MATH 346 - STATISTICS (KUTER)

Kristin Kuter
Saint Mary's College

LibreTexts™

Saint Mary's College

MATH 346 - Statistics (Kuter)

Kristin Kuter

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

Licensing

Labs

- Lab 10: Simple Linear Regression
- Lab 11: More Regression
- Lab 1: Getting Started with R and EDA
- Lab 2: Intro to Hypothesis Testing - Permutation Tests
- Lab 3: Parameter Estimation
- Lab 4: Samping Distributions
- Lab 5: Confidence Intervals
- Lab 6: More Hypothesis Testing - Classical Approach
- Lab 7: ANOVA
- Lab 8: More ANOVA
- Lab 9: Categorical Data

Index

Glossary

Detailed Licensing

Licensing

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

Labs

This page titled Labs is shared under a not declared license and was authored, remixed, and/or curated by Kristin Kuter.

Lab 1: Getting Started with R and EDA

Objectives:

1. Use RStudio to read and examine a data set.
2. Become familiar with R Markdown files and create a PDF from one.

Definitions:

- variable vs. observation
- quantitative vs. categorical variables
- bar chart, contingency table, histogram
- center, shape, spread of a distribution
- sample mean, sample variance, sample standard deviation, sample range
- percentile (aka quantile); first, second, and third quartiles; median
- 5-number summary, boxplot

Introduction:

One of the goals of this course is to help you gain computational fluency with a statistical language. We will use the language R in this class. R is a powerful, widely used statistical language that is free and open source. The purpose of this lab is to help you get started with this language by performing *exploratory data analysis* (EDA). Exploratory data analysis is an approach to examining and describing data to gain insight, discover structure, and detect anomalies and outliers.

Activities:

Getting Organized: The first goal is to get organized. If you haven't already done so, make a folder on your laptop or in your H: drive (personal network drive on Saint Mary's campus, for info see http://sites.saintmarys.edu/~resnet/..._10resnet.html) that will store all your materials for this class. Make a subfolder called "Labs", and within "Labs" make another subfolder called "Lab1". Finally, download the lab notebook and data file for this lab into your "Lab1" folder. (The files are on the class Blackboard site.)

Configuring RStudio: The next step is to configure RStudio. Open RStudio by double clicking the RStudio icon. By default, RStudio has four "panes", and these can be configured. Click on "RStudio > Preferences... > Pane Layout", and see what your configuration is. I like to have the following configuration:

- Upper Left: Source
- Lower left: Console
- Upper Right: Environment, History, etc.
- Lower Right: Files, Plots, etc.

Set your pane configuration to match this, if it doesn't already. Once you have done so, find and click on the "Files" tab in the lower right panel. Navigate to the "Lab1" folder you just created. Then click the "More" button in the file browser (i.e., the menu at the top of the lower right panel in RStudio) and click "Set as working directory".

Open the Lab Notebook: Before you can start experimenting, you need to set up your "lab notebook" that will store your conjectures, responses to reflections, and any results. You will be using R Markdown files to create your lab notebooks. I have provided a template for you to use on this first lab, it is the "lab1_notebook.Rmd" file you have downloaded and saved in your Lab1 folder. To open it, click on it in the file browser in the lower right pane of RStudio. The document will open in the upper-left pane.

You will edit the "lab1_notebook.Rmd" file as you work through the lab. To start, add your name as the author in line 3 of the document. Once you have completed the lab, you will "Knit" the .Rmd file which will render a PDF for you to turn in. You can try this now by clicking the "Knit" button.

Running R By Command Line: Our next task is figure out how to "run" R. The simplest approach is to enter commands one at a time on a command line. If you've configured your panes as above, there should be a "Console" tab in the lower left pane. Click on

it. You should see a ">" prompt. This is where you type commands and then hit "Enter" or "Return" to run them. Try the following:

```
1 + 2

## [1] 3
```

The answer "3" should materialize on the screen. In other words, you can use R like a calculator. Try something more advanced:

```
sqrt(3^2 + 4^2)/5

## [1] 1
```

You can save the output of these calculations in variables. For example, try the following:

```
x = sqrt(3^2 + 4^2)/5
```

Note that this time, you don't see any output when you run the above line. But if you type `x` in the Console, you will see that `x=1`. In other words, you have defined a variable in the workspace and assigned it a value. If you look at the upper right pane in RStudio and click on the "Environment" tab, you should see a list of all the variables in your workspace. (At the moment there should only be one, namely `x`.)

Pause for Reflection #1:

Suppose you didn't know that `sqrt()` was the square root function in R, but you needed to take a square root. How might you figure out that this function existed? (Hint: How do you figure *anything* out in this day and age?) Type a sentence in your "lab1_notebook.Rmd" file in RStudio that explains your approach. Then test your theory by trying to find the R function that gives absolute value. Make a note in your lab notebook about the results of this attempt.

Running R From Scripts: If you want to enter a lot of commands, it is easier to type them all in a single file and then tell R to "run the script", i.e., execute the commands in sequence, one at a time. To generate a script in RStudio, click "File > New File > R Script". In the upper left pane a new blank tab entitled "Untitled1" should materialize. The first thing to do is to give this document a proper name. Click "File > Save As", and save it as "<YourFirstName>_lab1.R".

To test out your script, write the following lines in the document:

```
y=3
z=2
```

Then put your cursor on the `y=3` line and click the "Run" button, which is in the right-hand corner of the pane. You should see the command echoed in the Console pane, and the variable `y` will show up in the Environment tab. Now put your cursor on the `z=2` line and repeat - once again, the new variable should be reflected in the Environment tab.

If you want to execute the two commands together, in one fell swoop, you can click the "Source" button. "Sourcing a script" sends the entire script to the Console to be run. You can also select a portion of lines in a script with your mouse and then, with the lines highlighted, click the "Source" button.

Running R in R Markdown Files: What if you would like to include some code in your lab notebook? Well you would be in luck! One of the reasons I am having you use R Markdown files to create your lab notebooks is because they offer the flexibility to combine text (including LaTeX!) and code in one place. To add code to an R Markdown file insert a *code chunk* either by typing the chunk delimiters directly into the R Markdown file or by using one of the following shortcuts:

- keyboard shortcut: **Ctrl + Alt + I** (Mac **Cmd + Option + I**)
- RStudio shortcut: click the "Add Chunk" button in the editor toolbar of the .Rmd file

When you "Knit" an R Markdown file, the code in any code chunks will be run and the results will be displayed in the resulting document.

Pause for Reflection #2:

The "lab1_notebook.Rmd" file already contains two code chunks in the Reflection #2 section. Each chunk has the same code, but note that the second chunk has been customized with the additional argument in the chunk header, i.e., the `echo=FALSE` set in the `{ }`. Knit the file and inspect the rendered PDF.

Add a third code chunk with the same code as the other two to your lab notebook and add the following argument: `include=FALSE`. Knit the file again and inspect the rendered PDF.

Explain what the effects of the `echo=FALSE` and `include=FALSE` arguments are on the rendered PDF. Type your explanation below the code chunks in the "lab1_notebook.Rmd" file.

Loading Data: To do statistical analysis on a computer, you need to have some way of getting your computer to read and store datasets. In R there are a number of ways to do this. It is often the case that you find data in a spreadsheet, or simply stored in a regular text file. In fact, the data file for this lab is a .csv file. To load it, write the following command into your R script:

```
FlightDelays = read.csv("FlightDelays.csv", header = TRUE, sep = ",")
```

and click the "Run" button. With any luck, you should see a new variable in your Workspace (i.e., Environment tab in upper right pane) called "FlightDelays", along with a tagline that says "4029 obs. of 10 variables". Click on this data in the Environment tab, and in the upper left pane you should be able to view the data.

Pause for Reflection #3:

Note that the name of the data file is the same as the name of the variable used to store the data in RStudio. Suppose that you didn't want to work with a variable called `FlightDelays`, and instead wanted to work with a variable simply called `data`. How can you change the variable name? Write the commands in your lab1.R script, and execute them. (Hint: First copy it, then `remove` it.)

Create a code chunk in your "lab1_notebook.Rmd" file and add the code to change the name of the dataset to `data`.

Examining Datasets: Some Key Terms Take another look at the dataset (click on it in the Environment tab to get it to display in a new window). It describes information on 4029 departures of United Airlines and American Airlines from LaGuardia Airport during May and June 2009.

Each row of the dataset is an *observation*. Each column represents a *variable* - some feature or characteristic obtained for each observation. To view the names of the variables in a dataset, try the following:

```
names(data)
```

There are two types of variables:

- *quantitative* (aka numerical) - variables that have numerical values and arithmetic operations are meaningful

- *categorical* (aka factor) - variables that have non-numerical values or numerical values but arithmetic operations are **not** meaningful

---

Pause for Reflection #4:

Consider (with your partner) what these things are in terms of this dataset. Then in your lab notebook, respond to the following questions:

1. How many observations are there?
2. How many quantitative variables are there?
3. and how many categorical ones?

Tables and Bar Charts: The categorical variable `Carrier` in the dataset assigns each flight to one of the two airlines: `UA` for United and `AA` for American. To obtain a summary of this variable, try the following:

```
table(data$Carrier)
```

To visualize the distribution of the `Carrier` variable, create a *bar chart*:

```
barplot(table(data$Carrier))
```

We can also use the `table()` function to investigate the relationship between two categorical variables. The following creates a *contingency table* to investigate the relationship between the airline `(Carrier)` and whether or not a flight was delayed more than 30 minutes `(Delayed30)`:

```
table(data$Carrier, data$Delayed30)
```

Pause for Reflection #5:

In your lab notebook, create a code chunk that will include the tables and bar chart that we just produced in your rendered PDF. Comment on the distribution of the carrier data and its relationship with delays.

Histograms, Numerical Summaries, and Boxplots: Now focus on the quantitative variable `FlightLength` in the dataset, which gives the length of flight time in minutes. Because it is a bit cumbersome to use the syntax `data$FlightLength` each time we want to work with the `FlightLength` variable, we can streamline things by giving it a new name:

```
fl = data$FlightLength
```

To see the distribution of a numeric variable, we create a *histogram*:

```
hist(fl)
```

Pause for Reflection #6:

In your lab notebook, create a code chunk that will include the histogram that we just produced in your rendered PDF.

Inspect the histogram and comment on the *center*, *shape*, and *spread* of the flight length data.

The histogram is great, but we'd like some hard numbers, too. We start with the definitions of some commonly used *sample statistics*.

Definition 1.1: Let $\{x_1, x_2, \dots, x_n\}$ be n data points, i.e., a set of quantitative data collected from a sample of the population.

1. *sample mean:* $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$
2. *sample variance:* $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$
3. *sample standard deviation:* $s = \sqrt{s^2}$
4. *sample range:* $\max\{x_1, \dots, x_n\} - \min\{x_1, \dots, x_n\}$

Definition 1.2: Let $0 < p < 1$. The $(100p)$th *percentile* of a set of quantitative data is a number, denoted π_p, that is greater than $(100p)\%$ of the data values. In particular, we have:

- *first quartile* (25th percentile): $Q_1 = \pi_{0.25}$
- *second quartile* or *median* (50th percentile): $Q_2 = m = \pi_{0.5}$
- *third quartile* (75th percentile): $Q_3 = \pi_{0.75}$

The *5-number summary* of a quantitative data consists of the minimum value, Q_1, m, Q_3, and maximum.

The sample statistics defined above can be computed for the flight length data as follows:

```
mean(fl)
var(fl)                    # sample variance
sd(fl)                     # sample standard deviation
max(fl)
min(fl)
range(fl)
median(fl)
quantile(fl)               # quartiles
quantile(fl, 0.3)          # 30th percentile
summary(fl)                # 5-number summary & mean
```

Boxplots provide a visualization of the 5-number summary. Try the following to generate a boxplot of the flight length data:

```
boxplot(fl, horizontal = FALSE) # change horizontal = TRUE to change orientation
```

Pause for Reflection #7:

In your lab notebook, copy the boxplot and write down the 5-number summary for the flight length data. Compare the two and discuss with your neighbor how boxplots are constructed from the 5-number summary.

Can you tell just from the boxplot whether or not the data is skewed? Comment. Which is bigger, the mean or the median? In what direction is the data skewed? With your neighbor, debate the claim that "if the data is skewed to the right, the mean is pulled to the

right of the median." Record your thoughts in your lab notebook.

This page titled Lab 1: Getting Started with R and EDA is shared under a not declared license and was authored, remixed, and/or curated by Kristin Kuter.

Lab 10: Simple Linear Regression

Objectives:

1. Use R to create scatterplots and calculate covariance and correlation.
2. Use linear functions to model relationships between quantitative variables.

Definitions:

- response (dependent) variable
- explanatory/predictor (independent) variable
- scatterplot
- covariance
- correlation
- residual
- RSS: residual sum or squares
- least squares line (or regression line)

Introduction:

Up to this point, our study of statistics has focused on using sample statistics to estimate population parameters. We have seen how to use those estimates to test claims about population parameters. Another use for statistics is to make *predictions* about parameter values. For example, suppose you find a footprint at the scene of a crime, can the corresponding shoe size be used to make a prediction about the height of the person that left it? In this context, variables now have different roles. There is a ***response*** variable, which is the variable that we want to predict (e.g., height), and a ***predictor*** variable (also called an ***explanatory*** variable), which is used to make the prediction (e.g., shoe size). Furthermore, data come in *pairs* of observations for both the response and predictor variables. The goal is to develop a model for the relationship between the predictor and response variables. This model can then be used to make predictions. In this lab, we will see how to create a *linear* model from sample data in order to make predictions. Before we look at developing a model, we review tools for visualizing and quantifying the relationship between variables.

Activities:

Getting Organized: *If you are already organized, and remember the basic protocol from previous labs, you can skip this section.*

Navigate to your class folder structure. Within your "Labs" folder make a subfolder called "Lab10". Next, download the lab notebook .Rmd file for this lab from Blackboard and save it in your "Lab10" folder. You will be working with the `BodyMeasures.csv` data set on this lab. You should download the data file into your "Lab10" folder from Blackboard.

Within RStudio, navigate to your "Lab10" folder via the file browser in the lower right pane and then click "More > Set as working directory". Get set to write your observations and R commands in an R Markdown file by opening the "lab10_notebook.Rmd" file in RStudio. Remember to add your name as the author in line 3 of the document. For this lab, enter all of your commands into code chunks in the lab notebook. You can still experiment with code in an R script, if you want. To set up an R Script in RStudio, in the upper left corner click “File > New File > R script”. A new tab should open up in the upper left pane of RStudio.

Scatterplots: The simplest graph for displaying two quantitative variables simultaneously is a ***scatterplot***, which uses the vertical axis for one of the variables and the horizontal axis for the other variable. For each observational pair, a dot is placed at the point with coordinates given by the pair of observations. The convention is to place the ***response*** variable on the vertical y-axis and the ***predictor*** variable on the horizontal x-axis.

For example, consider the file `BodyMeasures.csv`, which contains that data we collected in class last week on shoe size and height. Suppose we are interested in predicting an individual's height given their shoe size.

Pause for Reflection #1:

State the predictor (x) and response (y) variables in the `BodyMeasures` data set.

The `plot()` function in R is used to make scatterplots. Type the following code in a code chunk in your notebook to create a scatterplot of the `BodyMeasures` data. Note that the `text()` function is used to label points in the scatterplot, which emphasizes that the data set consists of *pairs* of observations. You can comment it out if it makes the resulting plot difficult to analyze.

```
data = read.csv("BodyMeasures.csv")
shoe = data$Shoe
height = data$Height

plot(shoe, height, xlab = "shoe size", ylab = "height (in.)",
     main = "Scatterplot of shoe size vs. height")

text(shoe, height, labels = data$ID, pos = 2)     # labels points in plot with ID numb
                                                  # pos = 2 places labels to left of p
```

Pause for Reflection #2:

Use the scatterplot you created to answer the following:

- How would you describe the **direction** and **strength** of the association between shoe size and height.
- Does the relationship between shoe size and height appear to be linear?

Covariance & Correlation: We defined two numeric measures in class last week that can be used to *quantify* the strength of a linear relationship between two numeric variables:

$$\textbf{\textit{covariance}}\text{: } \mathrm{Cov}(X,Y) = \mathrm{E}[(X-\mu_X)(Y-\mu_Y)]$$

$$\textbf{\textit{correlation}}\text{: } \rho(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y}$$

The above formulas give the *theoretical* values of covariance and correlation, i.e. the values if we knew the probability distributions of variables X and Y, as well as the *population* means and standard deviations. Since we will most likely not have such information, we *estimate* covariance and correlation using sample data.

The ***sample correlation***, denoted r, for data $(x_1, y_1), \ldots, (x_n, y_n)$, is calculated using

$$r = \frac{(1/n)\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{(1/n)\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{(1/n)\sum_{i=1}^{n}(y_i-\bar{y})^2}} \qquad \textbf{(Lab 10.1)}$$

Thankfully, there are functions in R to calculate the covariance and correlation for sample data:

```
cov(shoe, height)     # sample covariance
cor(shoe, height)     # sample correlation
```

Pause for Reflection #3:

Convert the height measurements in `BodyMeasures` from inches to *centimeters* and recalculate covariance and correlation. (Note that 1 in = 2.54 cm.)

- How do the two covariance values compare? Specifically, is there a way to get the covariance value for the measurements in centimeters from the covariance value in inches?
- Did you get the same value for correlation?

Least-Squares Regression: Whether or not you think the relationship between shoe size and height is linear, we will look at using a linear equation to model it. First, let's see how to use R to construct a "line of best fit" and then we will talk about how that line is actually calculated.

The function `lm()` in R is used to construct a `l`inear `m`odel from sample data. Type the following code in your lab notebook:

```
lin.reg = lm(height ~ shoe) # list the response first, then the predictor
```

Once the linear model is fit, we can visualize it in the scatterplot with the following:

```
plot(shoe, height, xlab = "shoe size", ylab = "height (in.)",
     main = "Scatterplot of shoe size vs. height")
abline(lin.reg, col="orange") # plot the line of best fit in orange
```

The equation of a generic line can be written as $\hat{y} = a + bx$, where y denotes the response variable and x denotes the predictor variable. In this case, x represents shoe size and y represents height, and it is good form to use variable names in the equation. The caret on the y (read as "y-hat") indicates that its values are ***predicted***, not actual, heights. The symbol a represents the value of the ***y*-intercept** of the line, and b represents the value of the **slope** of the line. The slope is the coefficient on x in the linear model. Typing the stored linear model into R displays the value of a and b for the linear model fit to the `BodyMeasures` data.

```
lin.reg
```

Pause for Reflection #4:

- Identify the values of a and b for the line of best fit relating shoe size to height.
- Write the equation of the line of best fit using the format $\hat{y} = a + bx$, but replace x and y with the variable names.

What do we mean by "best"? In most statistical applications, we pick the line $\hat{y} = a + bx$ to make the *vertical distances* from observations to the line as small as possible, as shown in Figure 1 below. The reason for using vertical distances is that we use x, the predictor variable, to predict or explain y, the response variable, and we try to make the *prediction errors* as small as possible.

The prediction errors are referred to as *residuals*. A ***residual*** is the difference between the observed y value and the y value predicted by the linear model for the corresponding x value:

$$\textit{residual} = y - \hat{y}$$

The line of best fit is found by *minimizing the* ***residual sum of squares*** **(RSS)**. The line that achieves the exact minimum value of RSS is called the ***least squares line***, or the ***regression line***.

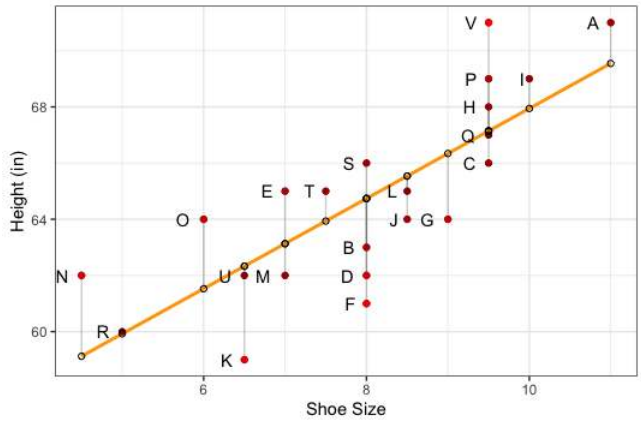


Figure 1: Least Squares Regression Line with Residuals

Pause for Reflection #5:

- Calculate the residual for the observation labeled C in the `BodyMeasures` data set. To do this, first use the least squares regression line to find the predicted height for C. Note that you found the equation of the least squares regression line in Reflection #4.
- In Figure 1, what is the ID with the *largest* residual? What is the ID of the *smallest* residual (in absolute value)?

Pause for Reflection #6:

- Use the least squares regression line to predict the height of someone whose shoe size is 8.
- Use the least squares regression line to predict the height of someone whose shoe size is 9.
- By how much do these predictions differ? Does this number look familiar? Explain.
- What height would the least squares regression line predict for a person with a shoe size of 0? Does this make sense? Explain.

Lab 10: Simple Linear Regression is shared under a not declared license and was authored, remixed, and/or curated by LibreTexts.

Lab 11: More Regression

Objectives:

1. Understand how to interpret the correlation coefficient and determine when it indicates a significant linear relationship.
2. Understand the difference between *causation* and *association*.

Introduction:

In the last lab we saw how to use R to construct least squares regression lines in order to fit linear models to data for the purpose of predicting the value of the response variable from values of the predictor variable. But before such a model is used in practice to make predictions, we should first determine whether or not the model (which is estimated from sample data) can reasonably be extended to the general population. In this lab, we further explore how to interpret correlation coefficients and how to determine when their values indicate that the linear relationship between two variables is significant.

Activities:

Getting Organized: *If you are already organized, and remember the basic protocol from previous labs, you can skip this section.*

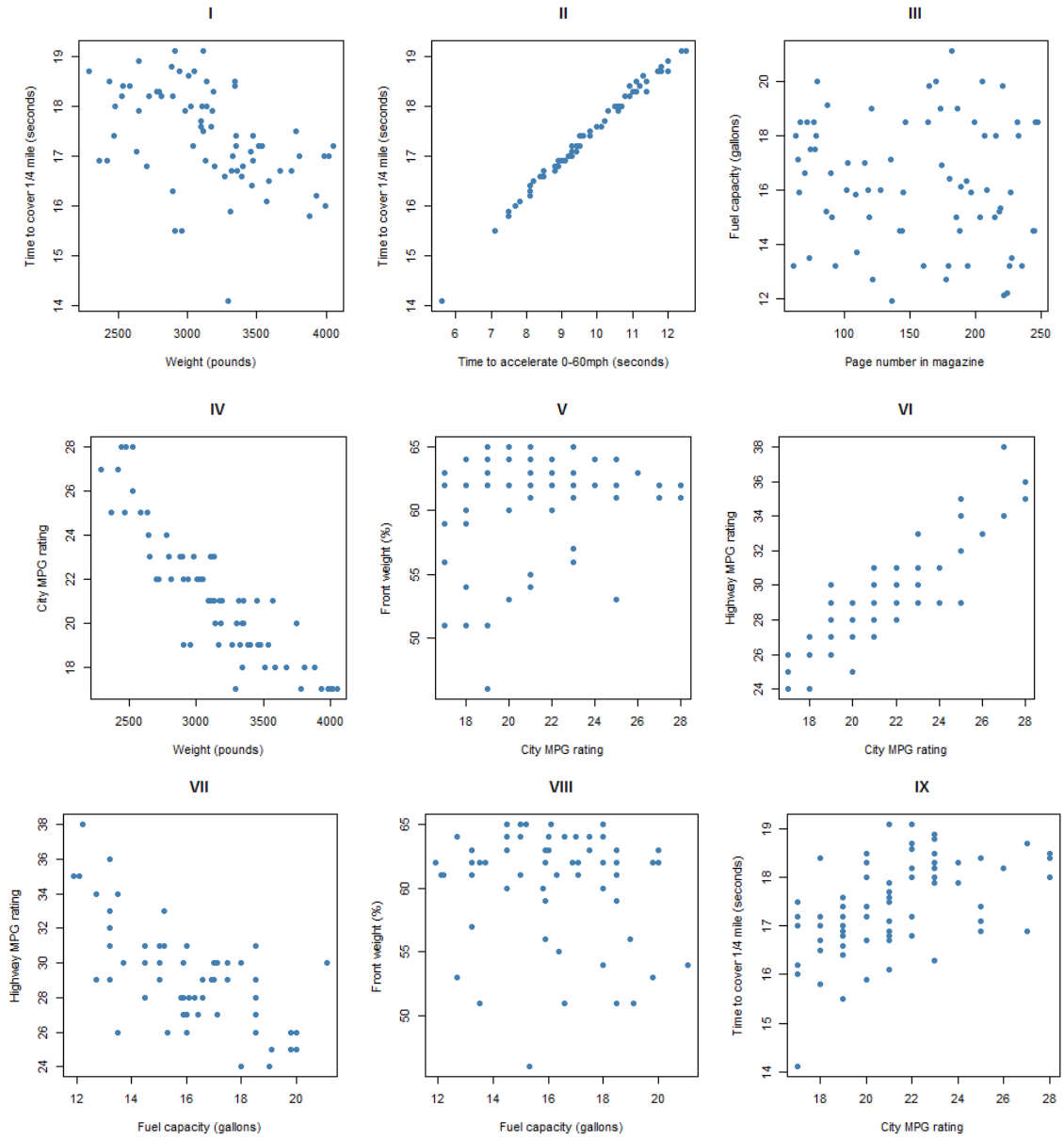
Navigate to your class folder structure. Within your "Labs" folder make a subfolder called "Lab11". Next, download the lab notebook .Rmd file for this lab from Blackboard and save it in your "Lab11" folder. You will be working with the `Cars.csv`, `TVlife.csv,` and `KYDerby.csv` data sets on this lab. You should download the data files into your "Lab11" folder from Blackboard.

Within RStudio, navigate to your "Lab11" folder via the file browser in the lower right pane and then click "More > Set as working directory". Get set to write your observations and R commands in an R Markdown file by opening the "lab11_notebook.Rmd" file in RStudio. Remember to add your name as the author in line 3 of the document. For this lab, enter all of your commands into code chunks in the lab notebook. You can still experiment with code in an R script, if you want. To set up an R Script in RStudio, in the upper left corner click “File > New File > R script”. A new tab should open up in the upper left pane of RStudio.

Correlation Coefficient: A correlation coefficient for two variables will always be between -1 and +1, and it only equals one of those values when the variables are perfectly correlated, which is evident when observations form a perfectly straight line in a scatterplot. The sign of the correlation coefficient reflects the direction of the association (e.g., positive values of r correspond to a positive linear association). When the form of the association is linear, the *magnitude* of the correlation coefficient indicates the *strength* of the association, with values closer to -1 or +1, signifying a stronger linear association.

The nine scatterplots below pertain to models of cars described in *Consumer Reports' New Car Buying Guide*. The eight variables represented here are the following:

- City MPG (in miles per gallon) rating
- Highway MPG (in miles per gallon) rating
- Weight (in pounds)
- Percentage of weight in the front half of the car
- Time to accelerate (in seconds) from 0 to 60 miles per hour
- Time to cover 1/4 mile (in seconds)
- Fuel capacity (in gallons)
- Page number of the magazine on which the car was described



Pause for Reflection #1:

Evaluate the *direction* and *strength* of the association between the variables in each scatterplot above. Do this by arranging the scatterplots for those that reveal the most strongly negative association between the variables, to those that reveal virtually no association, to those that reveal the most strongly positive association. Arrange them by number using the format of the following table.

	Negative				None	Positive			
	Strongest			Weakest		Weakest			Strongest
Number of Scatterplot									
Correlation Coefficient									

Furthermore, match the following correlation coefficients to each scatterplot:

a. 0.888
b. 0.51
c. -0.89
d. -0.157
e. -0.45
f. 0.222
g. 0.994
h. -0.094
i. -0.69

Pause for Reflection #2:

Comment on the results of the previous reflection. Specifically, reflect on why the strength and direction of the relationship between specific variables makes sense given the context. For example, scatterplot II looks at the relationship between the time to accelerate from 0 to 60 mph and the time to cover 1/4 mile. This scatterplot exhibits the strongest positive relationship. Why would the two variables considered here have the strongest positive relationship amongst all variables considered?

Significant Correlation: The slope, or steepness, of the points in a scatterplot is unrelated to the value of the correlation coefficient. If the points fall on a perfectly straight line with a positive slope, then the correlation coefficient equals 1.0 whether that slope is very steep or not steep at all. In other words, correlation can be +1 for points lying on a line with slope $m = 0.1$ or slope $m = 10$. What matters for the magnitude of the correlation is how closely the points concentrate around a line, not the steepness of a line.

For example, look at scatterplots VI and IX above. The linear trend in each scatterplot is positive with a similar slope. However, the points in VI are more tightly clustered along the line while the points in IX are more spread out. So we would say that the association between the variables in VI (city and highway mpg rating) is stronger than the association between the variables in IX (city mpg rating and time to cover 1/4 mile). Thus, we could more confidently conclude that there appears to be a linear relationship between city mpg and highway mpg, i.e., make an inference about the population of cars based on the sample of cars we have data for.

This begs the question, when does a *sample* correlation coefficient provide sufficient evidence of a linear relationship between two variables? In other words, we are essentially asking when is a sample correlation coefficient significantly different from 0 (close enough to -1 or +1) in order to conclude a relationship in the population based on sample data)? This is equivalent to testing whether or not the *slope* of the least squares regression line is significantly different from 0. Let's demonstrate how to do this for the correlation coefficient between city mpg and highway mpg for the cars data.

```
cars = read.csv("Cars.csv")
lin.reg = lm(cars$HighwayMPG ~ cars$CityMPG)     # first, construct linear model
summary(lin.reg)                                 # second, display a summary of the ir

##
## Call:
## lm(formula = cars$HighwayMPG ~ cars$CityMPG
##
## Residuals:
##     Min          1Q     Median         3Q        Max
## -3.6782    -0.9211    0.0181    0.9574    3.4433
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.19659    1.22854   7.486  1.5e-10 ***
## cars$CityMPG  0.93926    0.05781  16.247  < 2e-16 ***
## ---
## Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
##
## Residual standard error: 1.421 on 71 degrees of freedom
## Multiple R-squared:  0.788,    Adjusted R-squared:  0.785
## F-statistic:   264 on 1 and 71 DF,  p-value: < 2.2e-16
```

There is quite a bit of information displayed when calling the `summary()` function on the stored results of the linear model function `lm()`. But the relevant information used for testing the significance of the linear relationship is the P-value associated to the test of whether the slope of the least squares regression line, i.e., the coefficient on the x variable (city mpg in this case), is significantly different from 0, which is highlighted in yellow. As we can see, the P-value is very small in this case (less than 2×10^{-16}), which indicates that the slope is significantly different from 0 and thus so is the correlation coefficient.

Pause for Reflection #3:

Determine if the correlation coefficients between the pairs of variables in the following scatterplots from above for the cars data are significant.

- scatterplot IX
- scatterplot VIII

Pause for Reflection #4:

Explain why it is equivalent to test whether the *slope* of the least squares line is significantly different from 0 in order to determine if the corresponding *correlation coefficient* is significantly different from 0.

Correlation vs. Causation: We have to be very careful when interpreting correlation coefficients, especially when we find that they are significant. One of the major errors made in interpreting significant correlation between two variables is to conclude a *cause-and-effect* relationship between the variables.

For example, the data set `TVlife.csv` provides information on the life expectancy and number of televisions per thousand people in a sample of 22 countries, as reported by *The World Almanac and Book of Facts*. Suppose we are interested in predicting life expectancy in a country from the number of TVs.

Pause for Reflection #5:

Using the data in `TVlife.csv`, create a scatterplot of life expectancy vs. number of TVs.

Additionally, estimate the correlation coefficient between life expectancy and number of TVs. Is it significant?

Because the association between the variables is so strong, you might conclude that simply sending televisions to the countries with lower life expectancies would cause their inhabitants to live longer. Comment on this argument.

This example illustrates the very important distinction between *association* and *causation*. Two variables might be strongly associated without having a cause-and-effect relationship between them. Often with observational studies, both variables are related to a third (***confounding***) variable.

Pause for Reflection #6:

In the case of life expectancy and television sets, suggest a confounding variable that is associated both with a country's life expectancy and with the prevalence of televisions in that country.

Non-linear Relationships: Another common mistake when interpreting correlation coefficients is to conclude that the relationship is linear. The correlation coefficient measures the degree of *linear* association between two quantitative variables. But even when two variables display a *nonlinear* relationship, the correlation between them still might be quite close to $\pm$ 1. To demonstrate this consider the `KYDerby.csv` data set.

The Kentucky Derby is the most famous horse race in the world, held annually on the first Saturday in May at Churchill Downs race track in Louisville, Kentucky. This race has been called "The Most Exciting Two Minutes in Sports" because that's about how long it takes for a horse to run its 1.25-mile track. The file `KYDerby.csv` contains the winning time (in seconds) for every year since 1896, when the track length was changed to 1.25 mile, along with the track condition (fast, good, or slow) on the day of the race.

Pause for Reflection #7:

Create a scatterplot of winning time (y) vs. year (x). Comment on the shape of the relationship between the variables.

Additionally, calculate the correlation coefficient and determine if it is significant.

With these data, the relationship is clearly curved and not linear, and yet the correlation is still close to -1. Do not assume from a significant correlation coefficient that the relationship between the variables must be linear. Always look at a scatterplot, in conjunction with the correlation coefficient, to assess the form (linear or not) of the association.

Lab 11: More Regression is shared under a not declared license and was authored, remixed, and/or curated by LibreTexts.

Lab 2: Intro to Hypothesis Testing - Permutation Tests

Objective:

Understand how to use R to perform permutation tests.

Definitions

- hypothesis testing, statistical significance
- null vs. alternative hypothesis
- test statistic, observed test statistic, *P*-value
- permutation test: permutation resample, permutation distribution

Introduction:

Recall the question posed in class on Tuesday: "If you could stop time and live forever in good health at a particular age, at what age would you like to live?"

Suppose we are interested in testing the claim that the average ideal age for women is *greater* than men. A random sample of 3 women and 3 men were asked this question resulting in the following responses:

	Women	Men
	49 42 38	29 38 50

The average response for the women is 43, and the average age response for the men is 39.

The difference in mean response between the women and men is 43 - 38 = 4 years.

In the observed sample, the average response for women is greater than men, but this result could be due to random chance alone, rather than an actual difference between men and women. If there is no real difference, then the split of the 6 observations into the two groups is essentially random. We could have just as easily observed:

	Women	Men
	29 42 38	49 38 50

Now the difference in mean response between the women and men is -9.3 years.

So how do we determine if the result we actually observe provides evidence of a claim? We use probability and determine how easily pure random chance would produce a given result. This is the core idea of ***statistical significance*** or classical ***hypothesis testing***, to calculate the probability that pure random chance would give an effect as extreme as that observed in the data, in the absence of any real effect. If that probability (referred to as the ***P-value***) is small enough, we conclude that the data provide convincing evidence of a real effect.

Activities:

Getting Started: Navigate to your class folder structure. Within your "Labs" folder make a subfolder called "Lab2". Next, download the lab notebook for this lab from Blackboard and save it in your "Lab2" folder. You will again be working with

the `FlightDelays` dataset on this lab. You should either re-download the data file into your "Lab2" folder from Blackboard, or just copy the file from your "Lab1" folder into the "Lab2" folder.

Within RStudio, navigate to your "Lab2" folder via the file browser in the lower right pane and then click "More > Set as working directory". Get set to write your observations and R commands in an R Markdown file by opening the "lab2_notebook.Rmd" file in RStudio. Remember to add your name as the author in line 3 of the document. For this lab, enter all of your commands into code chunks in the lab notebook. You can still experiment with code in an R script, if you want. To set up an R Script in RStudio, in the upper left corner click “File > New File > R script”. A new tab should open up in the upper left pane of RStudio.

Ideal Age Example: If we let μ_W denote the true mean response for women to the question posed above, and we let μ_M denote the true mean response for men, then we set up in class on Tuesday the following ***null*** and ***alternative hypotheses*** to test the claim that the average ideal age for women is greater than men.

$$\begin{aligned} H_0 &: \mu_W - \mu_M = 0 \quad \text{(there is no difference in the mean response between women and men)} \\ &\text{vs.} \\ H_A &: \mu_W - \mu_M > 0 \quad \text{(the mean response for women is greater than the mean response for men)} \end{aligned}$$

We also stated that the ***test statistic*** used to determine the result of the above test is given by the difference in the respective sample means. So, if we let $\bar{X}_W$ denote the sample mean of the responses from the women, and $\bar{X}_M$ denotes the sample mean for the men, the test statistic is $T = \bar{X}_W - \bar{X}_M$. Given the above data, the ***observed test statistic*** in this case is $t = \bar{x}_W - \bar{x}_M = 43 - 39 = 4$.

To determine whether or not the observed difference of 4 indicates a real difference, we will compute the associated P-value by working out all possible splits of the 6 observed responses into two groups and calculating how many produce a test statistic as large or larger than what was actually observed. Formally, the P-value is given by

$$P\text{-value } = P(T \geq 4).$$

Pause for Reflection #1:

How many different splits of the 6 numbers {29, 38, 38, 42, 49, 50} into two groups of 3 (ignoring ordering within each group) are possible? (*Hint:* How did we count the number of unordered things in probability?)

With your neighbor, write out all the possible splits in your lab notebook and calculate the corresponding test statistic for each of the possible splits. Then find the P-value by calculating the proportion of splits that resulted in a difference in mean response between women and men as large or larger than what was actually observed. Code has been started in the lab notebook for you to do these calculations with R.

Based on the P-value you find, do you think the true mean response from women is greater than men? Discuss this with your neighbor.

Statistically Significant: A result is considered ***statistically significant*** if it would rarely occur by chance. This begs the question, "how rare does the result need to be?" The answer: It depends on the context! But, for example, a P-value of 0.0002 would indicate that assuming the null hypothesis is true, the observed outcome would occur just 2 out of 10000 times by chance alone, which is most circumstances seems pretty rare and you would conclude that the evidence supports the alternative hypothesis.

Flight Delays Example: Recall the `FlightDelays` dataset from Lab 1, which contains information on 4029 departures of United Airlines and American Airlines from LaGuardia Airport during May and June 2009. In this lab, you will focus on the variable `Delay`, which gives the minutes that a flight was delayed (note that negative values indicate early departures). So, first load the dataset and then create a new object for easy reference to the `Delay` variable:

```
FlightDelays = read.csv("FlightDelays.csv", header = TRUE, sep = ",")
delay = FlightDelays$Delay
```

Recall from Lab 1, that a higher proportion of United flights were delayed more than 30 minutes. So we are going to test the claim that the mean delay for United flights is more than the mean delay for American flights. We can compute the average delay for the two airlines using the `tapply()` function in R. The `tapply()` function allows you to compute numeric summaries of quantitative variables based on levels of a categorical variable. For instance, the following finds the sample mean flight delay length by airline in the `FlightDelays` dataset:

```
tapply(delay, FlightDelays$Carrier, mean)

##       AA         UA
## 10.09738   15.98308
```

The mean delay for the sample of United flights is $\bar{x}_U = 15.98$ and the mean delay for American flights is $\bar{x}_A = 10.10$. The sample means are clearly different, but the difference (15.98 - 10.10 = 5.88 min) could have arisen by chance. Can the difference *easily* be explained by chance alone? If not, we will conclude that there are genuine differences in the mean delay times for the two airlines.

Hypotheses: In order to perform a hypothesis test of the stated claim, let μ_U denote the true mean delay time for United flights, and let μ_A denote the true mean delay for American flights. We will use $T = \bar{X}_U - \bar{X}_A$ as the test statistic, with an observed value of $t = 5.88$ min.

Pause for Reflection #2:

In your lab notebook, write the null and alternative hypotheses to test the claim that United flights have a longer mean delay than American flights. Note that you can use LaTeX in R Markdown files, which will help you typeset the notation used in stating the hypotheses. The syntax has been provided in the lab notebook.

Permutation Resampling: Suppose there really is no difference in the mean delay between the two airlines. Then the 4029 observed delay times come from a single population, the way they were divided into two groups (by labeling some as American flights and others as United) is essentially random, and any other division is equally likely. We could proceed, as in the ideal age example, calculating the difference in means for *every* possible way to split the data into two samples. However, that would entail looking at the number of ways to choose 1123 objects (the number of United flights in the dataset) from a total of 4029 objects (the total number of observations). This number is *astronomical!* Instead, we use sampling.

We create a ***permutation resample*** by randomly drawing $m = 1123$ observations *without* replacement from the pooled data to be one sample (the United flights), leaving the remaining $n = 2906$ observations to be the second sample (the American flights). We then calculate the test statistic for the new samples. Repeating this process many times (1000 or more), we can then calculate the P-value by finding the proportion of times the resulting test statistic equals or exceeds the original observed test statistic.

The distribution of the test statistic across all permutation resamples is the ***permutation distribution***. This may be exact (i.e., calculated exhaustively as in the ideal age example) or approximate (i.e., implemented by sampling, as you will do next for the flight delays).

Two-sample Permutation Test: The following code walks you through performing a permutation test of the claim that the mean delay for United flights is longer than the mean delay for American flights.

Type each command below into the code chunk provided under the "Two-sample Permutation Test" heading in the lab notebook.

First, create an object to store the value of the observed test statistic:

```
observed = 15.98308 - 10.09738
```

To draw a permutation resample, you will draw a random sample of size 1123 from the numbers 1 through 4029 (there are 4029 observations total). The times corresponding to these positions in the `delay` vector you created earlier will be values for the

United flights and the remaining ones for the American flights. The difference in means for this permutation will be stored in an object called `result`. This will be repeated many times.

```
N = 10^5 - 1                    # number of times to repeat this process
result = numeric(N)             # space to save the random differences in each permutat
for (i in 1:N)
{ # sample of size 1123, from 1 to 4029, without replacement
  index = sample(4029, size = 1123, replace = FALSE)
  result[i] = mean(delay[index]) - mean(delay[-index])
}
```

To analyze the results, first create a histogram of the (approximate) permutation distribution and add a vertical line at the observed test statistic.

```
hist(result, xlab = "xbarU - xbarA", main = "Permutation Distribution for delays")
abline(v = observed, col = "blue")      # add line at observed mean difference
```

Finally, compute the ***P***-value by finding how many times a permutation resample produced a test statistic as large or larger than the observed value.

```
(sum(result >= observed) + 1)/(N + 1)      # P-value
```

The code snippet `result >= observed` results in a vector of `TRUE`'s and `FALSE`'s depending on whether or not the mean difference computed for a resample is greater than the observed mean difference. `sum(result >= observed)` then counts the number of `TRUE`'s.

Pause for Reflection #3:

Consider the histogram and *P*-value that the above code produced. Is the result statistically significant? In other words, do you think there is a real difference in the mean delay times between United and American flights? Type a response in your lab notebook.

Choice of Test Statistic: In the examples above, we used the difference in means. We could have equally well used a variety of other test statistics, e.g., a difference in medians. It turns out, that if two statistics are monotonically related, i.e., one is always larger than the other, then the choice of one or the other as test statistic will result in exactly the same ***P***-value. Let's explore this.

Repeat the permutation test of flight delays using (i) the difference in means, (ii) the mean of the United delay times, (iii) the sum of United delay times, and (iv) the difference in medians. You want to compute these statistics for the same permutation resamples, so find them all in the same `for` loop. The following code has already been added to the lab notebook:

```
result1 = numeric(N)                # space to save the differences in means
result2 = numeric(N)                # space to save the United means
result3 = numeric(N)                # space to save the sums of United delays
result4 = numeric(N)                # space to save the differences in medians
for (i in 1:N)
{ # sample of size 1123, from 1 to 4029, without replacement
  index = sample(4029, size = 1123, replace = FALSE)
  result1[i] = mean(delay[index]) - mean(delay[-index])
```

```
  result2[i] = mean(delay[index])
  result3[i] = sum(delay[index])
  result4[i] = median(delay[index]) - median(delay[-index])
}
```

Pause for Reflection #4:

Compute and compare the P-values obtained for the four different test statistics used in your lab notebook. What do you observe?

Note: You will need to compute the corresponding observed test statistic for the three new test statistics. To do so, make use of the `tapply()` function.

Adding One: When computing the P-value for the permutation test, we add one to both the numerator and denominator. This corresponds to including the original data as an extra resample.

Pause for Reflection #5:

Discuss with your neighbor why you should add one, and include the original data, when computing the P-value. Record your thoughts in your lab notebook.

One- and Two-sided Tests: In the flight delays example, we had an initial hunch that United flights had a longer mean delay than American, so we performed a one-sided permutation test for a claim of "increase". However, we could have also tested the claim as a statement of "decrease", i.e., that American flights have a shorter mean delay than United. This would still be a one-sided test.

Instead of performing a one-sided test altogether though, we could have also performed a two-sided test, which would simply be a test of no difference, not claiming that one airline's mean delay time is longer or shorter than the other. When performing a two-sided permutation test, we calculate both one-sided P-values, multiply the smaller by 2, and if necessary round down to 1.0.

Two-sided P-values are the default in statistical practice: you should perform a two-sided test unless there is a clear reason to pick a one-sided alternative hypothesis. It is not fair to look at the data before deciding to use a one-sided hypothesis.

Pause for Relection #6:

Consider the one-sided permutation test for a claim of "decrease" in the flight delays example. Write down in your lab notebook what the alternative hypothesis would be in this case and describe how you would calculate the P-value. Also, comment on why we multiply by 2 when calculating a two-sided P-value.

Lab 2: Intro to Hypothesis Testing - Permutation Tests is shared under a not declared license and was authored, remixed, and/or curated by LibreTexts.

Lab 3: Parameter Estimation

Objective

Explore properties of estimators and understand what makes an estimator preferred.

Definitions

- estimator vs. estimate
- maximum likelihood estimation: likelihood function, log-likelihood
- method of moments estimation
- bias, unbiased estimator
- efficiency of estimators
- mean square error (MSE)
- bias-variance trade-off

Introduction

In class this week, we went over two procedures for estimating parameters: ***maximum likelihood estimation*** and ***method of moments***. There are other methods of estimation that may exist in a given context, including using "plug-in" estimators. This begs the question of which method is best. In this lab, you will explore properties of estimators and using these properties learn what criteria we think good estimators should satisfy. Each property provides a "sniff test": an estimator that fails these just doesn't smell right.

Activities

Getting Started: Navigate to your class folder structure. Within your "Labs" folder make a subfolder called "Lab3". Next, download the lab notebook .Rmd file for this lab from Blackboard and save it in your "Lab3" folder. There are no datasets used in this lab.

Within RStudio, navigate to your "Lab3" folder via the file browser in the lower right pane and then click "More > Set as working directory". Get set to write your observations and R commands in an R Markdown file by opening the "lab3_notebook.Rmd" file in RStudio. Remember to add your name as the author in line 3 of the document. For this lab, enter all of your commands into code chunks in the lab notebook. You can still experiment with code in an R script, if you want. To set up an R Script in RStudio, in the upper left corner click “File > New File > R script”. A new tab should open up in the upper left pane of RStudio.

Maximum Likelihood Estimation: Before we get into the properties, let's revisit maximum likelihood estimation.

Likelihood Function, Maximum Likelihood Estimate

Suppose $X_1, ..., X_n$ represent a random sample from a probability distribution with associated parameter θ and pmf/pdf given by $f(x;\theta)$. The ***likelihood function*** $L(\theta) = L(\theta|x_1, ..., x_n)$ gives the likelihood of θ, given the observed sample values $x_1, ..., x_n$, and is calculated as follows:

$$L(\theta) = \prod_{i=1}^{n} f(x_i;\theta) = f(x_1;\theta) \cdot f(x_2;\theta) \cdots f(x_n;\theta)$$

The ***maximum likelihood estimate*** (MLE) $\hat{\theta}_{MLE}$ is a value of θ that maximizes the likelihood function, or equivalently that maximizes the log-likelihood function: $\ln L(\theta)$.

So, the likelihood function $L(\theta)$ is a function of the unknown parameter θ, and we estimate θ by maximizing $L(\theta)$. Remember that we can use calculus to find the value of θ that maximizes the likelihood by setting the derivative equal to 0, $L'(\theta) = 0$, and then solving for θ to find the MLE.

In practice, we usually maximize the ***log-likelihood***, because taking the logarithm of a product results in a sum:

$$\ln L(\theta)) = \ln\,[f(x_1;\theta)f(x_2;\theta)...f(x_n;\theta)]$$

$$=\ln f(x_1;\theta))+\ln(f(x_2;\theta)+...+\ln f(x_n;\theta)) = \sum_{i=1}^{n}\ln(f(x_i;\theta))$$

In most cases, we are able to find a closed-form expression for the MLE. However, this is not always possible, as the following example demonstrates.

Example: Suppose $X_1, ..., X_n$ are a random sample from the Cauchy distribution, which has pdf given by $f(x;\theta) = \dfrac{1}{\pi(1+(x-\theta)^2}$, for x, $\theta \in \mathbb{R}$. The likelihood function for θ is

$$L(\theta) = \frac{1}{\pi^n \prod_{i=1}^{n}[1+(X_i-\theta)^2]}. \qquad (1)$$

Thus, $L(\theta)$ will be maximized when $\prod_{i=1}^{n}[1+(X_i-\theta)^2]$ is minimized, or equivalently when $\sum_{i=1}^{n}\ln(1+(X_i-\theta)^2)$ is minimized. The value of θ that minimizes this expression must be determined by numerical methods.

Pause for Reflection #1:

1. On a separate piece of paper, write out the details for deriving the likelihood function $L(\theta)$ in Equation (1).
2. Next, explain why $L(\theta)$ will be maximized when the expression $\sum_{i=1}^{n}\ln(1+(X_i-\theta)^2)$ is minimized. Type your response directly into your lab notebook in RStudio.
3. Finally, on the same piece of paper you worked out step 1, take the derivative (with respect to θ) of the sum expression given in step 2 to see why the mathematical approach of setting the derivative equal to 0 will not work in this example.

Take a picture of your written work for steps 1 and 3 and upload it to your Lab3 folder in order to include in your lab notebook.

Continuing with the Cauchy distribution, suppose you make the following observations for a random sample of size four: $x_1 = 1, x_2 = 2, x_3 = 2$, and $x_4 = 3$. Then, to maximize $L(\theta)$, you need to minimize

$$\ln(1+(1-\theta)^2)+\ln(1+(2-\theta)^2)+\ln(1+(2-\theta)^2)+\ln(1+(3-\theta)^2).$$

Since we cannot find the minimum of the above analytically, you will use the `optimize()` function in R. Type the following in a code chunk in your Lab 3 notebook and run each line:

```
x = c(1, 2, 2, 3)
g = function(theta) sum(log(1 + (x-theta)^2))
optimize(g, interval = c(0,4))
```

Pause for Reflection #2

1. In your lab notebook, describe what each of the three lines of code above are doing. You may find it helpful to type `?c`, `help("function")`, and `?optimize` one at a time in the console window to pull up info in the Help window (lower right pane) for each of these commands.
2. Based on the results of this code, what is the maximum likelihood estimate of θ based on the given data?

Unbiasedness: The first property of estimators that we will consider is *bias*. An estimator $\hat{\theta}$ is biased if, on average, it tends to be too high or too low, relative to the true value of θ. Formally, this is defined using expected values:

Definition 3.1

The ***bias*** of an estimator $\hat{\theta}$ is given by

$$\text{Bias}(\hat{\theta}) = \text{E}[\hat{\theta}] - \theta.$$

In other words, Definition 3.1 states that a statistic used to estimate a parameter is ***biased*** when the mean of its sampling distribution is not equal to the true value of the parameter. We will explore sampling distributions more in depth in next week. For now, we will use R to *approximate* sampling distributions.

We like an estimator to be, on average, equal to the parameter it is estimating. That is, we like estimators that are ***unbiased***, or equivalently, Bias($\hat{\theta}$) = 0. You will show in the homework that the sample mean is always an unbiased estimator of the population mean μ. It can also be shown that the sample proportion is also an unbiased estimator of the population proportion.

The case of the sample variance is less straightforward. Given a sample of values $x_1, x_2, \ldots, x_n$, the "plug-in" estimator of the population variance σ^2 is

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2.$$

However, in Lab 1, we defined the sample variance as

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2,$$

which is computed in R using the function `var()`. Notice the difference between the two estimators, namely, the division by "n" versus "$n-1$". It turns out that the plug-in estimator $\hat{\sigma}^2$ is biased, but the sample variance s^2 is unbiased.

Let's explore this in the context of the standard normal distribution: $N(\mu = 0, \sigma = 1)$. In this context, we know the value of the parameter we are estimating, namely $\sigma^2 = 1$. So we know that in order for an estimator of σ^2 to be unbiased, its expected value needs to equal 1. You will run a simulation in R to see how the two estimators, $\hat{\sigma}^2$ and s^2, perform. With the following code (already added to the lab notebook for you), you will draw random samples of size 15 from $N(0,1)$. For each sample, you will compute $\hat{\sigma}^2$ and s^2 and record the values. You will repeat this 1000 times.

```
sample.var = numeric(1000)          # object to store sample variances
plugin = numeric(1000)              # object to store plug-in estimates
n = 15                              # set sample size
for (i in 1:1000)
{
  x = rnorm(n)                      # draw a random sample of size n from N(0,1) popula
  sample.var[i] = var(x)            # compute and store sample variance of ith sample
  plugin[i] = ((n-1)/n)*var(x)      # compute and store plug-in estimate from ith sampl
}
```

We can now investigate the results of the simulation by finding the mean for the estimates of σ^2 based on the two estimators $\hat{\sigma}^2$ (`plugin.var`) and s^2 (`sample.var`). We can also visualize the results using histograms.

Pause for Reflection #3

1. In your lab notebook, calculate the respective means for the 1000 samples of $\hat{\sigma}^2$ and s^2 you found with the simulation.
2. Code has been provided in your lab notebook to create histograms of the simulated values for $\hat{\sigma}^2$ and s^2. Run the code chunk and answer the following: Do the results you found support the claim that $\hat{\sigma}^2$ is a biased estimator of σ^2 and s^2 is unbiased? Why or why not?

Efficiency: What happens when you have two estimators that are both unbiased? Which one should you use? The next property we consider, *efficiency*, provides a criterion for comparing unbiased estimators that depends on their variance.

Definition 3.2

If $\hat{\theta}_1$ and $\hat{\theta}_2$ are both unbiased estimators of θ and $\mathrm{Var}(\hat{\theta}_1) < \mathrm{Var}(\hat{\theta}_2)$, then $\hat{\theta}_1$ is said to be more ***efficient*** than $\hat{\theta}_2$.

We will again explore this property with a simulation, this time in the context of the uniform distribution on the closed interval [0, β].

At the start of lab, we will find the ***method of moments estimator*** for β to be $\hat{\beta}_1 = 2\bar{X}$, i.e., twice the mean of a given sample. It can be shown that this estimator is unbiased (a fact we will prove later). Using maximum likelihood estimation, we can find another unbiased estimator of β given by $\hat{\beta}_2 = ((n+1)/n)X_{\max}$, where $X_{\max}$ denotes the largest value in a random sample (it is referred to as the *largest order statistic*).

Use the following code (already provided in your lab notebook) to run a simulation to see how these two estimators perform in the specific context of drawing random samples of size 25 from uniform [0,12].

```
beta.1hat = numeric(1000)
beta.2hat = numeric(1000)
for (i in 1:1000)
{
  x = runif(25, 0, 12)            # draw a random sample of size 25 from uniform[0,12]
  beta.1hat[i] = 2 * mean(x)
  beta.2hat[i] = ((25 + 1)/25) * max(x)
}
# descriptive statistics
mean(beta.1hat)
sd(beta.1hat)
mean(beta.2hat)
sd(beta.2hat)
# graphical comparison
hist(beta.1hat, xlim = c(8,16), ylim = c(0,650), xlab = "2*mean")
hist(beta.2hat, xlim = c(8,16), ylim = c(0,650), xlab = "((25+1)/25)*max")
```

Pause for Reflection #4

1. Do the results support the claim that both $\hat{\beta}_1$ (`beta.1hat`) and $\hat{\beta}_2$ (`beta.2hat`) are unbiased estimators for β? Why or why not?
2. Which estimator is more efficient, i.e., which estimator exhibits a smaller amount of variability?
3. Given these results, which estimator do you think you should use?

Mean Square Error: The final criterion we consider combines both bias and variance. This is useful for comparing estimators that are not both unbiased. We may prefer an estimator with small bias and small variance over one that is unbiased but with large variance. The following definition provides a way to quantify the preference.

Definition 3.3

The ***mean square error*** (MSE) of an estimator is $\text{MSE}(\hat{\theta}) = \text{E}[(\hat{\theta} - \theta)^2]$.

MSE measures the average squared difference between the estimator and the parameter; it takes both the variability and bias of the estimator into account, as the following proposition shows.

Proposition 3.1

$\text{MSE}(\hat{\theta}) = \text{Var}(\hat{\theta}) + [\text{Bias}(\hat{\theta})]^2$

It follows from Proposition 3.1 that if $\hat{\theta}$ is unbiased, then MSE($\hat{\theta}$) = Var($\hat{\theta}$). So, for unbiased estimators, one is more efficient than a second if and only if its MSE is smaller. But, in general, when comparing two estimators $\hat{\theta}_1$ and $\hat{\theta}_2$ of θ, we are faced with a trade-off between variability and bias.

Example: Let's explore this ***bias-variance trade-off*** in the context of the binomial distribution, where the number of trials n is known but the probability of "success" p is unknown. Let $X \sim \text{binomial}(n, p)$. The sample proportion X/n (the proportion of "successes" in n observed trials) is an unbiased estimator of p. Denote this estimator as $\hat{p}_1$, then

$$\text{E}[\hat{p}_1] = \text{E}\left[\frac{X}{n}\right] = \frac{np}{n} = p.$$

Furthermore, the mean square error of the sample proportion is

$$\text{MSE}[\hat{p}_1] = \text{Var}(\hat{p}_1) = \text{Var}\left(\frac{X}{n}\right) = \frac{np(1-p)}{n^2} = \frac{p(1-p)}{n}.$$

Consider the alternative estimator of p given by

$$\hat{p}_2 = \frac{X+1}{n+2},$$

which adds one artificial success and one failure to the real data.

Pause for Reflection #5

1. On a piece of paper, write out the details to derive the following:

$$\text{E}[\hat{p}_2] = \frac{np+1}{n+2} \text{ and } \text{Var}(\hat{p}_2) = \frac{np(1-p)}{(n+2)^2}.$$

2. Then, using Proposition 3.1, show that the mean square error for $\hat{p}_2$ is given by

$$\text{MSE}(\hat{p}_2) = \frac{np(1-p) + (1-2p)^2}{(n+2)^2}.$$

Take a picture of your written work for steps 1 and 2 and upload it to your Lab3 folder to include in your lab notebook. Refer to the code provided in the lab notebook for Reflection #1.

We can compare the two estimators $\hat{p}_1$ and $\hat{p}_2$ for p by comparing their mean squared errors. Note that we have the MSE for both estimators as a function of p. Thus, we can graphically compare the MSE for $\hat{p}_1$ and $\hat{p}_2$ by plotting curves in R using the following code. Note that we use $n = 16$ just to have a specific example to work with.

```
n = 16
curve(x*(1-x)/n, from=0, to=1, xlab="p", ylab="MSE")
curve((n*x*(1-x)+(1-2*x)^2)/(n+2)^2, add=TRUE, col="blue", lty=2)
```

The MSE for $\hat{p}_1$ is in solid black, and the MSE for $\hat{p}_2$ is the dashed blue curve.

Pause for Reflection #6

Inspect the graphs of the MSE curves and answer the following:

1. For approximately what values of p does $\hat{p}_2$ have smaller MSE than $\hat{p}_1$?
2. For the values identified in step 1, even though $\hat{p}_2$ is biased, it has a smaller MSE than $\hat{p}_1$. Comment on why $\hat{p}_2$ may be preferred over $\hat{p}_1$ as an estimator of p for these values.
3. Now alter the code above to recreate the MSE graphs for the following sample sizes: $n = 30, n = 50, n = 100, n = 200$. What do you see is the effect of increasing the sample size?

Optional Reflection #7

Prove Proposition 3.1.

Lab 3: Parameter Estimation is shared under a not declared license and was authored, remixed, and/or curated by LibreTexts.

Lab 4: Samping Distributions

Objectives

1. Understand the difference between the ***distribution of a population*** and the ***distribution of a sample***.
2. Understand how to use the ***Central Limit Theorem*** to approximate sampling distributions.
3. Assess normality of a sample using ***normal quantile plots***.

Definitions

- random sample
- estimator
- sample mean
- sampling distribution
- normal quantile plot
- Central Limit Theorem

Introduction

As we have seen, we obtain random samples from populations in order to understand characteristics of the population. Last week we introduced methods for estimating parameters based on a random sample. These methods produced ***estimators***, which are functions of the random sample, and are more generally referred to as *statistics*. The values of an estimator (or statistic) depend on the random sample and because of this they are random variables themselves. Thus, we can use probability theory to help analyze estimators and statistics. In this lab, you will explore the *sampling distribution* of a statistic, which simply refers to the probability distribution of the statistic.

Activities

Getting Organized: *If you are already organized, and remember the basic protocol from previous labs, you can skip this section.*

Navigate to your class folder structure. Within your "Labs" folder make a subfolder called "Lab4". Next, download the lab notebook .Rmd file for this lab from Blackboard and save it in your "Lab4" folder. There are no datasets used in this lab.

Within RStudio, navigate to your "Lab4" folder via the file browser in the lower right pane and then click "More > Set as working directory". Get set to write your observations and R commands in an R Markdown file by opening the "lab4_notebook.Rmd" file in RStudio. Remember to add your name as the author in line 3 of the document. For this lab, enter all of your commands into code chunks in the lab notebook. You can still experiment with code in an R script, if you want. To set up an R Script in RStudio, in the upper left corner click “File > New File > R script”. A new tab should open up in the upper left pane of RStudio.

Random Samples: In this class, we model random samples using random variables. Formally, when we say that $X_1, \ldots, X_n$ is a ***random sample*** from a population we are saying that each X_i is a random variable (in the probability sense from MATH 345 last spring semester) with probability distribution given by the probability distribution of the population it came from. Furthermore, we assume that the random variables in the sample are *independent*.

If we are interested in a population with unknown mean μ and standard deviation σ, and we obtain a random sample $X_1, \ldots, X_n$ from this population, then each of the X_i's have mean μ and standard deviation σ. So we can write

$$\mathrm{E}[X_i] = \mu \quad \text{and} \quad \mathrm{Var}(X_i) = \sigma^2, \quad \text{for } i = 1, \ldots, n.$$

Statistics/Estimators: In order to estimate population parameters like μ and σ, we use *functions* of random samples, which we refer to as ***statistics*** (or ***estimators***). For example, we use the ***sample mean*** $\bar{X}$ to estimate the population mean μ. For a random sample $X_1, \ldots, X_n$, the sample mean is given by the following function of the random sample:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i.$$

Sampling Distributions: Since statistics are functions of random variables, they are also random variables themselves and as such have probability distributions, which we refer to as ***sampling distributions***.

In Lab 3, you explored three properties of estimators, each of which has to do with a property of the sampling distribution:

- *Unbiased*: center/mean of sampling distribution equals true parameter value, i.e., $\mathrm{E}[\hat{\theta}] = \theta$
- *Efficient*: (only for unbiased estimators) small variability/spread in sampling distribution, i.e., $\mathrm{Var}(\hat{\theta})$ is small
- *MSE*: combines variance and bias, where bias is given by the difference between the mean of the estimator and the parameter it is estimating, i.e., $\mathrm{Bias}(\hat{\theta}) = \mathrm{E}[\hat{\theta}] - \theta$

In the Week 3 Homework Assignment, you were asked to show that the sample mean is *always* an unbiased estimator for the population mean, i.e., $\mathrm{E}[\bar{X}] = \mu$. This fact follows from the linear properties of expectation that we learned last spring in probability.

Pause for Reflection #1

Suppose that $X_1, \ldots, X_n$ is a random sample from a population with unknown variance σ^2, which means that $\mathrm{Var}(X_i) = \sigma^2$, for each $i = 1, \ldots, n$. Using properties of variance that you learned in probability, show that the variance of the sample mean is $\frac{\sigma^2}{n}$, i.e.,

$$\mathrm{Var}(\bar{X}) = \mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) = \frac{\sigma^2}{n}.$$

Upload a photo of your work to RStudio and include it in your lab notebook. (Make sure to save the image file in the "Lab4" folder, i.e., the same folder your lab notebook is saved in.)

There are essentially three approaches for finding sampling distributions of statistics:

1. *Exact calculation* by exhaustive calculation (like you did in the "Ideal Age" example in Lab 2) or formulas (which you will explore in this lab)
2. *Simulation* (like you did in Lab 3 when exploring properties of estimators)
3. *Formula approximations* (which you will explore in this lab)

The "Ideal Age" example in Lab 2 was small enough to calculate the exact *permutation distribution* (i.e., sampling distribution of test statistic used in the permutation test), but you approximated the flight delays permutation distribution using simulation. In some cases, we can obtain exact answers by formulas rather than exhaustive calculation. We have already seen one such example last spring in probability, in the case when sampling from a normal population.

Sampling Distribution of Sample Mean from Normal Population

If $X_1, \ldots, X_n$ are a random sample from a $N(\mu, \sigma)$ population, then the sample mean $\bar{X}$ is normally distributed with mean μ and standard deviation $\sigma/\sqrt{n}$.

We will explore this fact with simulation, but first a brief detour to explore how we can assess whether or not a random sample does appear to come from a normally distributed population.

Normally Distributed Data: First, let's look at data that is genuinely normally distributed. R has a nice function called `rnorm()` that produces pseudo-random samples from a normal distribution. For example, to generate a random sample of size $n = 10$ from a

standard normal distribution, enter the following into a code chunk in your lab notebook:

```
data = rnorm(10, 0, 1)          #rnorm(size, mean, sd)
data
```

It's useful to look at this random data in a histogram form:

```
hist(data)
```

Pause for Reflection #2

Your data may or may not look particularly like a bell curve. Comment on why this data set might *not* look like a bell curve, even though you presumably selected it from a $N(0,1)$ distribution.

Let's increase the sample size. Modify the `rnorm` command to produce a random sample of size $n = 1000$ from a $N(0,1)$ distribution and form a histogram of the result.

```
data = rnorm(1000, 0, 1)
hist(data)
```

Pause for Reflection #3

Does this histogram look more like a bell curve? Explain this phenomenon by writing out in your own words what it means to say that "this data comes from a normal distribution with mean 0 and standard deviation 1."

Recall the definition of ***percentiles*** from Lab 1: The $100p$th percentile π_p is the number such that $100p\%$ of values fall below π_p. For example, the 50th percentile for a $N(0,1)$ distribution is $\pi_{0.5} = 0$, since half of the distribution fall below the mean, which is equal to the median in this case.

We can use the function `qnorm()` to find any percentile we wish for a normal distribution. For example, enter the following in a code chunk in your lab notebook to find the 10th percentile for a $N(0,1)$ distribution:

```
qnorm(0.1, 0, 1)          #qnorm(p, mean, sd)
```

If we have a random sample of n data points, we can *estimate* percentiles of the population distribution by putting the data in order from smallest to largest. Then the kth data point, for $k = 1, \ldots, n$, estimates the percentile of order $p = \frac{k}{n+1}$.

Pause for Reflection #4

Using the above logic, the percentile estimates for the simple data set {1, 3, 4, 7} are as follows:

	data point	percentile
	1	20
	3	40
	4	60
	7	80

Convince yourself that this is correct. Additionally, in your lab notebook, comment on whether the numbers in the right-hand column have any bearing on the actual values of the data.

Normal Quantile Plots: To determine whether it is valid to assume that a random sample came from a normally distributed population, we compare the estimated percentiles from the sample to the corresponding percentiles of a standard normal distribution. To make the comparison, we construct a ***normal quantile plot*** by graphing the pairs of actual percentiles and data points. This is done in R using the `qqnorm()` function.

```
qqnorm(data)     #constructs quantile aka percentile plot
```

If the pairs exhibit a linear relationship, i.e., approximately lie on a straight line, then we conclude that the sample supports the assumption of normality for the population. You can add a reference line to the normal quantile plot using `qqline()` to help judge whether or not a linear relationship exists.

```
qqline(data)     #adds reference line through 1st & 3rd quartiles
```

So, in other words, if a data set is roughly normal, we expect the data percentiles and the distribution percentiles to be similar, and the resulting plot will be a straight line. Let's check that this idea works by forming another random sample, this time from a $N(85, 7)$ distribution, and then form a normal quantile plot.

```
data2 = rnorm(100, 85, 7)
qqnorm(data2); qqline(data2)     #construct quantile plot & add reference line
```

Your plot should look pretty much like a line, with small deviations, perhaps. Now let's look at a normal quantile plot for a data set we know comes from a *non-normal* distribution.

Let's simulate drawing a random sample from an exponential distribution with mean 15. Recall that the mean of an exponential distribution with parameter λ is given by $\frac{1}{\lambda}$. Thus, in this example $\lambda = \frac{1}{15}$. Figure 1 below shows a graph of the pdf.

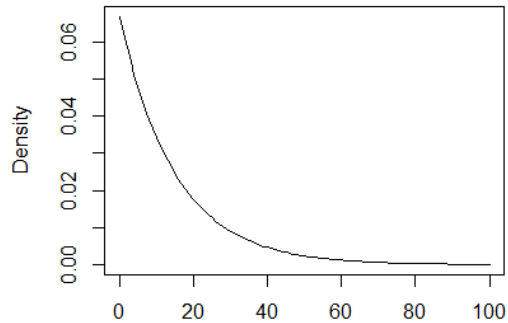


Figure 1: pdf of exponential(1/15)

We can use the `rexp()` function in R to draw a random sample from an exponential distribution.

```
data.exp = rexp(100, rate = 1/15)     #draw random sample from exponential(lambda=1/15
qqnorm(data.exp); qqline(data.exp)
```

Pause for Reflection #5

Why does the normal quantile plot for the exponential data indicate that it is not normal? Support this conclusion by looking at a histogram of `data.exp`.

Sample Mean from Normal Population: Let's now return to exploring the fact that when sampling from a normally distributed population, the sample mean will also be normally distributed. The following code selects 1000 random samples of size $n = 100$ from a $N(85, 7)$ distribution, computes the mean of each sample, and stores this mean in the vector `Xbar`. It has been provided in your lab notebook.

```
Xbar = numeric(1000)
for (i in 1:1000)
{
  x = rnorm(100, 85, 7)
  Xbar[i] = mean(x)
}
hist(Xbar)
qqnorm(Xbar); qqline(Xbar)
```

Pause for Reflection #6

See how close the simulation-based mean and standard deviation of the sampling distribution for the sample mean are to what the above fact claims they are. In other words, compare the simulated values

```
mean(Xbar)
sd(Xbar)
```

to the theoretical values (which you need to compute)

$$\mathbf{E}[\bar{X}] = \mu \qquad \text{and} \quad \mathbf{SD}(\bar{X}) = \sqrt{\frac{\sigma^2}{n}} = \frac{\sigma}{\sqrt{n}}.$$

Note that the simulated data were random samples of size 100 drawn from a normal distribution with mean 85 and standard deviation 7. Record the results in your lab notebook.

Sample Mean from Non-Normal Population: It turns out that even if the distribution the random samples are taken from is *not* normal, the sampling distribution of the sample mean is still *approximately* normal.

To demonstrate this, let's simulate the sampling distribution for the sample mean of random samples from an exponential distribution with mean 15. We can simulate the sampling distribution of a sample mean from this exponential distribution in the same way as we did above for the normal distribution. The following code has been provided in your lab notebook.

```
Xbar.exp = numeric(1000)
for (i in 1:1000)
{
  x = rexp(100, rate = 1/15)
  Xbar.exp[i] = mean(x)
}
hist(Xbar.exp)
qqnorm(Xbar.exp); qqline(Xbar.exp)
mean(Xbar.exp)
sd(Xbar.exp)
```

In contrast to the highly skewed distribution of the population (seen in Figure 1 above), the sampling distribution of $\bar{X}$ is nearly bell shaped, with the normal quantile plot only indicating a hint of skewness.

Pause for Reflection #7

We know that the mean of $\bar{X}$ should be equal to the mean of the population, which in this case we know to be 15. Does the mean obtained by your simulation approximate this reasonably well?

Note that for an exponential distribution the standard deviation is also given by $1/\lambda$. Using this, compare the estimated standard deviation of the sampling distribution for $\bar{X}$ to the theoretical standard deviation:

$$\frac{\sigma}{\sqrt{n}} = \frac{15}{\sqrt{100}} = 1.5.$$

Record the results of the above in your lab notebook.

Central Limit Theorem (CLT): The reason that the sampling distribution of the sample mean for random samples from non-normal distributions is approximately normal follows from the CLT.

Central Limit Theorem (CLT)

Let $X_1, \ldots, X_n$ be a random sample from a population with mean μ and variance σ^2. Then, for any constant $z \in \mathbb{R}$,

$$\lim_{n\to\infty} P\left(\frac{\bar{X}-\mu}{\sigma/\sqrt{n}} \le z\right) = \Phi(z),$$

where Φ denotes the cdf of the standard normal distribution.

The CLT means that for n "sufficiently large", the sampling distribution of $\bar{X}$ is approximately normal with mean μ and standard deviation $\sigma/\sqrt{n}$, regardless of the distribution from which the sample was drawn. Thus, the standardized random variable

$$Z = \frac{\bar{X}-\mu}{\sigma/\sqrt{n}} = \frac{\bar{X}-\mathrm{E}[\bar{X}]}{\mathrm{SD}(\bar{X})}$$

is approximately normal with mean 0 and standard deviation 1.

Pause for Reflection #8

Return to the simulated sampling distribution of $\bar{X}$for a sample from an exponential population with mean 15. We can now use the CLT estimate the probability $P(\bar{X} > 18)$, as follows:

$$P(\bar{X} > 18) = P\left(\frac{\bar{X}-15}{1.5} > \frac{18-15}{1.5}\right) \approx P(Z > 2), \quad \text{where} Z \sim N(0,1).$$

In your lab notebook, explain how the CLT is being used in the above equation.

We can calculate normal probabilities in R using the function `pnorm()`. So, $P(Z > 2)$ is given by

```
pnorm(2, 0, 1, lower.tail=FALSE)
```

Compare this to the proportion of simulated sample means that were above 18 using the following:

```
sum(Xbar.exp > 18)/1000
```

Are the probability given by the CLT and the proportion from the simulation close?

Lab 4: Samping Distributions is shared under a not declared license and was authored, remixed, and/or curated by LibreTexts.

Lab 5: Confidence Intervals

Objectives:

1. Find confidence intervals for μ.
2. Explore the t distribution.

Definitions:

- point estimate vs. interval estimate
- confidence intervals
- confidence level
- *t* distribution

Introduction:

In the past few weeks, we have learned how to find *point estimates* for population parameters, in other words, single value estimates of an unknown parameter. But these point estimates are based on random samples and so are inherently variable and uncertain. We can model that uncertainty with the sampling distribution of a statistic, and last week we focused on the sampling distribution of the sample mean. This week, we use that sampling distribution to construct *interval estimates* for population parameters. Interval estimates give a range of plausible values for a parameter based on a random sample, and incorporate the variability of point estimates.

Activities:

Getting Organized: *If you are already organized, and remember the basic protocol from previous labs, you can skip this section.*

Navigate to your class folder structure. Within your "Labs" folder make a subfolder called "Lab5". Next, download the lab notebook .Rmd file for this lab from Blackboard and save it in your "Lab5" folder. You will be working with the `NCBirths2004` dataset on this lab. You should download the data file into your "Lab5" folder from Blackboard.

Within RStudio, navigate to your "Lab5" folder via the file browser in the lower right pane and then click "More > Set as working directory". Get set to write your observations and R commands in an R Markdown file by opening the "lab5_notebook.Rmd" file in RStudio. Remember to add your name as the author in line 3 of the document. For this lab, enter all of your commands into code chunks in the lab notebook. You can still experiment with code in an R script, if you want. To set up an R Script in RStudio, in the upper left corner click “File > New File > R script”. A new tab should open up in the upper left pane of RStudio.

Confidence Intervals for a Mean, Variance Known: At the end of class on Tuesday, we looked at estimating the mean birth weight for girls born in South Bend. We assumed that the population of birth weights was normally distributed with unknown mean μ, but known standard deviation $\sigma = 1.1$. In that case, the sampling distribution of a sample mean birth weight for a random sample of size $n = 100$ is normal with mean μ and standard deviation $1.1/\sqrt{100}$. Given this, we then derived the following:

$$0.95 = P(-1.96 < Z < 1.96) = P\left(-1.96 < \frac{\bar{X} - \mu}{1.1/\sqrt{100}} < 1.96\right) = P\left(\bar{X} - 1.96\frac{1.1}{\sqrt{100}} < \mu < \bar{X} + 1.96\frac{1.1}{\sqrt{100}}\right)$$

The *random* interval given by

$$\left(\bar{X} - 1.96\frac{1.1}{\sqrt{100}},\ \bar{X} + 1.96\frac{1.1}{\sqrt{100}}\right) \quad (1)$$

has a probability of 0.95 of containing the true value of the mean μ. Now, once the public health officials in South Bend have drawn a random sample, the random variable $\bar{X}$ is replaced by the (observed) sample mean birth weight of $\bar{x} = 7.1$ lb, giving the specific interval

$$\left(7.1 - 1.96\frac{1.1}{\sqrt{100}},\ 7.1 + 1.96\frac{1.1}{\sqrt{100}}\right) \quad \Rightarrow \quad (6.884, 7.316), \quad (2)$$

which is no longer random. We interpret this interval by saying that we are 95% *confident* that the population mean birth weight of girls born in South Bend is between 6.9 and 7.3 lb. In other words, if we repeated the same process of drawing samples and computing intervals many times, then in the long run, 95% of the intervals would include μ.

Pause for Reflection #1:

Explain in your own words why the interval in equation (1) is random, but the interval in equation (2) is not.

In general, if a random sample of size n is drawn from a normal distribution with unknown mean μ and known standard deviation σ, then a 95% *confidence interval* for μ is

$$\left(\bar{X} - 1.96\frac{\sigma}{\sqrt{n}},\ \bar{X} + 1.96\frac{\sigma}{\sqrt{n}}\right) \quad (3)$$

If we repeatedly draw random samples from the population and compute the above 95% confidence interval for each sample, then we expect about 95 of those intervals will contain the actual value of μ.

We can demonstrate the interpretation of confidence intervals with a simulation. The following code simulates drawing random samples of size 30 from a $N(25, 4)$ distribution. For each sample, we construct the 95% confidence interval and check to see if it contains the population mean, $\mu = 25$. We do this 1000 times and keep track of the number of times the interval contains μ using a `counter`. We can also visualize the first 100 intervals computed.

```
counter = 0                                          # set counter to 0
plot(x = c(22, 28), y = c(1, 100), type = "n",       # set up a blank plot
     xlab = "", ylab = "")                           # with no axis labels
for (i in 1:1000)
{
  x = rnorm(30, 25, 4)                               # draw random sample of size 30
  L = mean(x) - 1.96*4/sqrt(30)                      # lower endpoint of interval
  U = mean(x) + 1.96*4/sqrt(30)                      # upper endpoint of interval
  if (L < 25 && 25 < U)                              # check if 25 is in interval
    counter = counter + 1                            # if yes, increase counter by 1
  if (i <= 100)                                      # plot first 100 intervals
    segments(L, i, U, i)
}
abline(v = 25, col = "red")                          # vertical line at mu
counter/1000                                         # proportion of times mu in interval
```

Pause for Reflection #2:

In your lab notebook, run the above simulation (code provided) and comment on the proportion of times the intervals in the simulation contain $\mu = 25$. Is it close to 95%?

Explain in your own words what the plot produced by the simulation demonstrates.

In the first example with birth weights and the above simulation, we constructed 95% confidence intervals. The formula in equation (3) was derived starting from the fact that for the standard normal random variable Z, we have

$$0.95 = P(-1.96 < Z < 1.96).$$

This fact can be confirmed using the `qnorm()` function in R, which calculates quantiles for the normal distribution given a probability. With 0.95 probability in the middle, that leaves 0.05/2 = 0.025 probability in each of the two tails, as Figure 1 demonstrates.

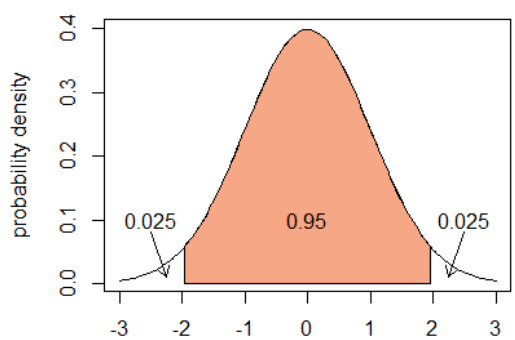


Figure 1: Standard normal density with shaded area 0.95

In this case, we also have that $0.025 = P(Z > 1.96)$, which is found in R as follows:

```
qnorm(0.025, lower.tail = FALSE)      # returns the value q such that P(Z > q) = 0.025
```

Pause for Reflection #3:

If we want 0.93 probability in the middle of a standard normal density curve, how much probability does that leave in each of the two tail regions? Sketch a figure similar to Figure 1, but with 0.93 middle probability and upload the image to your lab notebook. Use the `qnorm()` function to find the value of q satisfying $0.93 = P(-q < Z < q)$.

Pause for Reflection #4:

Redo the simulation above, but find 93% confidence intervals this time. You will need to use the value of q you found in Reflection #3 and alter in some way the following two lines in the `for` loop of the simulation:

```
L = mean(x) - 1.96*4/sqrt(30)                    # lower endpoint of interval
U = mean(x) + 1.96*4/sqrt(30)                    # upper endpoint of interval
```

What is the proportion of times the intervals in the simulation contain $\mu = 25$ equal to now? Is it what you expected? Explain.

In general, we let $z_{\alpha/2}$ denote the $(1-\alpha/2)$ quantile for the standard normal distribution. In other words, $z_{\alpha/2}$ is the value such that $P(Z > z_{\alpha/2}) = \alpha/2$. By symmetry then, the middle probability given by $(1-\alpha)$ falls between $-z_{\alpha/2}$ and $z_{\alpha/2}$. (See Figure 2 below.) For example, in a 95% confidence interval, $\alpha = 0.05$ and $z_{\alpha/2} = z_{0.025} = 1.96$.

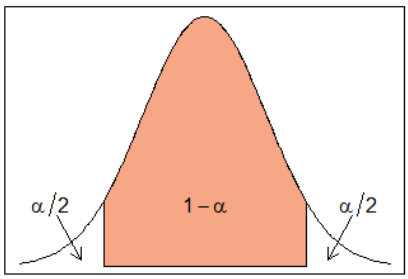


Figure 2: Standard normal density with shaded area (1-alpha)

We can now give a general formula for a $100(1-\alpha)$% confidence interval of μ, when σ is know.

Z Confidence Interval for a Normal Mean with Known Standard Deviation

If $X_i \sim N(\mu, \sigma)$, for $i = 1, \ldots, n$, with σ known, then a $100(1-\alpha)$% ***confidence interval*** *for* μ is given by

$$\left(\bar{X} - z_{\alpha/2}\frac{\sigma}{\sqrt{n}},\ \bar{X} + z_{\alpha/2}\frac{\sigma}{\sqrt{n}}\right)$$

Pause for Reflection #5:

Return to the first example and find a 90% confidence interval for the mean birth weight of girls born in South Bend. How does this interval compare to the 95% confidence interval we found? Which interval is wider?

Confidence Intervals for a Mean, Variance Unknown: In practice, we will not know either the mean or the standard deviation of the population we are interested in. As we have seen in previous weeks, if we want to know the value of a population parameter, we can use a statistic computed from a random sample to estimate it. As we use $\bar{X}$ to estimate μ, we can use S, the ***sample standard deviation***, to estimate σ. This leads to the question: does replacing σ with S in the following change the distribution?

$$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \sim N(0,1) \quad \Rightarrow \quad T = \frac{\bar{X} - \mu}{S/\sqrt{n}} \overset{?}{\sim} N(0,1)$$

It turns out that estimating σ with S does indeed change the sampling distribution. As we have done many times already, we explore this question with a simulation.

```
N = 10^4
z = numeric(N)
t = numeric(N)
n = 15                                # sample size
for (i in 1:N)
{
  x = rnorm(n, 25, 4)                 # draw 15 numbers from N(25, 4)
  Xbar = mean(x)                      # calculate sample mean
  S = sd(x)                           # calculate sample sd
  z[i] = (Xbar - 25) / (4/sqrt(n))    # standardize sample mean using sigma
  t[i] = (Xbar - 25) / (S/sqrt(n))    # standardize sample mean using sample sd
}
hist(z)
hist(t)
qqnorm(z); qqline(z)                  # assess normality for z
qqnorm(t); qqline(t)                  # assess normality for
```

Pause for Reflection #6:

In your lab notebook, run the above simulation (code provided). Explain how the results of the simulation show that the distribution of $T = (\bar{X} - \mu)/(S/\sqrt{n})$ is *not* normally distributed.

The distribution of $T = (\bar{X} - \mu)/(S/\sqrt{n})$ is actually a ***Student's t distribution*** *with* $n - 1$ *degrees of freedom*, provided that the population is normally distributed. The pdf of t distribution with k degrees of freedom is bell-shaped and symmetric about 0, like the standard normal pdf. But, unlike the standard normal pdf, it has *thicker* tails. As k tends to infinity, the pdf of the t distribution approaches the standard normal pdf. Figure 3 below shows the pdf's for the standard normal distribution and three t distributions.

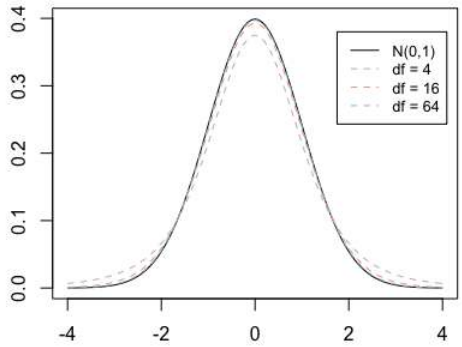


Figure 3: Comparison of pdf's for standard normal and t distributions

We derive the confidence interval for μ when σ is unknown in the same way as when σ is know, except we use the t distribution to find quantiles. We let $t_{\alpha/2,n-1}$ denote the $(1-\alpha)$ quantile for a t distribution with $n-1$ degrees of freedom, i.e., the value such that

$$P(T > t_{\alpha/2,n-1}) = \alpha/2,$$

where T has a t distribution with $n-1$ degrees of freedom. The quantiles $t_{\alpha/2,n-1}$ replace the standard normal quantiles $z_{\alpha/2}$ in the formula, and we arrive at the following.

T Confidence Interval for a Normal Mean with Unknown Standard Deviation

If $X_i \sim N(\mu, \sigma)$, for $i = 1, \ldots, n$, with σ known, then a $100(1-\alpha)$ *confidence interval for* μ is given by

$$\left(\bar{X} - t_{\alpha/2,n-1}\frac{S}{\sqrt{n}},\ \bar{X} + t_{\alpha/2,n-1}\frac{S}{\sqrt{n}}\right)$$

The functions `pt()` and `qt()` give probabilities and quantiles, respectively, for the t distribution. For example, to find $P(T < 2.8)$ for the random variable T from a t distribution with 27 degrees of freedom, try the following:

```
pt(2.8, 27)
```

And to find the quantile $t_{0.05,27}$, try

```
qt(0.05, 27, lower.tail = FALSE)
```

Pause for Reflection #7:

Compare the quantile $t_{0.05,27}$ to the corresponding standard normal quantile $z_{0.05}$. Which one is larger? Can you explain why? What effect on the width of a 90% confidence interval does using the t distribution quantile have? Can you explain why this makes sense given that confidence intervals based on the t distribution are used when σ is unknown?

Pause for Reflection #8:

Suppose that the public health officials in South Bend are also interested the mean birth weight of boys in their city. They are willing to again suppose that the distribution of boys weights in South Bend is normal, but they do not want to assume a value for the standard deviation. Instead, they obtain a random sample of 28 boys, resulting a sample mean of 7.6 lb and a sample standard deviation of 1.3 lb. Use their results to construct a 90% confidence interval for the true mean birth weight of boys born in South Bend. Upload an image of any hand-written work.

If you have the full data set of observations in a random sample available in R, then the function `t.test()` can calculate confidence intervals quickly. We will demonstrate this with the `NCBirths2004` data set, which contains information on a random sample of 1009 babies born in North Carolina during 2004, and construct a 99% confidence interval for the mean birth weight (in grams) of girls born in North Carolina.

```
NCBirths2004 = read.csv("NCBirths2004.csv")
girls = subset(NCBirths2004, select = Weight, subset = Gender == "Female", drop = TRUE
t.test(girls, conf.level = 0.99)$conf

## [1] 3343.305 3453.328
## attr(,"conf.level")
## [1] 0.99
```

Thus, the 99% confidence interval for the mean birth weight of girls born in North Carolina in 2004 is (3343.3, 3453.3) g.

Pause for Reflection #9:

Alter the `t.test()` function and find 95% and 90% confidence intervals for the mean birth weight of girls born in North Carolina in 2004. How do these intervals compare to each other and to the 99% confidence interval? Explain what the effect of decreasing the confidence level (i.e., going from 99% to 90%) has on the width of the confidence interval.

Pause for Reflection #10:

Note that in order to use the t distribution to construct confidence intervals, the population must be normally distributed. Assess whether or not the population of birth weights for babies born in North Carolina in 2004 is normally distributed.

Lab 5: Confidence Intervals is shared under a not declared license and was authored, remixed, and/or curated by LibreTexts.

Lab 6: More Hypothesis Testing - Classical Approach

Objectives:

1. Understand how to perform hypothesis tests for means (one population and two populations) using the classical approach.
2. Understand how to use the `t.test()` function in R to calculate P-values

Definitions:

- hypothesis testing
- null vs. alternative hypothesis
- left- vs. right- vs. two-tail test
- test statistic
- P-value
- statistical significance
- t-test
- matched pairs vs. independent samples

Introduction:

In Lab 2, we introduced hypothesis testing, a formal procedure for testing the validity of a claim about a population or populations. Specifically, we developed a procedure called *permutation testing* in the context of testing hypotheses about two population means. Permutation testing does not make any assumptions about the distributions of the populations involved in the hypotheses. In this lab, we consider the classical approach to hypothesis testing, where now we will make assumptions about the distribution of the population or at least use a probability distribution to compute (approximate) P-values. This builds on the work we did in Labs 4 & 5, where we used either the standard normal or t distributions.

We will still use the same framework for performing a hypothesis test that we established in Lab 2. Namely, we compute a test statistic from the data and then a corresponding P-value that tells us the probability of getting a value as extreme as or more extreme than the observed test statistic *assuming the null hypothesis is true*. The smaller the P-value, the more evidence we have against the null hypothesis, because the observed result cannot be easily explained by chance alone.

Activities:

Getting Organized: *If you are already organized, and remember the basic protocol from previous labs, you can skip this section.*

Navigate to your class folder structure. Within your "Labs" folder make a subfolder called "Lab6". Next, download the lab notebook .Rmd file for this lab from Blackboard and save it in your "Lab6" folder. You will be working with the `SAT` and `NCBirths2004` data sets on this lab. You should download the data files into your "Lab6" folder from Blackboard.

Within RStudio, navigate to your "Lab6" folder via the file browser in the lower right pane and then click "More > Set as working directory". Get set to write your observations and R commands in an R Markdown file by opening the "lab6_notebook.Rmd" file in RStudio. Remember to add your name as the author in line 3 of the document. For this lab, enter all of your commands into code chunks in the lab notebook. You can still experiment with code in an R script, if you want. To set up an R Script in RStudio, in the upper left corner click “File > New File > R script”. A new tab should open up in the upper left pane of RStudio.

Pause for Reflection #1:

In Exercise #1 from Tuesday's class, we tested the claim that the mean dissolved oxygen content μ in a certain stream is less than 5 mg per liter based on a sample of 45 specimens with a mean of 4.62 mg/l. We assumed that dissolved oxygen content varies among locations in the stream according to a normal distribution with standard deviation $\sigma = 0.92$ mg, and so calculated the P-value as follows:

$$P\text{-value} = P(\bar{X} \leq 4.62 \mid \mu = 5) = P\left(\frac{\bar{X} - 5}{0.92/\sqrt{45}} \leq \frac{4.62 - 5}{0.92/\sqrt{45}}\right) = P(Z \leq -2.77) = 0.0028$$

Comment on why we calculated the probability that a sample mean $\bar{X}$ would be *"less than or equal to"* the observed sample mean 4.62, instead of simply *"equal to"* or *"greater than or equal to"*. Next, explain why we subtract 5 and divide by $0.92/\sqrt{45}$ in the middle probability expression.

One-Sided Tests: The claim we tested in Exercise #1 on Tuesday claimed that the actual population mean was *less* than a specific number. In particular, the alternative hypothesis for Exercise #1 was $H_A : \mu < 5$ mg/l. Testing a claim of "less than" is a *one-sided test*, specifically referred to as a ***left-tail test***. We now consider an example of a claim of "greater than", i.e., a ***right-tail test***.

SAT Example: We suspect that on average students will score higher on their second attempt at the SAT math exam than on their first attempt. The data set `SAT` gives the changes in score (second try minus first try) results for 46 randomly chosen high school students. We will perform a hypothesis test to see if these data provide good evidence that the mean change in the population is greater than zero.

Pause for Reflection #2:

For the SAT example just introduced, state the null and alternative hypotheses being tested using appropriate parameter notation.

We now load the SAT data and compute the sample mean and standard deviation:

```
SAT = read.csv("SAT.csv")
xbar = mean(SAT$SAT.change)
s = sd(SAT$SAT.change)
```

If we assume that changes in SAT scores are normally distributed, then the test statistic

$$T = \frac{\bar{X} - \mu}{S/\sqrt{46}}, \quad (1)$$

has a t distribution with 45 degrees of freedom, since the population standard deviation is unknown.

Pause for Reflection #3:

Calculate the P-value associated with the observed SAT results and clearly state your conclusion for testing the hypotheses you stated in Reflection #2. Remember that the P-value is the probability of getting a value as extreme as or more extreme than the observed test statistic assuming the null hypothesis is true.

Pause for Reflection #4:

Does the sample of changes in SAT scores support the assumption that the population is normally distributed?

Pause for Reflection #5:

Comment on why you think the terminology *left-tail test* and *right-tail test* are used to describe one-sided tests. Specifically, why is testing a claim of "less than" done with a left-tail test, and a claim of "greater than" with a right-tail test?

Two-Sided Tests: Suppose that we were not sure whether on average students score higher or lower on their second attempt, and instead we just want to test the claim that on average the scores are not the same. In this case, we are testing a claim of either "less than" or "greater than". Simply put, we are testing a claim of "difference", which includes both cases. This type of test is referred to as a ***two-sided test*** or a ***two-tail test***.

SAT Example: For the change in SAT scores, the null hypothesis remains the same, but the alternative is now stated simply as the mean change in SAT scores for all students μ is not equal to zero:

$$H_0 : \mu = 0 \quad \text{vs.} \quad H_A : \mu \neq 0.$$

We use the same test statistic given in equation (1), but to calculate the P-value in a two-sided test we consider both less than and greater than values as extreme, which is accomplished using absolute values:

$$P\text{-value} = P(|T| \geq |t|) = P(T \leq -|t|) + P(T \geq |t|),$$

where t denotes the observed value of the test statistic.

Pause for Reflection #6:

Calculate the P-value for the two-sided test. How does it compare to the P-value you calculated in Reflection #3 for the one-sided, right-tail test?

The T-Test in General: Notice the general procedure we have followed in the water quality example and the change in SAT scores example:

1. state the null and alternative hypotheses,
2. calculate a test statistic from observed data,
3. find or estimate a sampling distribution for the test statistic, assuming the null hypothesis is true,
4. calculate a P-value using that distribution,
5. and finally state a conclusion of the test, rejecting H_0 if P-value is small.

In permutation testing, the sampling distribution found in step 3 is given by the permutation distribution obtained by permuting the data. In the classical approach taken in the above examples, the sampling distributions are *parametric*, normal or t distributions.

We note that the second example is more likely, i.e., it is more likely that the population standard deviation is unknown and so the P-value will be calculated using a t distribution. The following summarizes the classical approach for testing a claim about a population mean when the population standard deviation is unknown, known as a *t-test*.

T-Test for a Normal Mean

Let $X_1, \ldots, X_n$ be a random sample from a normal population with unknown μ and σ. Let $\bar{X}$ and S denote the sample mean and standard deviation. For a null hypothesis given by

$$H_0 : \mu = \mu_0,$$

where μ_0 is a constant, we form the t-test statistic

$$T = \frac{\bar{X} - \mu_0}{S/\sqrt{n}}.$$

If the null hypothesis is true, then T has a t distribution with $(n-1)$ degrees of freedom. The P-value is the probability that chance alone would produce a test statistic as extreme as or more extreme than the observed value of the test statistic $t = (\bar{x} - \mu_0)/(s/\sqrt{n})$, if the null hypothesis is true. What is considered "extreme" depends on the alternative hypothesis: left-tail, right-tail, or two-tail.

Note that the *t*-test is *exact* when the sample comes for a normally distributed population. If the population is not normal, then the *t*-test provides an approximate P-value, since by the Central Limit Theorem the sample mean $\bar{X}$ will be approximately normal even though the population is not. So, in practice, the *t*-test is used to compute approximate P-values when the sample size is large and the sample is not too skewed. If the sample size is not large and/or the sample is severely skewed, then a permutation test should be used.

Using R to do all the calculations: The `t.test()` function in R performs the *t-test* for a normal mean, provided that we have all sample data and not just sample statistics (i.e., sample mean and standard deviation values). For the change in SAT scores example, the following performs the two-sided test:

```
t.test(SAT$SAT.change)
```

The default settings of `t.test()` are to perform a two-sided test that the population mean is zero. But the type of test can be changed by adding the argument `alternative = "less"` for a left-tail test or `alternative = "greater"` for a right-tail test. So, for the one-sided test that the change in SAT scores is greater than zero, try the following and compare to what you found in Reflection #3:

```
t.test(SAT$SAT.change, alternative = "greater")
```

We can also change the value of μ claimed under the null hypothesis. For example, suppose that in the past it was determined that on average students score 10 points higher on their second attempt at the SAT math test, and we want to test that the average change is now more than 10 points. In this case, the null hypothesis is now $H_0 : \mu = 10$, and we can test this with the data as follows:

```
t.test(SAT$SAT.change, alternative = "greater", mu = 10)
```

Pause for Reflection #7:

Comment on the effect that changing the null hypothesis to $H_0 : \mu = 10$ had on the resulting test statistic and P-value compared to the corresponding values you found in Reflection #3 for the test that $\mu = 0$. Explain why the test statistic and P-value changed in this way.

Tests Comparing Two Population Means: In Lab 2, we actually performed a permutation test for hypotheses concerning two populations, women and men. There is also a t-test for comparing two populations.

If μ_1 and μ_2 denote the two population means, $\bar{X}_1$ and $\bar{X}_2$ denote sample means for samples of size n_1 and n_2, respectively, taken from the two populations, and S_1 and S_2 denote the sample standard deviations, then the test statistic

$$T = \frac{(\bar{X}_1 - \bar{X}_2) - (\mu_1 - \mu_2)}{\sqrt{(S_1^2/n_1) + (S_2^2/n_2)}}$$

has approximately a t distribution. Notice that the test statistic involves *differences* in the means, and because of this hypotheses for comparing two population means are typically stated in terms of the difference in the means. Furthermore, the null hypothesis is almost always the claim that there is no difference, i.e., that $\mu_1 - \mu_2 = 0$. Let's look at an example.

Births Example: Consider again the `NCBirths2004` data set from Lab 5, which contains information on a random sample of babies born in North Carolina during 2004. In addition to the birth weights of the babies, the variable `Tobacco` in the data set indicates whether or not the mothers of the babies smoked during the pregnancy. Using this data, we can test the claim that smoking during pregnancy results in a lower birth weight than not smoking, on average.

Pause for Reflection #8:

Let μ_1 and μ_2 denote the true mean weight of babies born to smoking and nonsmoking mothers, respectively. Using the difference $\mu_1 - \mu_2$, state the null and alternative hypotheses to test the claim that smoking during pregnancy results in a lower birth weight than not smoking, on average.

We can use R/RStudio to do all of the necessary calculations for us, so that we do not have to work with equation (2) directly. The same `t.test()` function can be used to perform a two-sample test:

```
births = read.csv("NCBirths2004.csv")
smoker = subset(births, select = Weight, subset = Tobacco == "Yes", drop = TRUE)
nonsmoker = subset(births, select = Weight, subset = Tobacco == "No", drop = TRUE)
t.test(smoker, nonsmoker, alternative = "less")
```

Pause for Reflection #9:

Based on the P-value for the above two-sample t-test, is the result statistically significant? Write a conclusion to the test of the hypotheses stated in Reflection #8.

Look at the output from the `t.test()` and locate the `sample estimates`. What is the mean birth weight for babies born to smoking mothers and what is the mean for nonsmoking mothers?

Assumptions of the Two-Sample T-Test: As with the one-sample t-test, we need to check that the two populations being compared in a two-sample t-test are normally distributed. If this assumption is violated, in particular, if the distributions are skewed and the sample sizes of the two samples are different, then the actual distribution of the test statistic given in equation (2) may differ substantially from the t distribution. In that case, a permutation test would be more reliable.

Pause for Reflection #10:

Check that the populations of birth weights for babies born to smokers and babies born to nonsmokers both appear to be normally distributed.

Matched Pairs: The two-sample t-test also requires that the two samples be from two *independent* populations. If the two samples are ***paired*** (not independent), then we need to perform a ***paired t-test***, which is done with the same `t.test()` function by adding the argument `paired = TRUE`. The change in SAT scores example can be performed as a paired t-test, instead of a one-sample t-test as we did previously.

SAT Example: Take a closer look at the `SAT` data set:

```
View(SAT)
summary(SAT)
```

Notice that in addition to the variable `SAT.change`, which contains the differences between the second attempt and first attempt for each student, we also have the actual scores on the second attempt in variable `SAT.2` and a list of scores on the first attempt in variable `SAT.1` for each student.

In this case, the two samples of scores for the first and second attempts are paired, consisting of pairs of SAT scores for the random sample of students. In other words, the rows in the `SAT` data set correspond to a single student and so the values are *related*, not independent. Thus, if we set up the test of the claim that on average students will score higher on their second attempt at the SAT math exam than on their first attempt by comparing the two samples, we need to perform a paired t-test.

Let μ_1 denote the true mean of SAT scores for the first attempt and let μ_2 denote the true mean for the second attempt. Following the order we took the difference in earlier - second attempt minus first - we state the hypotheses being tested as follows:

$$H_0 : \mu_2 - \mu_1 = 0 \quad \text{vs.} \quad H_A : \mu_2 - \mu_1 > 0.$$

We can use the `t.test()` function to calculate the test statistic and associated P-value needed to perform the test:

```
t.test(SAT$SAT.2, SAT$SAT.1, alternative = "greater", paired = TRUE)
```

Notice the order that the samples of SAT scores are listed in the argument of `t.test()`, it matches the order we that we took the difference in population means when stating the hypotheses.

Pause for Reflection #11:

Compare the results of the paired t-test to the results you found in Reflection #3. How do the test statistics and corresponding P-values compare? Can you provide an explanation for why?

It is important to perform the paired t-test when samples are paired, because the results of the test are often very different from the results of the two-sample t-test. For example, omit the argument `paired = TRUE` from the `t.test()` we ran on the SAT scores:

```
t.test(SAT$SAT.2, SAT$SAT.1, alternative = "greater")
```

Pause for Reflection #12:

Now compare the results of the two-sample t-test you just ran to the results of the paired t-test. Again focus on the test statistics and corresponding P-values. What do you notice?

Lab 6: More Hypothesis Testing - Classical Approach is shared under a not declared license and was authored, remixed, and/or curated by LibreTexts.

Lab 7: ANOVA

Objectives:

1. Understand how to perform ANOVA F test for comparing three or more population means.
2. Understand how to use the `aov()` function in R to construct ANOVA tables.

Definitions:

- ANOVA (analysis of variance)
- treatment groups
- grand mean
- MSTR (mean sum of squares for treatment)
- MSE (mean sum of squares for error)
- F-distribution
- F statistic, F test

Introduction:

In Labs 2 and 6, we considered methods for testing claims about two population means, namely, permutation tests and t tests. In this lab, we consider a technique for testing claims about three or more population means known as *analysis of variance* (*ANOVA*). The ANOVA procedure compares the variation in the means of samples taken from the populations. The idea is to partition the variability in all the samples into the variability *between* each sample and the variability *within* each sample. If the population means are indeed equal, then the variability between and within each sample should be roughly the same. The ratio of the between and within variability provides a test statistic, and the classical approach, which we consider in this lab, uses a theoretical sampling distribution. In the next lab, we will develop a permutation test approach for performing ANOVA.

Activities:

Getting Organized: *If you are already organized, and remember the basic protocol from previous labs, you can skip this section.*

Navigate to your class folder structure. Within your "Labs" folder make a subfolder called "Lab7". Next, download the lab notebook .Rmd file for this lab from Blackboard and save it in your "Lab7" folder. There are no datasets used in this lab. You will be working with the `Zombies.csv` data set on this lab. You should download the data file into your "Lab7" folder from Blackboard.

Within RStudio, navigate to your "Lab7" folder via the file browser in the lower right pane and then click "More > Set as working directory". Get set to write your observations and R commands in an R Markdown file by opening the "lab7_notebook.Rmd" file in RStudio. Remember to add your name as the author in line 3 of the document. For this lab, enter all of your commands into code chunks in the lab notebook. You can still experiment with code in an R script, if you want. To set up an R Script in RStudio, in the upper left corner click “File > New File > R script”. A new tab should open up in the upper left pane of RStudio.

Notation: Before we see how to perform an ANOVA test in R/RStudio, let's formally set up the procedure, starting with defining the notation:

- G denotes the number of populations/samples
- n_i denotes the number of observations in the ith sample, $i = 1, \dots, G$
- $n = n_1 + \cdots + n_G$ denotes the total number of observations
- X_{ij} denotes the jth observation in the ith sample, $j = 1, \dots, n_i$
- $\bar{X}_{i.}$ denotes the mean of the ith sample
- $\bar{X}_{..}$ denotes the *grand mean*, i.e., the mean of all n observations in each sample

If we let μ_i denote the mean of the ith population, then we are testing the following hypotheses with the above sample data:

$$H_0: \mu_1 = \mu_2 = \cdots = \mu_G \quad \text{vs.} \quad H_A: \text{ at least one } \mu_i \text{ is different, i.e., } \mu_i \neq \mu_j \text{ for some } i \neq j$$

As discussed in the introduction above, we test these hypotheses by comparing the variability between the sample means to the variability within each sample. For the variability between the samples, we use the *mean sum of squares for treatment* (MSTR), which is given by

$$\text{MSTR} = \frac{1}{G-1}\sum_{i=1}^{G} n_i(\bar{X}_{i\cdot} - \bar{X}_{\cdot\cdot})^2.$$

For the variability within the samples, we use the *mean sum of squares for error* (MSE), which is given by

$$\text{MSE} = \frac{1}{n-G}\sum_{i=1}^{G}\sum_{j=1}^{n_i}(X_{ij} - \bar{X}_{i\cdot})^2.$$

Pause for Reflection #1:

Four chemical plants, producing the same products and owned by the same company, discharge liquid waste into streams in the vicinity of their locations. To monitor the extent of pollution created by the liquid waste and determine whether this differs from plant to plant, the company collected random samples of liquid waste from each plant, resulting in the following data.

Plant	Polluting Effluents (lb/gal of waste)	Sample Mean
A	1.65 1.72 1.50 1.37 1.60	1.568
B	1.70 1.85 1.46 2.05	1.765
C	1.40 1.75 1.38 1.65 1.55	1.546
D	2.10 1.95 1.65 1.88	1.895

- State the hypotheses we will test to determine if there is a difference in the mean weight of polluting effluents per gallon in the liquid waste discharged from the four plants. Be sure to define your notation.
- Identify what the values of G and n are, and for each sample identify the values of n_i and $\bar{X}_{i\cdot}$ are.
- Finally, find the values of the grand mean $\bar{X}_{\cdot\cdot}$ and the variability between the samples' MSTR.

The ANOVA F Test: If H_0 is true, i.e., the population means are all equal, then the variability between the samples should be roughly the same as the variability within the samples (assuming also that the populations have equal variance). If H_0 is false, then the variability between the samples will be larger than the variability within the samples. Thus, we use the ratio of the between and within variability measures as the test statistic,

$$F = \frac{\text{MSTR}}{\text{MSE}},$$

which has a F distribution with $(G-1)$ and $(n-G)$ df. The observed test statistic based on the sample data obtained is denoted f, and then its associated P-value is calculated using the F distribution as follows:

$$P\text{-value} = P(F \geq f)$$

Note that the P-value is given by the probability of obtaining a test statistic as large *or larger* than what was observed, i.e., the P-value for an ANOVA F test is always a right-tail probability. This is because "more extreme" in this context would be sample data that produced *more* between sample variability resulting in a larger ratio of MSTR to MSE.

Pause for Reflection #2:

Return to the pollution example and compute the observed F statistic using the value of MSTR you found in Reflection #1 and given that MSE = 0.03336. Then use the following code (with the corresponding values of `f`, `G-1`, and `n-G` substituted in) to calculate the corresponding P-value:

```
pf(f, G-1, n-G, lower.tail = FALSE)
```

Based on the P-value, do the data provide sufficient evidence to indicate a difference in the mean weight of polluting effluents per gallon in the liquid waste discharged from the four plants?

The ANOVA Table: As we can see, there are alot of calculations that go into performing ANOVA. The ANOVA table given below is a tool that summarizes and organizes these calculations in an easy to use format.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Factor	G-1	$\sum n_i(\bar{X}_{i.} - \bar{X}_{..})^2$	MSTR	MSTR/MSE	P-value
Error	n-G	$\sum\sum(X_{ij} - \bar{X}_{i.})^2$	MSE		
Total	n-1	$\sum\sum(X_{ij} - \bar{X}_{..})^2$			

Notice how the ANOVA table is arranged:

- The last two columns are the most useful, since they give the test statistic and its associated P-value:
  - the last column with heading `Pr(>F)` gives the P-value, so it is easy to read off;
  - the second to last column with heading `F value` gives the observed F statistic, which the P-value is based on.
- The other columns give the supporting calculations used to find the test statistic and P-value:
  - the first column provides labels for the source of variability, where `Factor` corresponds to the between samples variability and `Error` corresponds to the within sample variability;
  - the second column gives the corresponding degrees of freedom within each row, note that the sum of the `Factor` and `Error` df equals the `Total` df;
  - the third column with heading `Sum Sq` gives the sum of squares corresponding to each source, note that the sum of squares for the `Factor` and `Error` add up to the `Total` sum of squares (this is where the partitioning of the variability occurs that makes ANOVA possible);
  - the fourth column with heading `Mean Sq` gives the mean sum of squares corresponding to each source, note that these are found by dividing the sum of squares in each row by the corresponding df.

Pause for Reflection #3:

Copy the following partial ANOVA table for the pollution example into your lab notebook and fill in the missing values, denoted by `---`. Upload an image of your work into your lab notebook.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Factor	---	---	---	---	---
Error	---	---	0.03336		
Total	---	---			

Performing the ANOVA *F* Test in R: Thankfully, there is a function in R that performs the extensive calculations needed to perform ANOVA, given by `aov()`. Calling the `aov()` function on the data performs the calculations, and then using the `summary()` function on the results constructs the ANOVA table. The following code demonstrates how this works in the pollution example. The first step is to format the data in R. Notice that an object `Plant` is created to store labels for the observed waste amounts so that we can sort the observations into the appropriate *treatment groups* corresponding to the four populations given by the four plants.

```
# format the data in R
Waste = c(1.65, 1.72, 1.50, 1.37, 1.60,
          1.70, 1.85, 1.46, 2.05,
          1.40, 1.75, 1.38, 1.65, 1.55,
          2.10, 1.95, 1.65, 1.88)
Plant = rep(c("A", "B", "C", "D"), c(5, 4, 5, 4))

# Perform the ANOVA:
results = aov(Waste ~ Plant)     # store the ANOVA calculations in results
summary(results)                 # construct the ANOVA table
```

By running the above code for yourself (already provided in the Lab 7 Notebook), you can check your answers to Reflection #3.

Pause for Reflection #4:

In the above code, explain what the following line does:

```
Plant = rep(c("A", "B", "C", "D"), c(5, 4, 5, 4))
```

In particular, what does the function `rep()` do?

Zombies: Let's look at another example to see how to use the `aov()` function given a data set. The `Zombies.csv` file contains data about the number of zombies killed (`killed`) and by what household weapon (`weapon`) for a sample of 31 apocalypse survivors. Load the data and view it:

```
Zombies = read.csv("Zombies.csv")
View(Zombies)
```

Pause for Reflection #5:

Conduct some EDA:

- What are the mean and standard deviation of zombies killed across weapons (hint: the `tapply()` function will be useful)?
- How many observations of zombies killed are there for each of the weapons (hint: the `table()` function will be useful)?

- Create side-by-side boxplot to compare the distributions of zombies killed across weapons.

From the EDA, it sure looks as though there are differences in the number of zombies killed by each weapon, but are these differences due to sampling error, or do they represent real differences in zombie-killing effectiveness? To answer that question, we need to run an ANOVA test.

```
aov = aov(killed ~ weapon, Zombies)     # store results of ANOVA test
summary(aov)                            # view the ANOVA table
```

Pause for Reflection #6:

State the hypotheses being tested by the above ANOVA calculations. Report the P-value you find and state the conclusion.

Assumptions for the ANOVA F Test: In performing the ANOVA F test, the following assumptions are made:

- the samples are independent
- the populations are normally distributed
- the populations have equal variance

The independence assumption is critical, if the samples are related in some way then a different procedure is needed. Violations of the assumptions of normality and equal variances are less important.

The big problem with non-normality in t tests is the effect of skewness on one-sided tests. But ANOVA tests are inherently two sided (we are testing for any differences between means, not differences in one direction) so non-normal distributions generally have little effect as long as the sample sizes are reasonably large.

If the sample sizes n_i are roughly equal, then unequal variances do not have a great impact, but if the population variances differ, then the actual sampling distribution of the F statistic could be very different from an F distribution. In particular, if there is a small sample from a population with large variance, then the F statistic can explode.

We will run simulations to explore the assumptions for ANOVA: in particular, how does "un-balancedness" (sample sizes not the same) and unequal population variances affect the outcome? We consider the hypotheses

$$H_0 : \mu_A = \mu_B = \mu_C \quad \text{vs.} \quad H_A : \text{at least one mean is different.}$$

The code below simulates drawing three random samples from populations (called A, B, C) with the same mean ($\mu = 20$) and standard deviation ($\sigma = 3$) and then performs an ANOVA test. Using a significance level of 0.05, the object `counter` keeps track of how many times the null hypothesis is incorrectly rejected (false positive) and then corresponding proportion is computed.

```
n.A = 50                                        # set sample sizes
n.B = 50
n.C = 50

Group = rep(c("A","B","C"), c(n.A, n.B, n.C))     # create group labels

counter = 0
N = 10^4

for (i in 1:N)
{
 a = rnorm(n.A, 20, 3)                          # Draw samples from N(20,3) pop
 b = rnorm(n.B, 20, 3)
```

```
  c = rnorm(n.C, 20, 3)
  X = c(a, b, c)                                  # Combine into one vector

  Pvalue = summary(aov(X ~ Group))[1,5]           # Extract P-value from ANOVA table
  if (Pvalue < 0.05)                              # Reject H0, at 0.05 sig level?
   counter = counter + 1                          # If yes, increase counter
 }

 counter/N                                        # proportion of times H0 rejected
```

Pause for Reflection #7:

What type of error is `counter` keeping track of? Is the proportion given by `counter/N` close to what you would expect the probability of making that type of error to be?

_____

Pause for Reflection #8:

Alter the code so that the sample size from A is 10 (`n.A = 10`) and redo the simulation. What happens to the proportion of times H_0 is rejected?

_____

Pause for Reflection #9:

Alter the code again by increasing the standard deviation of population A to 9 and trying samples of size 50 and 10 (keeping the other sample sizes to 50). What proportion of times do you reject the null hypothesis in each case?

_____

Pause for Reflection #10:

Explore other scenarios: What if the population means are all different, but the population variances are the same? How do sample sizes affect the outcome? Try with all sample sizes the same and then unequal. Now try different variances and again, with balanced and unbalanced samples.

Record in your lab notebook what scenarios you tried and what results you found, i.e., how the proportion of times H_0 was rejected is impacted.

_____

Lab 7: ANOVA is shared under a not declared license and was authored, remixed, and/or curated by LibreTexts.

Lab 8: More ANOVA

Objective:

1. Understand when and how to perform *post hoc* analysis of a significant ANOVA F test for comparing three or more population means.
2. Understand how to perform ANOVA using a permutation test.

Definitions:

- pairwise comparisons
- *post hoc* test
- Tukey's HSD method
- classes (produced by *post hoc* analysis of ANOVA results)

Introduction:

In Lab 7, we considered a technique for testing claims about three or more population means known as *analysis of variance* (*ANOVA*). In particular, we used the classical approach to performing ANOVA based on the F distribution. The classical approach requires that the populations being compared are normally distributed with equal variances. In this lab, we will see how to use the permutation test procedure, introduced in Lab 2, to perform ANOVA when these requirements do not appear to be met.

Before developing the permutation test for ANOVA, we will discuss what to do *after* obtaining a significant result for the ANOVA F test.

Activities:

Getting Organized: *If you are already organized, and remember the basic protocol from previous labs, you can skip this section.*

Navigate to your class folder structure. Within your "Labs" folder make a subfolder called "Lab8". Next, download the lab notebook .Rmd file for this lab from Blackboard and save it in your "Lab8" folder. You will again be working with the `Zombies.csv` data set on this lab. You should either re-download the data file into your "Lab8" folder from Blackboard, or just copy the file from your "Lab7" folder into the "Lab8" folder.

Within RStudio, navigate to your "Lab8" folder via the file browser in the lower right pane and then click "More > Set as working directory". Get set to write your observations and R commands in an R Markdown file by opening the "lab8_notebook.Rmd" file in RStudio. Remember to add your name as the author in line 3 of the document. For this lab, enter all of your commands into code chunks in the lab notebook. You can still experiment with code in an R script, if you want. To set up an R Script in RStudio, in the upper left corner click “File > New File > R script”. A new tab should open up in the upper left pane of RStudio.

After ANOVA: If an ANOVA test is significant, then all we know is that *there is a difference*, but we don't know *exactly what* the difference is. For example, let's revisit the four chemical plants we considered in Lab 7. Recall that we analyzed the amount of polluting effluents per gallon in samples of liquid waste discharged from the four plants. We used the following code to perform an ANOVA F test to determine if there is a significant difference between the mean weight of polluting effluents discharged by the four plants (note that the data below are slightly different than the data used in Lab 7):

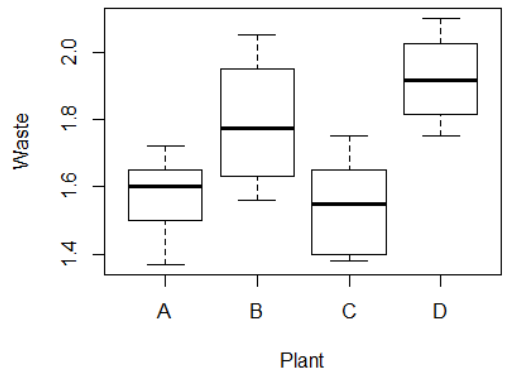
```
Waste = c(1.65, 1.72, 1.50, 1.37, 1.60,
          1.70, 1.85, 1.56, 2.05,
          1.40, 1.75, 1.38, 1.65, 1.55,
          2.10, 1.95, 1.75, 1.88)
Plant = rep(c("A", "B", "C", "D"), c(5, 4, 5, 4))
aov.pollution = aov(Waste ~ Plant)
```

```
summary(aov.pollution)

##                Df  Sum Sq  Mean Sq   F value  Pr(>F)
## Plant           3  0.4297  0.14323      5.39  0.0112 *
## Residuals      14  0.3720  0.02657
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the ANOVA table, we see that the P-value is significant at the $\alpha = 0.05$ level, so we conclude that there is sufficient evidence that the mean weight of polluting effluents discharged by the four plants is *not equal*. However, this result alone says nothing about which plant or plants have the largest mean or the smallest.

If we look at side-by-side boxplots of the sample waste amounts across the four plants, we can see that the sample from plant D appears to have the highest mean and plant C the lowest. But is the mean for plant D significantly higher than the mean for plant B? In other words, when we make ***pairwise comparisons*** between the plant means are the differences significant? To answer this question, we need to run more hypothesis tests.



Pause for Reflection #1:

Why do we need to run more hypothesis tests to determine whether the differences in the means between the pairs are significant? Why isn't it enough to just look at the side-by-side boxplots?

***Post Hoc* Tests:** As we discussed in class, we cannot simply perform a series of independent samples t tests for each pairwise comparison because it is not efficient and it inflates the possibility of committing a Type I error. Instead, we will perform a ***post hoc* test**. Generally speaking, a *post hoc* test is a test of significance that decreases the probability of making a Type I error. In particular, we will look at using the *post hoc* test given by **Tukey's HSD** (honestly significant difference).

The method of Tukey's HSD is to *simultaneously* construct confidence intervals for all differences $(\mu_i - \mu_j)$ that *collectively* hold at the desired confidence level. The pairwise comparisons are then made as follows:

- If the confidence interval for $(\mu_i - \mu_j)$ *includes 0*, then μ_i and μ_j are *not* significantly different.
- If the confidence interval for $(\mu_i - \mu_j)$ *does not include 0*, then μ_i and μ_j *are significantly* different.

Tukey's method is performed in R by calling the function `TukeyHSD()` on the results of the ANOVA F test. For the pollution example, try the following:

```
TukeyHSD(aov.pollution, conf.level = 0.95)     # default conf.level is 95%
```

In the table produced by the above code, the `diff` column gives the difference in the observed sample means, `lwr` gives the lower end point of the confidence interval, `upr` gives the upper end point and `p adj` gives the p-value after adjustment for the multiple comparisons. Note that we can also visualize the results of Tukey's method by calling the `plot()` function on the results:

```
plot(TukeyHSD(aov.pollution))
```

Pause for Reflection #2:

Using the results of Tukey's method, identify which pairs of chemical plants appear to have significantly different means. Then sort the plants into ***classes***, where each class contains plants that have similar means (i.e., means that were determined to *not* be significantly different). It may be helpful to redo the side-by-side boxplot so that the plots are placed in order of decreasing means (the default ordering is alphabetical):

```
Plant = factor(Plant, levels = c("D", "B", "A", "C")) # reorder the labels
boxplot(Waste ~ Plant)
```

Permutation Test for ANOVA: In Lab 7 and in class on Tuesday, we explored using simulation to see what happens when the assumptions required for the ANOVA F test are not met. Specifically, we saw that when the data do not support the assumption of equal variance for the populations, the risk of making a Type I error increases.

For example, let's revisit the data set in the `Zombies.csv` file, containing data about the number of zombies killed (`killed`) and by what household weapon (`weapon`) for a sample of 31 apocalypse survivors. Using the `tapply()` function, we can compute the variances for the samples across the weapons:

```
Zombies = read.csv("Zombies.csv")
tapply(Zombies$killed, Zombies$weapon, var)

## baseball bat        chainsaw        golf club
##     6.454545        6.011111         2.322222
```

Based on these results, we may question the validity of assuming that the population variances are equal, which then calls into question the reliability of an ANOVA performed using the F test approach.

In the case that the assumptions of the F test do not appear to be met, we can use the ideas presented in Lab 2 to form a permutation distribution of the test statistic, rather than using the F distribution. Specifically, the permutation test approach for performing ANOVA uses the following steps:

1. randomly permute the group labels on the observations;
2. compute the test statistic given by the ratio of the between group variability to the within group variability (i.e., MSTR/MSE) for the new groups using the permuted labels;
3. repeat steps 1 & 2 many times storing the resulting test statistic;
4. compute the P-value by finding the proportion of times the resulting test statistics from steps 1 & 2 exceed the original observed test statistic.

The following code implements the permutation test approach to ANOVA for the Zombie example:

```
observed = summary(aov(Zombies$killed ~ Zombies$weapon))[[1]][1,4]     #original obse

n = length(Zombies$killed)                                              #total number o
```

```
N = 10^4 - 1                                                          #number of time
results = numeric(N)

for (i in 1:N)
{
  index = sample(n)                                                   #create permuta
  killed.perm = Zombies$killed[index]                                 #reorder observ
  results[i] = summary(aov(killed.perm ~ Zombies$weapon))[[1]][1,4]   #store new valu
}

(sum(results >= observed) + 1) / (N + 1)                        #P-value
```

Pause for Reflection #3:

Run the above code to perform the permutation test. Record the P-value you find in your lab notebook and compare the results to what you found in Lab 7 when performing the ANOVA F test. Comment on whether you think that a permutation test was necessary.

Pause for Reflection #4:

Comment on why the P-value of the permutation test for ANOVA is given by the proportion of times the resulting test statistics for the permutations *exceed* the original observed test statistic. In particular, explain why a test statistic for a permutation that is as large or larger than what was originally observed is considered *more extreme* in this context.

Lab 8: More ANOVA is shared under a not declared license and was authored, remixed, and/or curated by LibreTexts.

Lab 9: Categorical Data

Objectives:

1. Understand how to analyze categorical data
2. Understand how to perform chi-square tests in R.

Definitions:

- categorical (qualitative) data
- chi-square distribution
- observed vs. expected counts
- goodness-of-fit test
- contingency table
- test of homogeneity
- test of independence

Introduction:

Recall that *categorical data* is data based on some attribute or characteristic. The observations fall into *categories*. Up to this point, we have performed hypothesis tests primarily about population means. But if we are interested in testing claims about categorical data, then we need a new approach, since we cannot compute means for categorical variables. Instead we focus on proportions, and we have only developed tests for comparing two proportions at a time. In this lab, we will look at methods to analyze relationships between categorical variables and to check how well a probability model fits a single categorical variable.

Activities:

Getting Organized: *If you are already organized, and remember the basic protocol from previous labs, you can skip this section.*

Navigate to your class folder structure. Within your "Labs" folder make a subfolder called "Lab9". Next, download the lab notebook .Rmd file for this lab from Blackboard and save it in your "Lab9" folder. There are no datasets used in this lab.

Within RStudio, navigate to your "Lab9" folder via the file browser in the lower right pane and then click "More > Set as working directory". Get set to write your observations and R commands in an R Markdown file by opening the "lab9_notebook.Rmd" file in RStudio. Remember to add your name as the author in line 3 of the document. For this lab, enter all of your commands into code chunks in the lab notebook. You can still experiment with code in an R script, if you want. To set up an R Script in RStudio, in the upper left corner click “File > New File > R script”. A new tab should open up in the upper left pane of RStudio.

Goodness-of-Fit Tests: In class on Tuesday, we considered whether any one day of the week is more or less likely to be a person’s birthday than any other day of the week. Let p_{M} denote the proportion of *all* people that were born on a Monday, or equivalently, the probability that a randomly selected person was born on a Monday. Similarly, define p_{Tu}, p_{W}, p_{Th}, p_{F}, p_{Sa}, and p_{Su} We are testing the following hypotheses:

$$H_0 : p_{\mathrm{M}} = p_{\mathrm{Tu}} = p_{\mathrm{W}} = p_{\mathrm{Th}} = p_{\mathrm{F}} = p_{\mathrm{Sa}} = p_{\mathrm{Su}} = 1/7$$
$$H_A : p_i \neq 1/7 \text{ for at least one day of the week}$$

In other words, we are testing whether the probability model stated in the null hypothesis fits the data well.

To test these hypotheses, you created a version of the following table:

	Days of the Week							Total: n
	Mon	Tues	Wed	Thu	Fri	Sat	Sun	
Observed counts: O_i	17	26	22	23	19	15	25	147
Expected counts: $E_i = np_i$	21	21	21	21	21	21	21	147
$(O_i - E_i)^2/E_i$	0.76	1.19	0.05	0.19	0.19	1.71	0.76	4.86

The test statistic in this case, 4.86, follows a **chi-square distribution**, with degrees of freedom equal to the number of categories (i.e., days of the week) minus one, and so the P-value is calculated in R as follows:

```
pchisq(4.86, df = 6, lower.tail = FALSE)

## [1]    0.5618907
```

Pause for Reflection #1:

Suppose we suspect that weekend days are less likely to be a birthday, perhaps because doctors want the weekend off and so do not schedule Caesarean deliveries for weekends. Let's test whether the data provide evidence against the hypothesis that weekend days are half as likely as other days to be someone's birthday and that all weekdays are equally likely.

- State the hypotheses being tested in this case. The null hypothesis should give the proposed probability model for the data. Note that not all the days of the week will have the same probabilities, but we will still need the probabilities to add up to 1.
- Redo the table above to calculate the test statistic in this case. Note that we are using the same data, so the observed counts stay the same, but the expected counts will change.
- Alter the R code above to calculate the corresponding P-value and state the conclusion of the test.
- Which category (day) has the largest contribution to the test statistic? Explain what this reveals.

Chi-Square Test in R: As you may have already guessed, there is a function in R, `chisq.test()`, that performs the calculations you just did. To use this function, store the observed counts in a list:

```
birthdays = c(17, 26, 22, 23, 19, 15, 25)
```

For the test that each day of the week is equally likely, all we have to do is call the `chisq.test()` function on the object containing the observed counts as follows, since by default R tests the data against the null hypothesis that all probabilities are equal:

```
chisq.test(birthdays)

##
## Chi-squared test for given probabilities
##
## data: birthdays
## X-squared = 4.8571, df = 6, p-value = 0.5623
```

The output above gives the value of the observed test statistic `X-squared` and the degrees of freedom `df` for the chi-square distribution used to calculate the corresponding `p-value`.

For the test that weekend days are half as likely as other days, we need to specify the probabilities stated in the null hypothesis in the `chisq.test()` function as follows:

```
probs = c(rep(1/6, 5), 1/12, 1/12)
chisq.test(birthdays, p = probs)
```

Pause for Reflection #2:

Explain the code above, specifically the line defining the object `probs`. Does the output of the `chisq.test` match the results you found in Reflection #1?

Newspaper Reading: Are Americans today less likely to read a newspaper every day than in previous years? The General Social Survey (GSS) interviews a random sample of adult Americans every two years, and one of the questions asks respondents,"How often do you read the newspaper?" Sample results for the years 1978, 1988, 1998, 2008, and 2018 are given in the *contingency table* below.

	1978 1988 1998 2008 2018	total
Every day	874 500 805 431 321	2922
Not every day	654 488 1065 898 1247	4352
total	1528 988 1870 1329 1559	7274

In asking whether or not these sample data provide evidence that the proportion of Americans who read the newspaper every day differed among the five populations for these years, we have to ask how likely it is to have observed such sample data if, in fact, the

"every day" proportions were the same for all five populations (years). However, it's a little harder to quantify this now that we are comparing more than two groups.

We adopt a strategy similar to the goodness-of-fit test: Compare the *observed* counts in the table with the counts *expected* under the null hypothesis of equal population proportions/distributions. The farther the observed counts are from the expected counts, the more extreme we will consider the data to be.

Pause for Reflection #3:

Use appropriate symbols to state the null hypothesis that the population proportion of adult Americans who read the newspaper every day was the same for these five years: 1978, 1988, 1998, 2008, and 2018.

Pause for Reflection #4:

For the five years combined, what proportion of respondents read the newspaper every day? If this same proportion of the 1528 respondents in the year 1978 had read the newspaper every day, how many people would this represent? Record your answer with two decimal places, and repeat for the other four years.

We have now calculated the *expected counts* under the null hypothesis that the population proportion of adult Americans who read the paper every day was the same for these four years (and consequently also the population proportions who did not read the paper every day). A more general technique for calculating the expected count of cell i is to take the marginal total for that row times the marginal total for that column, divided by the grand total (sample size of the study, n):

$$E_i = \frac{\text{row total} \times \text{column total}}{\text{grand total}} \tag{Lab 9.1}$$

Pause for Reflection #5:

Use the general formula in Equation (9.1) to calculate the expected count of "not every day" people in the year 1988 and complete the following table:

	1978 1988 1998 2008 2018	total
Every day	874 500 805 431 312 (613.80) (396.88) (751.19) (533.87) (626.26)	2922
Not every day	654 488 1065 898 1247 (914.20) ( ) (1118.81) (795.13) (932.74)	4352
total	1528 988 1870 1329 1559	7274

Now that we have the observed counts and the expected counted calculated, we need to find a *test statistic* to measure how far the observed counts deviate from the expected counts. To do this, we do the same calculation as with the goodness-of-fit test:

$$X^2 = \sum_{\text{all cells } i} \frac{(O_i - E_i)^2}{E_i}$$

Pause for Reflection #6:

Calculate the value of $(O_i - E_i)^2/E_i$ for the "not every day" people in 1988 (i.e., for the second cell in the second row of the table). Add this value to other contributions to the test statistic calculation provided below and compute the test statistic:

X^2 = 110.30 + 26.79 + 3.85 + 19.82 + 157.70
+ 74.06 + ?? + 2.59 + 13.31 + 105.88 = ??

What kind of values (e.g., large or small) of the test statistic provide evidence against the null hypothesis that the five populations (years) have the same proportion of Americans reading the newspaper every day? Explain.

Again in this case, the test statistic follows a **chi-square distribution**. However, in this case, the degrees of freedom are equal to $(r - 1)(c - 1)$, where r is the number of rows and c is the number of columns in the contingency table.

Pause for Reflection #7:

Calculate the degrees of freedom for the test statistic found in Reflection #6 and then use the `pchisq()` function to find the corresponding P-value. Based on the P-value, state your conclusion.

Tests of Homogeneity: The test we just performed is called a **chi-square test of equal proportions (homogeneity)**. It is used to test whether the proportions for independent samples from three or more populations are the same. And the calculations can also be done in R with the `chisq.test()` function. First, we need to format the observed counts in R, which can be done using the `rbind()` command:

```
years = rbind(c(874, 500, 805, 431, 312), c(654, 488, 1065, 898, 1247))
years

##         [,1]   [,2]   [,3]   [,4]   [,5]
## [1,]    874    500    805    431    312
## [2,]    654    488    1065   898    1247
```

Then, we simply call the `chisq.test()` function on the table of observed counts `years`:

```
chisq.test(years)

##
## Pearson's Chi-squared test
##
## data: years
## X-squared = 532.28, df = 4, p-value < 2.2e-16
```

We can see the expected counts in R with the following code:

```
chisq.test(years)$expected

##             [,1]        [,2]        [,3]         [,4]        [,5]
## [1,]    613.8048    396.8842     751.187     533.8655    626.2876
## [2,]    914.1952    591.1158    1118.8122    795.1345    932.7424
```

Tests of Independence: We continue to consider the GSS survey. But this time, we use only the year 2018 with another variable: the respondent's political inclination, classified as liberal, moderate, or conservative. The sample results are summarized in the table:

	Liberal	Moderate	Conservative
Every day	109	153	160
Few times a week	85	109	95
Once a week	52	82	63
Less than once a week	56	68	64
Never	52	65	63

Notice how this data is different from the data used in the previous example regarding newspaper reading. In this case, we have *one* random sample of individuals (2018 respondents) that are classified according to *two* variables (political inclination and how often they read the newspaper). Previously, we had *five* separate random samples (for the five years) that were classified on just *one* variable.

It turns out that the same chi-square test applies to two-way tables where the data are one random sample from a population classified on two variables. The difference in the null hypothesis being tested is that, in the population, the two variables are *independent*, and the alternative hypothesis is that there is a *relationship* between the variables.

For the above data, we perform a **chi-square test of independence** for the following hypotheses:

H_0 : political inclination and how often someone reads the paper are independent

H_A : political inclination is related to how often someone reads the paper

Pause for Reflection #8:

Format the data in R using the `rbind()` function. Then call the `chisq.test()` function on the data to perform the calculations for the test of independence. Record your conclusion in your lab notebook. If the test indicates strong evidence of a relationship between the variables, examine the table cells that contribute most to the value of the test statistic in order to describe the relationship.

Lab 9: Categorical Data is shared under a not declared license and was authored, remixed, and/or curated by LibreTexts.

Index

D

dire

Glossary

Sample Word 1 | Sample Definition 1

Detailed Licensing

Overview

Title: MATH 346 - Statistics (Kuter)

Webpages: 22

All licenses found:

- Undeclared: 100% (22 pages)

By Page

- MATH 346 - Statistics (Kuter) - *Undeclared*
  - Front Matter - *Undeclared*
    - TitlePage - *Undeclared*
    - InfoPage - *Undeclared*
    - Table of Contents - *Undeclared*
    - Licensing - *Undeclared*
  - Labs - *Undeclared*
    - Lab 1: Getting Started with R and EDA - *Undeclared*
    - Lab 2: Intro to Hypothesis Testing - Permutation Tests - *Undeclared*
    - Lab 3: Parameter Estimation - *Undeclared*
    - Lab 4: Samping Distributions - *Undeclared*
    - Lab 5: Confidence Intervals - *Undeclared*
    - Lab 6: More Hypothesis Testing - Classical Approach - *Undeclared*
    - Lab 7: ANOVA - *Undeclared*
    - Lab 8: More ANOVA - *Undeclared*
    - Lab 9: Categorical Data - *Undeclared*
    - Lab 10: Simple Linear Regression - *Undeclared*
    - Lab 11: More Regression - *Undeclared*
  - Back Matter - *Undeclared*
    - Index - *Undeclared*
    - Glossary - *Undeclared*
    - Detailed Licensing - *Undeclared*