics - *CC BY-SA 4.0*

- 19.1: Probabilistic Reasoning by Rational Agents - *CC BY-SA 4.0*
- 19.2: Bayesian Hypothesis Tests - *CC BY-SA 4.0*
- 19.3: Why Be a Bayesian? - *CC BY-SA 4.0*
- 19.4: Evidentiary Standards You Can Believe - *CC BY-SA 4.0*
- 19.5: The p-value Is a Lie. - *CC BY-SA 4.0*
- 19.6: Bayesian Analysis of Contingency Tables - *CC BY-SA 4.0*
- 19.7: Bayesian t-tests - *CC BY-SA 4.0*
- 19.8: Bayesian Regression - *CC BY-SA 4.0*
- 19.9: Bayesian ANOVA - *CC BY-SA 4.0*
- 19.10: Summary - *CC BY-SA 4.0*

20: Case Studies and Data - *Public Domain*

- 20.1: Angry Moods - *Public Domain*
- 20.2: Flatulence - *Public Domain*
- 20.3: Physicians Reactions - *Public Domain*
- 20.4: Teacher Ratings - *Public Domain*
- 20.5: Diet and Health - *Public Domain*
- 20.6: Smiles and Leniency - *Public Domain*
- 20.7: Animal Research - *Public Domain*
- 20.8: ADHD Treatment - *Public Domain*
- 20.9: Weapons and Aggression - *Public Domain*
- 20.10: SAT and College GPA - *Public Domain*
- 20.11: Stereograms - *Public Domain*
- 20.12: Driving - *Public Domain*
- 20.13: Stroop Interference - *Public Domain*
- 20.14: TV Violence - *Public Domain*
- 20.15: Obesity and Bias - *Public Domain*
- 20.16: Shaking and Stirring Martinis - *Public Domain*
- 20.17: Adolescent Lifestyle Choices - *Public Domain*
- 20.18: Chocolate and Body Weight - *Public Domain*
- 20.19: Bedroom TV and Hispanic Children - *Public Domain*
- 20.20: Weight and Sleep Apnea - *Public Domain*
- 20.21: Misusing SEM - *Public Domain*
- 20.22: School Gardens and Vegetable Consumption - *Public Domain*
- 20.23: TV and Hypertension - *Public Domain*
- 20.24: Dietary Supplements - *Public Domain*
- 20.25: Young People and Binge Drinking - *Public Domain*
- 20.26: Sugar Consumption in the US Diet - *Public Domain*
- 20.27: Nutrition Information Sources and Older Adults - *Public Domain*
- 20.28: Mind Set - Exercise and the Placebo Effect - *Public Domain*
- 20.29: Predicting Present and Future Affect - *Public Domain*
- 20.30: Exercise and Memory - *Public Domain*
- 20.31: Parental Recognition of Child Obesity - *Public Domain*
- 20.32: Educational Attainment and Racial, Ethnic, and Gender Disparity - *Public Domain*

21: Math Review for Introductory Statistics - *CC BY 4.0*

- 00: Front Matter - *Undeclared*
  - TitlePage - *Undeclared*
  - InfoPage - *Undeclared*
  - Table of Contents - *Undeclared*
  - Licensing - *Undeclared*
- 21.1: Decimals Fractions and Percents - *CC BY 4.0*
  - 21.1.1: Comparing Fractions, Decimals, and Percents - *CC BY 4.0*
  - 21.1.2: Converting Between Fractions, Decimals and Percents - *CC BY 4.0*
  - 21.1.3: Decimals- Rounding and Scientific Notation - *CC BY 4.0*
  - 21.1.4: Using Fractions, Decimals and Percents to Describe Charts - *CC BY 4.0*
- 21.2: The Number Line - *CC BY 4.0*

- 21.2.1: Distance between Two Points on a Number Line - *CC BY 4.0*
- 21.2.2: Plotting Points and Intervals on the Number Line - *CC BY 4.0*
- 21.2.3: Represent an Inequality as an Interval on a Number Line - *CC BY 4.0*
- 21.2.4: The Midpoint - *CC BY 4.0*

- 21.3: Operations on Numbers - *CC BY 4.0*
  - 21.3.1: Area of a Rectangle - *CC BY 4.0*
  - 21.3.2: Factorials and Combination Notation - *CC BY 4.0*
  - 21.3.3: Order of Operations - *CC BY 4.0*
  - 21.3.4: Order of Operations in Expressions and Formulas - *CC BY 4.0*
  - 21.3.5: Perform Signed Number Arithmetic - *CC BY 4.0*
  - 21.3.6: Powers and Roots - *CC BY 4.0*
  - 21.3.7: Using Summation Notation - *CC BY 4.0*
- 21.4: Sets - *CC BY 4.0*
  - 21.4.1: Set Notation - *CC BY 4.0*
  - 21.4.2: The Complement of a Set - *CC BY 4.0*
  - 21.4.3: The Union and Intersection of Two Sets - *CC BY 4.0*
  - 21.4.4: Venn Diagrams - *CC BY 4.0*
- 21.5: Expressions, Equations and Inequalities - *CC BY 4.0*
  - 21.5.1: Evaluate Algebraic Expressions - *CC BY 4.0*
  - 21.5.2: Inequalities and Midpoints - *Undeclared*
  - 21.5.3: Solve Equations with Roots - *CC BY 4.0*
  - 21.5.4: Solving Linear Equations in One Variable - *CC BY 4.0*
- 21.6: Graphing Points and Lines in Two Dimensions - *CC BY 4.0*
  - 21.6.1: Finding Residuals - *CC BY 4.0*
  - 21.6.2: Find the Equation of a Line given its Graph - *CC BY 4.0*
  - 21.6.3: Find y given x and the Equation of a Line - *CC BY 4.0*
  - 21.6.4: Graph a Line given its Equation - *CC BY 4.0*
  - 21.6.5: Interpreting the Slope of a Line - *CC BY 4.0*
  - 21.6.6: Interpreting the y-intercept of a Line - *CC BY 4.0*
  - 21.6.7: Plot an Ordered Pair - *CC BY 4.0*
- Back Matter - *Undeclared*
  - Index - *Undeclared*
  - Glossary - *Undeclared*
  - Detailed Licensing - *Undeclared*

- Back Matter - *Undeclared*
  - Index - *Undeclared*
  - Glossary - *Undeclared*
  - Detailed Licensing - *Undeclared*
  - Detailed Licensing - *Undeclared*

Detailed Licensing

Overview

Title: Introduction to Statistics with R

Webpages: 310

Applicable Restrictions: Noncommercial

All licenses found:

- CC BY-SA 4.0: 44.5% (138 pages)
- Public Domain: 22.9% (71 pages)
- CC BY 4.0: 12.3% (38 pages)
- Undeclared: 7.7% (24 pages)
- CC BY-NC 2.0: 6.8% (21 pages)
- CC BY-SA 3.0: 5.8% (18 pages)

By Page

- Introduction to Statistics with R - *Undeclared*
  - Front Matter - *Undeclared*
    - Note to Students and Instructors - *Undeclared*
    - TitlePage - *Undeclared*
    - InfoPage - *Undeclared*
    - Table of Contents - *Undeclared*
    - Licensing - *Undeclared*
  - 1: Basics - *Undeclared*
    - 1.1: Introduction - *CC BY-NC 2.0*
      - 1.1.1: What Is Statistical Thinking? - *CC BY-NC 2.0*
      - 1.1.2: Dealing with Statistics Anxiety - *CC BY-NC 2.0*
      - 1.1.3: What Can Statistics Do for Us? - *CC BY-NC 2.0*
      - 1.1.4: The Big Ideas of Statistics - *CC BY-NC 2.0*
      - 1.1.5: Causality and Statistics - *CC BY-NC 2.0*
    - 1.2: Working with Data - *CC BY-NC 2.0*
      - 1.2.1: What Are Data? - *CC BY-NC 2.0*
      - 1.2.2: Data Basics - *CC BY-SA 3.0*
      - 1.2.3: Scales of Measurement - *CC BY-NC 2.0*
      - 1.2.4: What Makes a Good Measurement? - *CC BY-NC 2.0*
      - 1.2.5: Overview of Data Collection Principles - *CC BY-SA 3.0*
      - 1.2.6: Observational Studies and Sampling Strategies - *CC BY-SA 3.0*
      - 1.2.7: Experiments - *CC BY-SA 3.0*
      - 1.2.8: How Not to Do Statistics - *CC BY-SA 4.0*
      - 1.2.9: Exercises - *Public Domain*
  - 2: Introduction to R - *CC BY-NC 2.0*
    - 2.1: Why Programming Is Hard to Learn - *CC BY-NC 2.0*
    - 2.2: Using RStudio - *CC BY-NC 2.0*
    - 2.3: Installing R - *CC BY-SA 4.0*
    - 2.4: Getting Started with R - *CC BY-NC 2.0*
    - 2.5: Variables - *CC BY-NC 2.0*
    - 2.6: Functions - *CC BY-NC 2.0*
    - 2.7: Letting RStudio Help You with Your Commands - *CC BY-SA 4.0*
    - 2.8: Vectors - *CC BY-NC 2.0*
    - 2.9: Math with Vectors - *CC BY-NC 2.0*
    - 2.10: Data Frames - *CC BY-NC 2.0*
    - 2.11: Using R Libraries - *CC BY-NC 2.0*
    - 2.12: Installing and Loading Packages - *CC BY-SA 4.0*
    - 2.13: Using Comments - *CC BY-SA 4.0*
    - 2.14: Navigating the File System - *CC BY-SA 4.0*
    - 2.15: Loading and Saving Data - *CC BY-SA 4.0*
    - 2.16: Useful Things to Know about Variables - *CC BY-SA 4.0*
    - 2.17: Factors - *CC BY-SA 4.0*
    - 2.18: Data frames - *CC BY-SA 4.0*
    - 2.19: Suggested Readings and Videos - *CC BY-NC 2.0*
  - 3: Summarizing Data Visually - *Undeclared*
    - 3.1: Qualitative Data - *CC BY-SA 4.0*
    - 3.2: Quantitative Data - *CC BY-SA 4.0*
    - 3.3: Other Graphical Representations of Data - *CC BY-SA 4.0*
    - 3.4: Statistical Literacy - *Public Domain*
  - 4: Summarizing Data Visually Using R - *CC BY-SA 4.0*

- 4.1: An Overview of R Graphics - *CC BY-SA 4.0*
- 4.2: An Introduction to Plotting - *CC BY-SA 4.0*
- 4.3: Histograms - *CC BY-SA 4.0*
- 4.4: Stem and Leaf Plots - *CC BY-SA 4.0*
- 4.5: Scatterplots - *CC BY-SA 4.0*
- 4.6: Bar Graphs - *CC BY-SA 4.0*
- 4.7: Saving Image Files Using R and Rstudio - *CC BY-SA 4.0*
- 4.8: Summary - *CC BY-SA 4.0*

5: Summarizing Data With Numbers - *Public Domain*

- 5.1: Central Tendency - *Public Domain*
- 5.2: What is Central Tendency - *Public Domain*
- 5.3: Measures of Central Tendency - *Public Domain*
- 5.4: Median and Mean - *Public Domain*
- 5.5: Measures of the Location of the Data - *CC BY 4.0*
- 5.6: Additional Measures - *Public Domain*
- 5.7: Comparing Measures - *Public Domain*
- 5.8: Variability - *Public Domain*
- 5.9: Measures of Variability - *Public Domain*
- 5.10: Shapes of Distributions - *Public Domain*
- 5.11: Effects of Linear Transformations - *Public Domain*
- 5.12: Variance Sum Law I - Uncorrelated Variables - *Public Domain*
- 5.13: Statistical Literacy - *Public Domain*
- 5.14: Case Study- Using Stents to Prevent Strokes - *CC BY-SA 3.0*
- 5.15: Measures of the Location of the Data (Exercises) - *CC BY 4.0*
- 5.E: Summarizing Distributions (Exercises) - *Public Domain*

6: Describing Data With Numbers Using R - *CC BY-SA 4.0*

- 6.1: Measures of Central Tendency - *CC BY-SA 4.0*
- 6.2: Measures of Variability - *CC BY-SA 4.0*
- 6.3: Skew and Kurtosis - *CC BY-SA 4.0*
- 6.4: Getting an Overall Summary of a Variable - *CC BY-SA 4.0*
- 6.5: Descriptive Statistics Separately for each Group - *CC BY-SA 4.0*
- 6.6: Standard Scores - *CC BY-SA 4.0*
- 6.7: Epilogue- Good Descriptive Statistics Are Descriptive! - *CC BY-SA 4.0*

7: Introduction to Probability - *CC BY-SA 4.0*

- 7.1: How are Probability and Statistics Different? - *CC BY-SA 4.0*
- 7.2: What Does Probability Mean? - *CC BY-SA 4.0*
- 7.3: Basic Probability Theory - *CC BY-SA 4.0*
- 7.4: The Binomial Distribution - *CC BY-SA 4.0*
- 7.5: The Normal Distribution - *CC BY-SA 4.0*
- 7.6: Other Useful Distributions - *CC BY-SA 4.0*
- 7.7: Summary - *CC BY-SA 4.0*
- 7.8: Statistical Literacy - *Public Domain*
- 7.E: Probability (Exercises) - *Public Domain*

8: Estimating Unknown Quantities from a Sample - *CC BY-SA 4.0*

- 8.1: Samples, Populations and Sampling - *CC BY-SA 4.0*
- 8.2: The Law of Large Numbers - *CC BY-SA 4.0*
- 8.3: Sampling Distributions and the Central Limit Theorem - *CC BY-SA 4.0*
- 8.4: Estimating Population Parameters - *CC BY-SA 4.0*
- 8.5: Estimating a Confidence Interval - *CC BY-SA 4.0*
- 8.6: Summary - *CC BY-SA 4.0*
- 8.7: Statistical Literacy - *Public Domain*
- 8.E: Estimation (Exercises) - *Public Domain*

9: Hypothesis Testing - *CC BY-SA 4.0*

- 9.1: A Menagerie of Hypotheses - *CC BY-SA 4.0*
- 9.2: Two Types of Errors - *CC BY-SA 4.0*
- 9.3: Test Statistics and Sampling Distributions - *CC BY-SA 4.0*
- 9.4: Making Decisions - *CC BY-SA 4.0*
- 9.5: The p value of a test - *CC BY-SA 4.0*
- 9.6: Reporting the Results of a Hypothesis Test - *CC BY-SA 4.0*
- 9.7: Running the Hypothesis Test in Practice - *CC BY-SA 4.0*
- 9.8: Effect Size, Sample Size and Power - *CC BY-SA 4.0*
- 9.9: Some Issues to Consider - *CC BY-SA 4.0*
- 9.10: Misconceptions of Hypothesis Testing - *Public Domain*
- 9.11: Summary - *CC BY-SA 4.0*
- 9.12: Statistical Literacy - *Public Domain*
- 9.13: Logic of Hypothesis Testing (Exercises) - *Public Domain*

10: Categorical Data Analysis - *CC BY-SA 4.0*

- 10.1: The $\chi 2$ Goodness-of-fit Test - *CC BY-SA 4.0*
- 10.2: The $\chi 2$ test of independence (or association) - *CC BY-SA 4.0*
- 10.3: The Continuity Correction - *CC BY-SA 4.0*
- 10.4: Effect Size - *CC BY-SA 4.0*
- 10.5: Assumptions of the Test(s) - *CC BY-SA 4.0*
- 10.6: The Most Typical Way to Do Chi-square Tests in R - *CC BY-SA 4.0*
- 10.7: The Fisher Exact Test - *CC BY-SA 4.0*

- 10.8: The McNemar Test - *CC BY-SA 4.0*
- 10.9: What's the Difference Between McNemar and Independence? - *CC BY-SA 4.0*
- 10.10: Summary - *CC BY-SA 4.0*
- 10.11: Statistical Literacy - *Public Domain*
- 10.12: Chi Square (Exercises) - *Public Domain*

- 11: Comparing Two Means - *CC BY-SA 4.0*
  - 11.1: The one-sample z-test - *CC BY-SA 4.0*
  - 11.2: The One-sample t-test - *CC BY-SA 4.0*
  - 11.3: The Independent Samples t-test (Student Test) - *CC BY-SA 4.0*
  - 11.4: The Independent Samples t-test (Welch Test) - *CC BY-SA 4.0*
  - 11.5: The Paired-samples t-test - *CC BY-SA 4.0*
  - 11.6: One Sided Tests - *CC BY-SA 4.0*
  - 11.7: Using the t.test() Function - *CC BY-SA 4.0*
  - 11.8: Effect Size - *CC BY-SA 4.0*
  - 11.9: Checking the Normality of a Sample - *CC BY-SA 4.0*
  - 11.10: Testing Non-normal Data with Wilcoxon Tests - *CC BY-SA 4.0*
  - 11.11: Summary - *CC BY-SA 4.0*
  - 11.12: Statistical Literacy - *Public Domain*
  - 11.E: Tests of Means (Exercises) - *Public Domain*
- 12: Comparing Several Means (One-way ANOVA) - *CC BY-SA 4.0*
  - 12.1: Summary - *CC BY-SA 4.0*
  - 12.2: An Illustrative Data Set - *CC BY-SA 4.0*
  - 12.3: How ANOVA Works - *CC BY-SA 4.0*
  - 12.4: Running an ANOVA in R - *CC BY-SA 4.0*
  - 12.5: Effect Size - *CC BY-SA 4.0*
  - 12.6: Multiple Comparisons and Post Hoc Tests - *CC BY-SA 4.0*
  - 12.7: Assumptions of One-way ANOVA - *CC BY-SA 4.0*
  - 12.8: Checking the Homogeneity of Variance Assumption - *CC BY-SA 4.0*
  - 12.9: Removing the Homogeneity of Variance Assumption - *CC BY-SA 4.0*
  - 12.10: Checking the Normality Assumption - *CC BY-SA 4.0*
  - 12.11: Removing the Normality Assumption - *CC BY-SA 4.0*
  - 12.12: On the Relationship Between ANOVA and the Student t Test - *CC BY-SA 4.0*
- 13: Introduction to Linear Regression - *CC BY-SA 3.0*
  - 13.1: Prelude to Linear Regression - *CC BY-SA 3.0*
  - 13.2: Line Fitting, Residuals, and Correlation - *CC BY-SA 3.0*
  - 13.3: Fitting a Line by Least Squares Regression - *CC BY-SA 3.0*
  - 13.4: Types of Outliers in Linear Regression - *CC BY-SA 3.0*
  - 13.5: Inference for Linear Regression - *CC BY-SA 3.0*
  - 13.6: Exercises - *CC BY-SA 3.0*
- 14: Multiple and Logistic Regression - *CC BY-SA 3.0*
  - 14.1: Introduction to Multiple Regression - *CC BY-SA 3.0*
  - 14.2: Model Selection - *CC BY-SA 3.0*
  - 14.3: Checking Model Assumptions using Graphs - *CC BY-SA 3.0*
  - 14.4: Introduction to Logistic Regression - *CC BY-SA 3.0*
  - 14.5: Exercises - *CC BY-SA 3.0*
  - 14.6: Statistical Literacy - *Public Domain*
  - 14.E: Regression (Exercises) - *Public Domain*
- 15: Regression in R - *CC BY-SA 4.0*
  - 15.1: What Is a Linear Regression Model? - *CC BY-SA 4.0*
  - 15.2: Estimating a Linear Regression Model - *CC BY-SA 4.0*
  - 15.3: Multiple Linear Regression - *CC BY-SA 4.0*
  - 15.4: Quantifying the Fit of the Regression Model - *CC BY-SA 4.0*
  - 15.5: Hypothesis Tests for Regression Models - *CC BY-SA 4.0*
  - 15.6: Correlations - *CC BY-SA 4.0*
  - 15.7: Handling Missing Values - *CC BY-SA 4.0*
  - 15.8: Testing the Significance of a Correlation - *CC BY-SA 4.0*
  - 15.9: Regarding Regression Coefficients - *CC BY-SA 4.0*
  - 15.10: Assumptions of Regression - *CC BY-SA 4.0*
  - 15.11: Model Checking - *CC BY-SA 4.0*
  - 15.12: Model Selection - *CC BY-SA 4.0*
  - 15.13: Summary - *CC BY-SA 4.0*
- 16: Research Design - *Public Domain*
  - 16.1: Scientific Method - *Public Domain*
  - 16.2: Measurement - *Public Domain*
  - 16.3: Data Collection - *Public Domain*
  - 16.4: Sampling Bias - *Public Domain*
  - 16.5: Experimental Designs - *Public Domain*
  - 16.6: Causation - *Public Domain*
  - 16.7: Statistical Literacy - *Public Domain*
  - 16.E: Research Design (Exercises) - *Public Domain*
- 17: Preparing Datasets and Other Pragmatic Matters - *CC BY-SA 4.0*

- 17.1: Tabulating and Cross-tabulating Data - *CC BY-SA 4.0*
- 17.2: Transforming and Recoding a Variable - *CC BY-SA 4.0*
- 17.3: A few More Mathematical Functions and Operations - *CC BY-SA 4.0*
- 17.4: Extracting a Subset of a Vector - *CC BY-SA 4.0*
- 17.5: Extracting a Subset of a Data Frame - *CC BY-SA 4.0*
- 17.6: Sorting, Flipping and Merging Data - *CC BY-SA 4.0*
- 17.7: Reshaping a Data Frame - *CC BY-SA 4.0*
- 17.8: Working with Text - *CC BY-SA 4.0*
- 17.9: Reading Unusual Data Files - *CC BY-SA 4.0*
- 17.10: Coercing Data from One Class to Another - *CC BY-SA 4.0*
- 17.11: Other Useful Data Structures - *CC BY-SA 4.0*
- 17.12: Miscellaneous Topics - *CC BY-SA 4.0*
- 17.13: Summary - *CC BY-SA 4.0*

- 18: Basic Programming - *CC BY-SA 4.0*
  - 18.1: Scripts - *CC BY-SA 4.0*
  - 18.2: Loops - *CC BY-SA 4.0*
  - 18.3: Conditional Statements - *CC BY-SA 4.0*
  - 18.4: Writing Functions - *CC BY-SA 4.0*
  - 18.5: Implicit Loops - *CC BY-SA 4.0*
  - 18.6: Summary - *CC BY-SA 4.0*
- 19: Bayesian Statistics - *CC BY-SA 4.0*
  - 19.1: Probabilistic Reasoning by Rational Agents - *CC BY-SA 4.0*
  - 19.2: Bayesian Hypothesis Tests - *CC BY-SA 4.0*
  - 19.3: Why Be a Bayesian? - *CC BY-SA 4.0*
  - 19.4: Evidentiary Standards You Can Believe - *CC BY-SA 4.0*
  - 19.5: The p-value Is a Lie. - *CC BY-SA 4.0*
  - 19.6: Bayesian Analysis of Contingency Tables - *CC BY-SA 4.0*
  - 19.7: Bayesian t-tests - *CC BY-SA 4.0*
  - 19.8: Bayesian Regression - *CC BY-SA 4.0*
  - 19.9: Bayesian ANOVA - *CC BY-SA 4.0*
  - 19.10: Summary - *CC BY-SA 4.0*
- 20: Case Studies and Data - *Public Domain*
  - 20.1: Angry Moods - *Public Domain*
  - 20.2: Flatulence - *Public Domain*
  - 20.3: Physicians Reactions - *Public Domain*
  - 20.4: Teacher Ratings - *Public Domain*
  - 20.5: Diet and Health - *Public Domain*
  - 20.6: Smiles and Leniency - *Public Domain*
  - 20.7: Animal Research - *Public Domain*
  - 20.8: ADHD Treatment - *Public Domain*
  - 20.9: Weapons and Aggression - *Public Domain*
  - 20.10: SAT and College GPA - *Public Domain*
  - 20.11: Stereograms - *Public Domain*
  - 20.12: Driving - *Public Domain*
  - 20.13: Stroop Interference - *Public Domain*
  - 20.14: TV Violence - *Public Domain*
  - 20.15: Obesity and Bias - *Public Domain*
  - 20.16: Shaking and Stirring Martinis - *Public Domain*
  - 20.17: Adolescent Lifestyle Choices - *Public Domain*
  - 20.18: Chocolate and Body Weight - *Public Domain*
  - 20.19: Bedroom TV and Hispanic Children - *Public Domain*
  - 20.20: Weight and Sleep Apnea - *Public Domain*
  - 20.21: Misusing SEM - *Public Domain*
  - 20.22: School Gardens and Vegetable Consumption - *Public Domain*
  - 20.23: TV and Hypertension - *Public Domain*
  - 20.24: Dietary Supplements - *Public Domain*
  - 20.25: Young People and Binge Drinking - *Public Domain*
  - 20.26: Sugar Consumption in the US Diet - *Public Domain*
  - 20.27: Nutrition Information Sources and Older Adults - *Public Domain*
  - 20.28: Mind Set - Exercise and the Placebo Effect - *Public Domain*
  - 20.29: Predicting Present and Future Affect - *Public Domain*
  - 20.30: Exercise and Memory - *Public Domain*
  - 20.31: Parental Recognition of Child Obesity - *Public Domain*
  - 20.32: Educational Attainment and Racial, Ethnic, and Gender Disparity - *Public Domain*
- 21: Math Review for Introductory Statistics - *CC BY 4.0*
  - 00: Front Matter - *Undeclared*
    - TitlePage - *Undeclared*
    - InfoPage - *Undeclared*
    - Table of Contents - *Undeclared*
    - Licensing - *Undeclared*
  - 21.1: Decimals Fractions and Percents - *CC BY 4.0*
    - 21.1.1: Comparing Fractions, Decimals, and Percents - *CC BY 4.0*
    - 21.1.2: Converting Between Fractions, Decimals and Percents - *CC BY 4.0*
    - 21.1.3: Decimals- Rounding and Scientific Notation - *CC BY 4.0*
    - 21.1.4: Using Fractions, Decimals and Percents to Describe Charts - *CC BY 4.0*
  - 21.2: The Number Line - *CC BY 4.0*

- 21.2.1: Distance between Two Points on a Number Line - *CC BY 4.0*
- 21.2.2: Plotting Points and Intervals on the Number Line - *CC BY 4.0*
- 21.2.3: Represent an Inequality as an Interval on a Number Line - *CC BY 4.0*
- 21.2.4: The Midpoint - *CC BY 4.0*

- 21.3: Operations on Numbers - *CC BY 4.0*
  - 21.3.1: Area of a Rectangle - *CC BY 4.0*
  - 21.3.2: Factorials and Combination Notation - *CC BY 4.0*
  - 21.3.3: Order of Operations - *CC BY 4.0*
  - 21.3.4: Order of Operations in Expressions and Formulas - *CC BY 4.0*
  - 21.3.5: Perform Signed Number Arithmetic - *CC BY 4.0*
  - 21.3.6: Powers and Roots - *CC BY 4.0*
  - 21.3.7: Using Summation Notation - *CC BY 4.0*
- 21.4: Sets - *CC BY 4.0*
  - 21.4.1: Set Notation - *CC BY 4.0*
  - 21.4.2: The Complement of a Set - *CC BY 4.0*
  - 21.4.3: The Union and Intersection of Two Sets - *CC BY 4.0*
  - 21.4.4: Venn Diagrams - *CC BY 4.0*
- 21.5: Expressions, Equations and Inequalities - *CC BY 4.0*
  - 21.5.1: Evaluate Algebraic Expressions - *CC BY 4.0*
  - 21.5.2: Inequalities and Midpoints - *Undeclared*
  - 21.5.3: Solve Equations with Roots - *CC BY 4.0*
  - 21.5.4: Solving Linear Equations in One Variable - *CC BY 4.0*
- 21.6: Graphing Points and Lines in Two Dimensions - *CC BY 4.0*
  - 21.6.1: Finding Residuals - *CC BY 4.0*
  - 21.6.2: Find the Equation of a Line given its Graph - *CC BY 4.0*
  - 21.6.3: Find y given x and the Equation of a Line - *CC BY 4.0*
  - 21.6.4: Graph a Line given its Equation - *CC BY 4.0*
  - 21.6.5: Interpreting the Slope of a Line - *CC BY 4.0*
  - 21.6.6: Interpreting the y-intercept of a Line - *CC BY 4.0*
  - 21.6.7: Plot an Ordered Pair - *CC BY 4.0*
- Back Matter - *Undeclared*
  - Index - *Undeclared*
  - Glossary - *Undeclared*
  - Detailed Licensing - *Undeclared*

- Back Matter - *Undeclared*
  - Index - *Undeclared*
  - Glossary - *Undeclared*
  - Detailed Licensing - *Undeclared*
  - Detailed Licensing - *Undeclared*