

CHEMOMETRICS USING R



David Harvey
DePauw University

DePauw University

Chemometrics Using R

David Harvey

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 adaptations contact info@LibreTexts.org or visit our main website at <https://LibreTexts.org>.

The LibreTexts name, logo, and book covers are trademarked by LibreTexts, Inc. (a 501(c)(3) not-for-profit organization) and protected against unauthorized use.

This text was compiled on 08/19/2026

TABLE OF CONTENTS

Licensing

What is Chemometrics and Why Study it?

1: R and RStudio

- 1.1: Installing and Accessing R and RStudio
- 1.2: The Basics of Working With R
- 1.3: Exercises

2: Types of Data

- 2.1: Ways to Describe Data
- 2.2: Using R to Organize and Manipulate Data
- 2.3: Exercises

3: Visualizing Data

- 3.1: Types of Visualizations
- 3.2: Using R to Visualize Data
- 3.3: Creating Plots From Scratch in R Using Base Graphics
- 3.4: Exercises

4: Summarizing Data

- 4.1: Ways to Summarize Data
- 4.2: Using R to Summarize Data
- 4.3: Exercises

5: The Distribution of Data

- 5.1: Terminology
- 5.2: Theoretical Models for the Distribution of Data
- 5.3: The Central Limit Theorem
- 5.4: Modeling Distributions Using R
- 5.5: Exercises

6: Uncertainty of Data

- 6.1: Properties of a Normal Distribution
- 6.2: Confidence Intervals
- 6.3: Using R to Model Properties of a Normal Distribution
- 6.4: Using R to Find Confidence Intervals
- 6.5: Exercises

7: Testing the Significance of Data

- 7.1: Significance Testing
- 7.2: Significance Tests for Normal Distributions
- 7.3: Analysis of Variance

7.4: Non-Parametric Significance Tests

7.5: Using R for Significance Testing and Analysis of Variance

7.6: Exercises

8: Calibrating Data

8.1: Unweighted Linear Regression With Errors in y

8.2: Weighted Linear Regression with Errors in y

8.3: Weighted Linear Regression With Errors in Both x and y

8.4: Curvilinear, Multivariable, and Multivariate Regression

8.5: Using R for a Linear Regression Analysis

8.6: Exercises

9: Optimizing Data

9.1: Response Surfaces

9.2: Searching Algorithms

9.3: One-Factor-at-a-Time Optimizations

9.4: Simplex Optimization

9.5: Mathematical Models of Response Surfaces

9.6: Using R to Model a Response Surface (Multiple Regression)

9.7: Exercises

10: Cleaning Up Data

10.1: Signals and Noise

10.2: Improving the Signal-to-Noise Ratio

10.3: Background Removal

10.4: Using R to Clean Up Data

10.5: Exercises

11: Finding Structure in Data

11.1: What Do We Mean By Structure?

11.2: Cluster Analysis

11.3: Principal Component Analysis

11.4: Multivariate Linear Regression

11.5: Using R for a Cluster Analysis

11.6: Using R for a Principal Component Analysis

11.7: Using R for a Multivariate Linear Regression

11.8: Exercises

12: Appendices

12.1: Single-Sided Normal Distribution

12.2: Critical Values for t-Test

12.3: Critical Values for F-Test

12.4: Critical Values for Dixon's Q-Test

12.5: Critical Values for Grubb's Test

[12.6: Critical Values for the Wilcoxon Signed Rank Test](#)

[12.7: Critical Values for the Wilcoxon Ranked Sum Test](#)

13: Resources

[13.1: Chemometric Resources](#)

[13.2: R Resources](#)

[Index](#)

[Glossary](#)

[Detailed Licensing](#)

This page titled [Chemometrics Using R \(Harvey\)](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by David Harvey.

Licensing

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

What is Chemometrics and Why Study it?

What is Chemometrics?

The definition of chemometrics is evident in its name, where *chemo-* means chemical and *-metrics* means measurement; thus, chemometrics is the study of chemical (and biochemical) measurements and is a branch of analytical chemistry. Examples of chemometric applications include

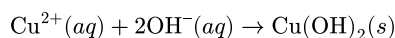
- ensuring that the data we collect is appropriate for our purposes
- enhancing the quality of an analytical signal by finding ways to minimize the contribution of noise
- reporting on an experiment in a way that estimates the uncertainty in its results and our confidence in those results
- building useful models that predict the outcomes of future experiments
- extracting from chemical data, hidden, but analytically useful information by finding underlying patterns in the data

These topics, and others, are the focus of this textbook.

Why Study Chemometrics?

Why chemometrics is important becomes clear when we consider a simple analytical problem: How do we determine the concentration of copper in a sample, and how and why has the analytical method used for this analysis changed over time.

Prior to the 1950s, gravimetry and titrimetry were the most common analytical methods for determining the concentration of copper in a variety of samples. Both of these methods rely on simple stoichiometric relationships. In a gravimetric analysis, for example, we bring copper into solution as $\text{Cu}^{2+}(\text{aq})$, precipitate it as $\text{Cu}(\text{OH})_2(\text{s})$



and isolate it as $\text{CuO}(\text{s})$ after heating it to a high temperature.

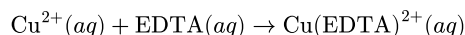


We then use the mass of $\text{CuO}(\text{s})$ to determine the amount of copper in the original sample by accounting for the simple stoichiometric relationship between Cu and CuO where each mole of Cu yields one mole of CuO.

Note

You can read more about gravimetry in Chapter 8 of the textbook *Analytical Chemistry 2.1*.

In a titrimetric analysis, we bring copper into solution as $\text{Cu}^{2+}(\text{aq})$ and slowly add a solution of ethylenediaminetetracetic acid, EDTA, until the moles of EDTA added is equal to the moles of Cu^{2+} in the original sample.



If we know the concentration of our EDTA solution, then it is easy to determine the amount of Cu^{2+} in the original sample using the simple stoichiometric relationship between Cu^{2+} and EDTA. For both of these analyses, a chemometric treatment of the data consists of little more than reporting an average, a standard deviation, and a confidence interval.

Note

You can read more about titrimetry in Chapter 9 of the textbook *Analytical Chemistry 2.1*.

Gravimetry and titrimetry are useful analytical methods when copper is a major ($> 1\%$ w/w) analyte or a minor analyte (0.01% w/w – 1% w/w) analyte, but less useful if it is a trace analyte ($10^{-7}\%$ w/w – 0.01% w/w). Neither method affords a rapid analysis, which makes them less useful if we need to analyze multiple analytes in a large number of samples.

Note

For more information about the scale of operations for analytical chemistry, including the relative concentrations of analytes in samples, see Chapter 3.4 of the textbook *Analytical Chemistry 2.1*.

Beginning in the 1950s, instrumental methods of analysis emerged in which an analytical signal is related to the analyte's concentration, not through the stoichiometry of one or more chemical reactions, but through a theoretical relationship in which at least one variable is not known to us. For example, a solution of $\text{Cu}^{2+}(\text{aq})$ is light blue in color because it absorbs light over a broad range of wavelengths between about 600–900 nm, as we see in Figure 1.

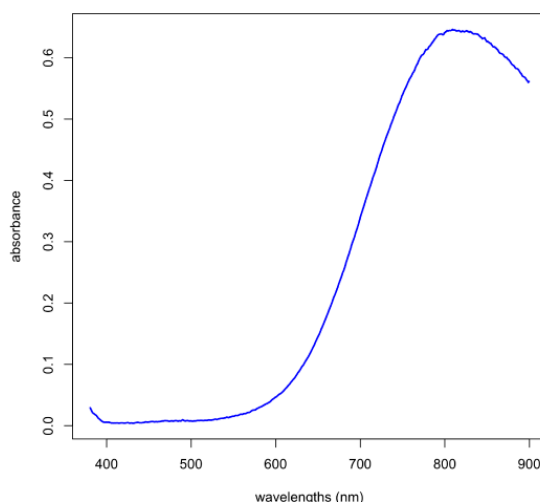


Figure 1: Visible absorbance spectrum for $\text{Cu}^{2+}(\text{aq})$.

The relationship between a solution's absorbance, A_λ , at a specific wavelength, λ , and a given concentration, C , of $\text{Cu}^{2+}(\text{aq})$ is given by Beer's law

$$A_\lambda = \epsilon_\lambda b C$$

where ϵ_λ is the analyte's molar absorptivity at the selected wavelength, λ , and b is the distance light travels through the sample. Of these variables— A_λ , ϵ_λ , b , and C —the value of ϵ_λ is not known to us. Contrast that to gravimetry and titrimetry where we almost always know the exact stoichiometric relationships.

Note

For more information about visible absorption spectroscopy and Beer's Law, see Chapter 10.2 in *Analytical Chemistry 2.1*.

Although we can measure A_λ and b , we cannot calculate C without first determining the value of ϵ_λ , which we do using a standard solution for which the concentration of analyte is known, C_{std} . If we use a single standard and a single wavelength—which is all early instrumentation allowed—then we have

$$[A_{\lambda, std}]_{1 \times 1} = [\epsilon_\lambda b]_{1 \times 1} \times [C_{std}]_{1 \times 1}$$

which we can solve exactly for $\epsilon_\lambda b$. With this value in hand, we can use the sample's absorbance to calculate the analyte's

concentration in the sample.

Note

Note that we are expressing Beer's Law here using the matrix notation $[\]_{r \times c}$, where r is the number of rows and c is the number of columns in the matrix. In this equation, each matrix holds a single value: an absorbance, a value for $\epsilon_{\lambda}b$, or a concentration. A matrix with a single value is a scalar. A matrix with a single column or a single row is a vector. The reason for expressing Beer's Law in this way will soon be evident.

If we use c standards instead of one standard, and if we continue to use a single wavelength, then we can write Beer's law this way

$$[\dots A_{\lambda, std} \dots]_{1 \times c} = [\epsilon_{\lambda}b]_{1 \times 1} \times [\dots C_{std} \dots]_{1 \times c} + [\dots E \dots]_{1 \times c}$$

where the absorbance values and the concentrations are vectors with dimensions of $1 \times c$ (1 wavelength and c standards), where the value of $\epsilon_{\lambda}b$ is a scalar (a constant), and where we have a vector of residual errors, E , that gives the uncertainties in our measured absorbance values. Having multiple standards provides a new source of information that allows us to consider experimental uncertainty!

Note

Note that the equation $A_{\lambda, std} = \epsilon_{\lambda}bC$ is in the form of a straight-line, $y = \beta_0x + \beta_1$, for which a standard linear regression analysis returns values for the two constants: the slope, β_0 , which is equivalent to $\epsilon_{\lambda}b$ and the y -intercept, β_1 , which is equivalent to the residual error.

If we use r wavelengths and c standards, then we can write Beer's law this way

$$\begin{bmatrix} \dots & \dots & \dots \\ \vdots & A_{\lambda, std} & \vdots \\ \dots & \dots & \dots \end{bmatrix}_{r \times c} = \begin{bmatrix} \vdots \\ \epsilon_{\lambda}b \\ \vdots \end{bmatrix}_{r \times 1} \times [\dots C_{std} \dots]_{1 \times c} + \begin{bmatrix} \dots & \dots & \dots \\ \vdots & E & \vdots \\ \dots & \dots & \dots \end{bmatrix}_{r \times c}$$

where the absorbance values and the residual errors are in matrices (with wavelengths in rows and standards in columns), the values for $\epsilon_{\lambda}b$ at each wavelength are in a vector, and the analyte's concentration in the standards are in a vector; this is a computationally more difficult form of regression, but, as we will learn in a later chapter, one we can solve.

But we can push this even further! Note that the $\epsilon_{\lambda}b$ matrix has one column because we are using a single wavelength, and the C matrix has one row because we assumed just one analyte. As long as the number of analytes is less than the smaller of the number of wavelengths or the number of standards, then we can include additional analytes. For example, if we have n analytes, then

$$\begin{bmatrix} \dots & \dots & \dots \\ \vdots & A_{\lambda, std} & \vdots \\ \dots & \dots & \dots \end{bmatrix}_{r \times c} = \begin{bmatrix} \dots & \dots & \dots \\ \vdots & \epsilon_{\lambda}b & \vdots \\ \dots & \dots & \dots \end{bmatrix}_{r \times n} \times \begin{bmatrix} \dots & \dots & \dots \\ \vdots & C_{std} & \vdots \\ \dots & \dots & \dots \end{bmatrix}_{n \times c} + \begin{bmatrix} \dots & \dots & \dots \\ \vdots & E & \vdots \\ \dots & \dots & \dots \end{bmatrix}_{r \times c}$$

where each column in the $\epsilon_{\lambda}b$ matrix holds the $\epsilon_{\lambda}b$ values for a different analyte at one of our wavelengths, and each row in the C matrix is the concentration of a different analyte in one of our standards; again, we can use linear regression to analyze the data.

Moving from the analysis of a single analyte in a single standard using a single wavelength

$$[A_{\lambda, std}]_{1 \times 1} = [\epsilon_{\lambda}b]_{1 \times 1} \times [C_{std}]_{1 \times 1}$$

to the analysis of multiple analytes using multiple standards and multiple wavelengths

$$\begin{bmatrix} \dots & \dots & \dots \\ \vdots & A_{\lambda, std} & \vdots \\ \dots & \dots & \dots \end{bmatrix}_{r \times c} = \begin{bmatrix} \dots & \dots & \dots \\ \vdots & \epsilon_{\lambda}b & \vdots \\ \dots & \dots & \dots \end{bmatrix}_{r \times n} \times \begin{bmatrix} \dots & \dots & \dots \\ \vdots & C_{std} & \vdots \\ \dots & \dots & \dots \end{bmatrix}_{n \times c} + \begin{bmatrix} \dots & \dots & \dots \\ \vdots & E & \vdots \\ \dots & \dots & \dots \end{bmatrix}_{r \times c}$$

required a significant increase in computational power and a significant growth in the capabilities of instrumentation; not surprisingly, new chemometric techniques rely on and are driven by developments in computer science and instrumental analysis! In turn, new chemometric techniques open up new areas of analysis and encourage innovations in computer science and instrumental analysis. This is why chemometrics is an important part of analytical chemistry.

1: R AND RSTUDIO

As we move through this textbook, we will make frequent use of the statistical programming language R, accessing the program through the RStudio Desktop interface, which provides a useful environment for managing files and for writing code. There are many programs you can use in place of R and RStudio: some, such as Python, are free, and others, such as SPSS or Matlab, are commercial packages. We will use R and RStudio for four reasons:

1. Both R and RStudio are available at no cost.
2. As a programming language, R is designed specifically for the analysis of data; this is one its great strength.
3. The base installation of R comes with most of the tools we need, including tools for visualizing data.
4. When we need additional tools, packages of functions built by other users are available to us.

To ensure that this textbook is not tied too directly to R—and, therefore, accessible to anyone interested in learning about chemometrics—each chapter begins with a general treatment of a chemometric topic that is software-independent, followed by specific examples of how to implement the topic using R.

1.1: Installing and Accessing R and RStudio

1.2: The Basics of Working With R

1.3: Exercises

1.1: Installing and Accessing R and RStudio

Installing R and RStudio

You can download and install R from the R-Project [website](#). On the left side of the page, click on the link to CRAN under the title "Downloads." Scroll through the list of CRAN mirror sites and click on the link to a site located near you. Versions are available for Mac OS, for Windows, and for Linux. Follow the directions for your operating system.

You can download and install the RStudio Desktop Interface from the RStudio [website](#). Click on the Download button for the free version of RStudio Desktop. From the list of available installers, click on the link that is appropriate for your operating system and follow the directions.

Navigating RStudio

When you launch RStudio, the program opens with the four panes as shown in Figure 1.1.1 (although some panes may be minimized).

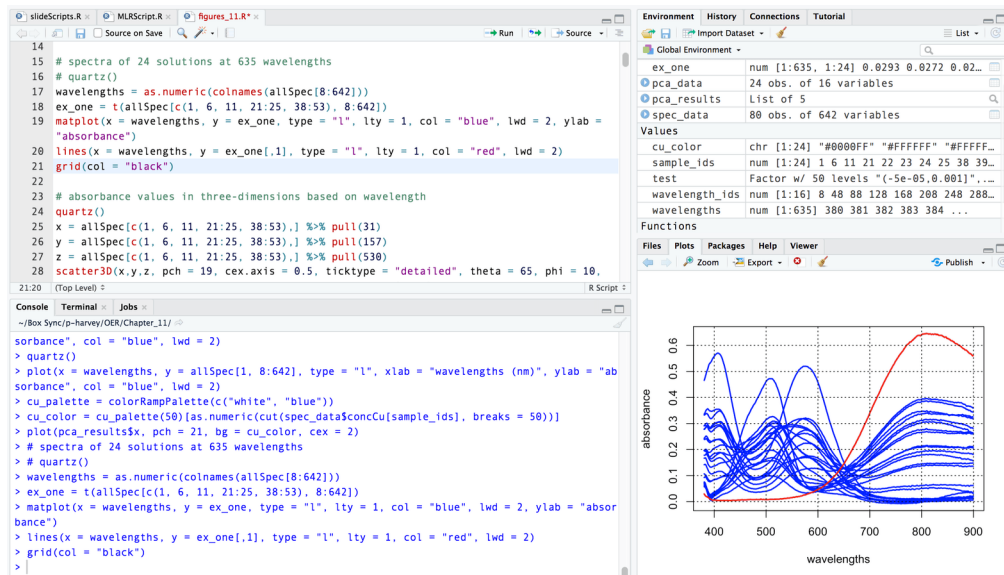


Figure 1.1.1: A screenshot showing the arrangement of RStudio's four panes while I was working on Chapter 11. You can customize the arrangement of these panes by selecting RStudio: Preferences from the menu bar.

Beginning in the lower left corner and moving clockwise, these panes are

- the *Console*, which provides access to R; this is where you can directly enter commands as you work on problems.
- the *Source Pane*, which provides access to a variety of different types of documents, including script files, which end with an extension of .R (more on these later). The source pane also provides a way to submit code to the console by highlighting the code and clicking on the Run button; this usually is a more efficient way to work.
- the *Environment & History Pane*, which provides access to your data and the functions you create while using R.
- the *Files, Plots, Packages, Help & Viewer Pane*, which provides access to your computer's file structure, to help files for R commands, to a list of R packages available to you (packages provide access to additional commands beyond those available to you when you first launch R; more on this in later chapters), to plots that you create, and to an internal web-like browser.

As you work with R, take time to examine each pane so that you become comfortable with them. For example, Figure 1.1.1 shows my RStudio screen after I highlighted lines 15–21 in the script file "figures_11.R" and clicked Run, sending the lines of code to the console where R processed them to create the figure in the lower right pane.

This page titled [1.1: Installing and Accessing R and RStudio](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [David Harvey](#).

1.2: The Basics of Working With R

Communicating With R

The symbol `>` in the console is the command prompt, which indicates that R is awaiting your instructions. When you type a command in the console and hit Enter or Return, R executes the command and displays any appropriate output in the console; thus, this command adds the numbers 1 and 3

```
1 + 3
```

and returns the number 4 as an answer.

```
[1] 4
```

Note

The text above is a code block that contains the line of code to enter into the console and the output generated by R. The command prompt (`>`) is not included here so that you can, if you wish, copy and paste the code into R; if you are copying and pasting the code, do not include the output or R will return an error message. Note that the output here is preceded by the number 1 in brackets, which is the id number of the first value returned on that line.

This is all well and good, but it is even less useful than a calculator because we cannot operate further on the result. If we assign this calculation to an object using an assignment operator, then the result of the calculation remains available to us.

There are two common leftward assignment operators in R: an arrow that points from right-to-left, `<=`, which means the value on the right is assigned to the object on the left, and an equals sign, `=`. Most style guides for R favor `<=` over `=`, but as `=` is the more common option in most other programming languages—such as Python, C++, and Matlab—we will use it here.

If we assign our calculation to the object `answer` then the result of the calculation is assigned to the object but not returned to us. To see an object's value we can look for it in RStudio's Environment Panel or enter the object's name as a command in the Console, as shown here.

```
answer = 1 + 3
```

```
answer
```

```
[1] 4
```

Note that an object's name is case-sensitive so `answer` and `Answer` are different objects.

```
Answer = 2 + 4
```

```
Answer
```

```
[1] 6
```

Note

There are just a few limitations to the names you can assign to objects: they can include letters (both upper and lower case), numbers, dots (`.`), or underscores (`_`), but not spaces. A name can begin with a letter or with a dot followed by a letter (but not a dot followed by a number). Here are some examples of valid names

```
answerone answer_one answer1 answerOne answer.one
```

and examples of invalid names

```
1stanswer answer* first answer
```

You will find it helpful to use names that remind you of the object's meaning and that are not overly long. My personal preference is to use all lowercase letters, to use a descriptive noun, and to separate words using an underscore as I find that these choices make my code easier to read. When I find it useful to use the same base name for several objects of different types, then I may append a two or three letter designation to the name similar to the extensions that designate, for example, a spreadsheet stored as a .csv file. For example, when I use R to run a linear regression based on Beer's law, I may store the concentrations and absorbances of my standards in a data frame (see below for a description of data frames) with a name such as `zinc.df` and store the output of the linear model (see Chapter 8 for a discussion of linear models) in an object with a name such as `zinc.lm`.

Objects for Storing Data

In the code above, `answer` and `Answer` are objects that store a single numerical value. There are several different types of objects we can use to store data, including vectors, data frames, matrices and arrays, and lists.

Vectors

A vector is an ordered collection of elements of the same type, which may be numerical values, integer values, logical values, or character strings. Note that ordered does not imply that the values are arranged from smallest-to-largest or from largest-to-smallest, or in alphabetical order; it simply means the vector's elements are stored in the order in which we enter them into the object. The length of a vector is the number of elements it holds. The objects `answer` and `Answer`, for example, are vectors with lengths of 1.

```
length(answer)
```

```
[1] 1
```

Most of the vectors we will use include multiple elements. One way to create a vector with multiple elements is to use the concatenation function, `c( )`.

Note

In the code blocks below and elsewhere, any text that follows a hashtag, `#`, is a comment that explains what the line of code is accomplishing; comments are not executable code, so R simply ignores them.

For example, we can create a vector of numerical values,

```
v00 = c(1.1, 2.2, 3.3)
```

```
v00
```

```
[1] 1.1 2.2 3.3
```

or a vector of integers,

```
v01 = c(1, 2, 3)
```

```
v01
```

```
[1] 1 2 3
```

or a vector of logical values,

```
v02 = c(TRUE, TRUE, FALSE) # we also could enter this as c(T, T, F)
```

```
v02
```

```
[1] TRUE TRUE FALSE
```

or a vector of character strings

```
v03 = c("alpha", "bravo", "charley")
```

```
v03
```

```
[1] "alpha" "bravo" "charley"
```

You can view an object's structure by examining it in the Environment Panel or by using R's structure command, `str( )` which, for example, identifies vector the `v02` as a logical vector with an index for its entries of 1, 2, and 3, and with values of TRUE, TRUE, and FALSE.

```
str(v02)
```

```
logi [1:3] TRUE TRUE FALSE
```

We can use a vector's index to correct errors, to add additional values, or to create a new vector using already existing vectors. Note that the number within the square brackets, `[ ]`, identifies the element in the vector of interest. For example, the correct spelling for the third element in `v03` is charlie, not charley; we can correct this using the following line of code.

```
v03[3] = "charlie" # correct the vector's third value
```

```
v03
```

```
[1] "alpha" "bravo" "charlie"
```

We can also use the square bracket to add a new element to an existing vector,

```
v00[4] = 4.4 # add a fourth element to the existing vector, increasing its length
```

```
v00
```

```
[1] 1.1 2.2 3.3 4.4
```

or to create a new vector using elements from other vectors.

```
v04 = c(v01[1], v02[2], v03[3])
```

```
v04
```

```
[1] "1" "TRUE" "charlie"
```

Note the the elements of `v04` are character strings even though `v01` contains integers and `v02` contains logical values. This is because the elements of a vector must be of the same type, so R coerces them to a common type, in this case a vector of character strings.

Here are several ways to create a vector when its entries follow a defined sequence, `seq( )`, or use a repetitive pattern, `rep( )`.

```
v05 = seq(from = 0, to = 20, by = 4)
```

```
v05
```

```
[1] 0 4 8 12 16 20
```

```
v06 = seq(0, 10, 2) # R assumes the values are provided in the order from, to, and by
```

```
v06
```

```
[1] 0 2 4 6 8 10
```

```
v07 = rep(1:4, times = 2) # repeats the pattern 1, 2, 3, 4 twice
```

```
v07
```

```
[1] 1 2 3 4 1 2 3 4
```

```
v08 = rep(1:4, each = 2) # repeats each element in the string twice before proceeding to next element
```

```
v08
```

```
[1] 1 1 2 2 3 3 4 4
```

Note

Note that `1:4` is equivalent to `c(1, 2, 3, 4)` or `seq(1, 4, 1)`. In R it often is the case that there are multiple ways to accomplish the same thing!

Finally, we can complete mathematical operations using vectors, make logical inquiries of vectors, and create sub-samples of vectors.

```
v09 = v08 - v07 # subtract two vectors, which must be of equal length
```

```
v09
```

```
[1] 0 -1 -1 -2 2 1 1 0
```

```
v10 = (v09 == 0) # returns TRUE for each element in v10 that equals zero
```

```
v10
```

```
[1] TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE
```

```
v11 = which(v09 < 1) # returns the index for each elements in v09 that is less than 1
```

```
v11
```

```
[1] 1 2 3 4 8
```

```
v12 = v09[!v09 < 1] # returns values for elements in v09 whose values are not less than 1
```

```
v12
```

```
[1] 2 1 1
```

Data Frames

A data frame is a collection of vectors—all equal in length but not necessarily of a single type of element—arranged with the vectors as the data frame's columns.

```
df01 = data.frame(v07, v08, v09, v10)
```

```
df01
```

```
v07 v08 v09 v10
```

```
1 1 1 0 TRUE
```

```
2 2 1 -1 FALSE
```

```
3 3 2 -1 FALSE
```

```
4 4 2 -2 FALSE
```

```
5 1 3 2 FALSE
```

```
6 2 3 1 FALSE
```

```
7 3 4 1 FALSE
```

```
8 4 4 0 TRUE
```

We can access the elements in a data frame using the data frame's index, which takes the form [row number(s), column number(s)], where [is the bracket operator.

```
df02 = df01[1, ] # returns all elements in the data frame's first row
```

```
df02
```

```
v07 v08 v09 v10
```

```
1 1 1 0 TRUE
```

```
df03 = df01[ , 3:4] # returns all elements in the data frame's third and fourth columns
```

```
df03
```

```
v09 v10
```

```
1 0 TRUE
```

```
2 -1 FALSE
```

```
3 -1 FALSE
```

```
4 -2 FALSE
```

```
5 2 FALSE
```

```
6 1 FALSE
```

```
7 1 FALSE
```

```
8 0 TRUE
```

```
df04 = df01[4, 3] # returns the element in the data frame's fourth row and third column
```

```
df04
```

```
[1] -2
```

We can also extract a single column from a data frame using the dollar sign (\$) operator to designate the column's name

```
df05 = df01$v08
```

```
df05
```

```
[1] 1 1 2 2 3 3 4 4
```

Note

If you look carefully at the output above you will see that extracting a single row or multiple columns using the `[]` operator returns a new data frame. Extracting a single element from a data frame using the bracket operator, or a single column using the `$` operator returns a vector.

Matrices and Arrays

A matrix is similar to a data frame, but every element in a matrix is of the same type, usually numerical.

```
m01 = matrix(1:10, nrow = 5) # places numbers 1:10 in matrix with five rows, filling by column
```

```
m01
```

```
[,1] [,2]
```

```
[1,] 1 6
```

```
[2,] 2 7
```

```
[3,] 3 8
```

```
[4,] 4 9
```

```
[5,] 5 10
```

```
m02 = matrix(1:10, ncol = 5) # places numbers 1:10 in matrix with five columns, filling by row
```

```
m02
```

```
[,1] [,2] [,3] [,4] [,5]
```

```
[1,] 1 3 5 7 9
```

```
[2,] 2 4 6 8 10
```

A matrix has two dimensions and an array has three or more dimensions.

Lists

A list is an object that holds other objects, even if those objects are of different types.

```
li01 = list(v00, df01, m01)
```

```
li01
```

```
[[1]]
```

```
[1] 1.1 2.2 3.3 4.4
```

```
[[2]]
```

```
v07 v08 v09 v10
```

```
1 1 1 0 TRUE
2 2 1 -1 FALSE
3 3 2 -1 FALSE
4 4 2 -2 FALSE
5 1 3 2 FALSE
6 2 3 1 FALSE
7 3 4 1 FALSE
8 4 4 0 TRUE
```

```
[[3]]
[,1] [,2]
[1,] 1 6
[2,] 2 7
[3,] 3 8
[4,] 4 9
[5,] 5 10
```

Note that the double bracket, such as `[[1]]`, identifies an object in the list and that we can extract values from this list using this notation.

```
li01[[1]] # extract first object stored in the list
[1] 1.1 2.2 3.3 4.4
li01[[1]][1] # extract the first value of the first object stored in the list
[1] 1.1
```

Script Files

Although you can enter commands directly into RStudio's Console Panel and execute them, you will find it much easier to write your commands in a script file and send them to the console line-by-line, as groups of two or more lines, or all at once by sourcing the file. You will make errors as you enter code. When your error is in one line of a multi-line script, you can fix the error and then rerun the script at once without the need to retype each line directly into the console.

To open a script file, select *File: New File: R Script* from the main menu. To save your script file, which will have .R as an extension, select *File: Save* from the main menu and navigate to the folder where you wish to save the file. As an exercise, try entering the following sequence of commands in a script file

```
x1 = runif(1000) # a vector of 1000 values drawn at random from a uniform distribution
x2 = runif(1000) # another vector of 1000 values drawn at random from a uniform distribution
y1 = rnorm(1000) # a vector of 1000 values drawn at random from a normal distribution
y2 = rnorm(1000) # another vector of 1000 values drawn at random from a normal distribution
old.par = par(mfrow = c(2,2)) # create a 2 x 2 grid for plots
plot(x1, x2) # create a scatterplot of two vectors
plot(y1, y2)
```

```
plot(x1, y1)
plot(x2, y2)
par(old.par) # restore the initial plot conditions (more on this later)
```

save it as `test_script.R` and then click the Source button; you should see the following plot appear in the Plot tab.

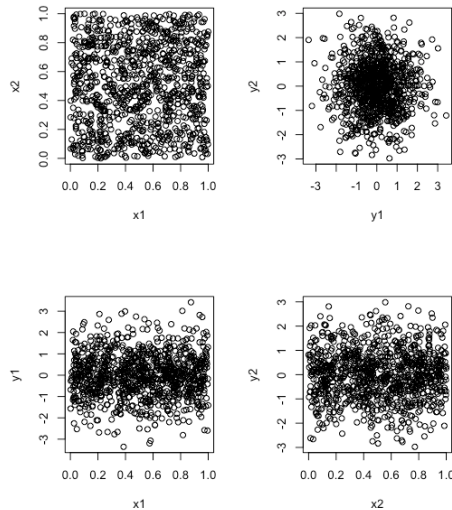


Figure \(\backslash\PageIndex{1}\backslash\): Grid of four scatterplots created by the code in `test_script.R`. The scatterplot in the upper left uses two vectors drawn at random from a uniform distribution. The scatterplot in the upper right uses two vectors drawn at random from a normal distribution. Each of the scatterplots in the bottom row use two vectors, one drawn at random from a uniform distribution and one drawn at random from a normal distribution. (Copyright; author via source)

Loading a Data File and Saving a Data File

Although creating a small vector, data frame, matrix, array, or list is easy, creating one with hundreds of elements or creating dozens of individual data objects is tedious at best; thus, the ability to load data saved during an earlier session, or the ability to read in a spreadsheet file is helpful.

To read in a spreadsheet file saved in .csv format (comma separated values), we use R's `read.csv()` function, which takes the general form

```
read.csv(file)
```

where `file` provides the absolute path to the file. This is easiest to manage if you navigate to the folder where your .csv file is stored using RStudio's file pane and then set it as the working directory by clicking on *More* and selecting *Set As Working Directory*. Download the file "element_data.csv" using this [link](#) and then store the file in a folder on your computer. Navigate to this folder and set it as your working directory. Enter the following line of code

```
elements = read.csv(file = "element_data.csv")
```

to read the file's data into a data frame named `elements`. To view the data frame's structure we use the `head()` function to display the first six rows of data.

```
head(elements)
```

```
name symbol at_no at_wt mp bp phase electronegativity electron_affinity
```

```
1 Hydrogen H 1 1.007940 14.01 20.28 Gas 2.20 72.8
```

```
2 Helium He 2 4.002602 NA 4.22 Gas NA 0.0
```

```
3 Lithium Li 3 6.941000 453.69 1615.15 Solid 0.98 59.6
4 Beryllium Be 4 9.012182 1560.15 2743.15 Solid 1.57 0.0
5 Boron B 5 10.811000 2348.15 4273.15 Solid 2.04 26.7
6 Carbon C 6 12.010700 3823.15 4300.15 Solid 2.55 153.9

block group period at_radius covalent_radius

1 s 1 1 5.30e-11 3.70e-11
2 p 18 1 3.10e-11 3.20e-11
3 s 1 2 1.67e-10 1.34e-10
4 s 2 2 1.12e-10 9.00e-11
5 p 13 2 8.70e-11 8.20e-11
6 p 14 2 6.70e-11 7.70e-11
```

Note that cells in the spreadsheet with missing values appear here as **NA** for not available. The melting points (mp) and boiling points (bp) are in Kelvin, and the electron affinities are in kJ/mol.

You can save to your working directory the contents of data frame by using the `write.csv()` function; thus, we can save a copy of the data in `elements` using the following line of code

```
write.csv(elements, file = "element_data_copy.csv")
```

Another way to save multiple objects is to use the `save()` function to create an .RData file. For example, to save the vectors `v00`, `v01`, and `v02` to a file with the name `vectors.RData`, enter

```
save(v00, v01, v02, file = "vectors.RData")
```

To read in the objects in an .RData file, navigate to the folder that contains the file, click on the file's name and RStudio will ask if you wish to load the file into your session.

Using Packages of Functions

The base installation of R provides many useful functions for working with data. The advantage of these functions is that they work (always a plus) and they are stable (which means they will continue to work even as R is updated to new versions). For the most part, we will rely on R's built in functions for these two reasons. When we need capabilities that are not part of R's base installation, then we must write our own functions or use packages of functions written by others.

To install a package of functions, click on the Packages tab in the *Files, Plots, Packages, Help & Viewer* pane. Click on the button labeled *Install*, enter the name of the package you wish to install, and click on *Install* to complete the installation. You only need to install a package once.

To use a package that is not part of R's base installation, you need to bring it into your current session, which you do with the command `library(name of package)` or by clicking on the checkbox next to the name of the package in the list of your installed packages. Once you have loaded the package into your session, it remains available to you until you quit RStudio.

Managing Your Environment

One nice feature of RStudio is that the *Environment Panel* provides a list of the objects you create. If your environment becomes too cluttered, you can delete items by switching to the *Grid* view, clicking on the check-box next to the object(s) you wish to delete, and then clicking on the broom icon. You can remove all items from the *List* view by simply clicking on the broom icon.

Getting Help

There are extensive help files for R's functions that you can search for using the Help Panel or by using the `help()` command. A help file shows you the command's proper syntax, including the types of values you can pass to the command and their default values, if any—more details on this later—and provides you with some examples of how the command is used. R's help files can be difficult to parse at times; you may find it more helpful to simply use a search engine to look for information about "how to use in R." Another good source for finding help with R is [stackoverflow](#).

1.3: Exercises

1. Gather the following information for the first 18 elements in the periodic table and create a vector for each:

- name
- symbol
- atomic number
- atomic weight
- phase (gas, liquid, solid)
- group number (1–18)
- row number
- atomic radius (in picometers)
- electronegativity
- first ionization potential (in electron volts)

Combine these vectors into a single data frame and save it as a .csv file. In addition, save the data frame and the individual vectors as a single .RData file. You will use these files to complete exercises in some of the chapters that follow.

2: TYPES OF DATA

At the heart of any analysis is data. Sometimes our data describes a category and sometimes it is numerical; sometimes our data conveys order and sometimes it does not; sometimes our data has an absolute reference and sometimes it has an arbitrary reference; and sometimes our data takes on discrete values and sometimes it takes on continuous values. Whatever its form, when we gather data our intent is to extract from it information that can help us solve a problem.

[2.1: Ways to Describe Data](#)

[2.2: Using R to Organize and Manipulate Data](#)

[2.3: Exercises](#)

2.1: Ways to Describe Data

If we are to consider how to describe data, then we need some data with which we can work. Ideally, we want data that is easy to gather and easy to understand. It also is helpful if you can gather similar data on your own so you can repeat what we cover here. A simple system that meets these criteria is to analyze the contents of bags of M&Ms. Although this system may seem trivial, keep in mind that reporting the percentage of yellow M&Ms in a bag is analogous to reporting the concentration of Cu^{2+} in a sample of an ore or water: both express the amount of an analyte present in a unit of its matrix.

At the beginning of this chapter we identified four contrasting ways to describe data: categorical vs. numerical, ordered vs. unordered, absolute reference vs. arbitrary reference, and discrete vs. continuous. To give meaning to these descriptive terms, let's consider the data in Table 2.1.1, which includes the year the bag was purchased and analyzed, the weight listed on the package, the type of M&Ms, the number of yellow M&Ms in the bag, the percentage of the M&Ms that were red, the total number of M&Ms in the bag and their corresponding ranks.

Table 2.1.1. Distribution of Yellow and Red M&Ms in Bags of M&Ms.

bag id	year	weight (oz)	type	number yellow	% red	total M&Ms	rank (for total)
a	2006	1.74	peanut	2	27.8	18	sixth
b	2006	1.74	peanut	3	4.35	23	fourth
c	2000	0.80	plain	1	22.7	22	fifth
d	2000	0.80	plain	5	20.8	24	third
e	1994	10.0	plain	56	23.0	331	second
f	1994	10.0	plain	63	21.9	333	first

The entries in Table 2.1.1 are organized by column and by row. The first row—sometimes called the header row—identifies the variables that make up the data. Each additional row is the record for one sample and each entry in a sample's record provides information about one of its variables; thus, the data in the table lists the result for each variable and for each sample.

Categorical vs. Numerical Data

Of the variables included in Table 2.1.1, some are categorical and some are numerical. A categorical variable provides qualitative information that we can use to describe the samples relative to each other, or that we can use to organize the samples into groups (or categories). For the data in Table 2.1.1, bag id, type, and rank are categorical variables.

A numerical variable provides quantitative information that we can use in a meaningful calculation; for example, we can use the number of yellow M&Ms and the total number of M&Ms to calculate a new variable that reports the percentage of M&Ms that are yellow. For the data in Table 2.1.1, year, weight (oz), number yellow, % red M&Ms, and total M&Ms are numerical variables.

We can also use a numerical variable to assign samples to groups. For example, we can divide the plain M&Ms in Table 2.1.1 into two groups based on the sample's weight. What makes a numerical variable more interesting, however, is that we can use it to make quantitative comparisons between samples; thus, we can report that there are $14.4 \times$ as many plain M&Ms in a 10-oz. bag as there are in a 0.8-oz. bag.

$$\frac{333 + 331}{24 + 22} = \frac{664}{46} = 14.4$$

Although we could classify year as a categorical variable—not an unreasonable choice as it could serve as a useful way to group samples—we list it here as a numerical variable because it can serve as a useful predictive variable in a regression analysis. On the other hand rank is not a numerical variable—even if we rewrite the ranks as numerals—as there are no meaningful calculations we can complete using this variable.

Nominal vs. Ordinal Data

Categorical variables are described as nominal or ordinal. A nominal categorical variable does not imply a particular order; an ordinal categorical variable, on the other hand, conveys a meaningful sense of order. For the categorical variables in Table 2.1.1, bag id and type are nominal variables, and rank is an ordinal variable.

Ratio vs. Interval Data

A numerical variable is described as either ratio or interval depending on whether it has (ratio) or does not have (interval) an absolute reference. Although we can complete meaningful calculations using any numerical variable, the type of calculation we can perform depends on whether or not the variable's values have an absolute reference.

A numerical variable has an absolute reference if it has a meaningful zero—that is, a zero that means a measured quantity of none—against which we reference all other measurements of that variable. For the numerical variables in Table 2.1.1, weight (oz), number yellow, % red, and total M&Ms are ratio variables because each has a meaningful zero; year is an interval variable because its scale is referenced to an arbitrary point in time, 1 BCE, and not to the beginning of time.

For a ratio variable, we can make meaningful absolute and relative comparisons between two results, but only meaningful absolute comparisons for an interval variable. For example, consider sample e, which was collected in 1994 and has 331 M&Ms, and sample d, which was collected in 2000 and has 24 M&Ms. We can report a meaningful absolute comparison for both variables: sample e is six years older than sample d and sample e has 307 more M&Ms than sample d. We also can report a meaningful relative comparison for the total number of M&Ms—there are

$$\frac{331}{24} = 13.8\times$$

as many M&Ms in sample e as in sample d—but we cannot report a meaningful relative comparison for year because a sample collected in 2000 is not

$$\frac{2000}{1994} = 1.003\times$$

older than a sample collected in 1994.

Discrete vs. Continuous Data

Finally, the granularity of a numerical variable provides one more way to describe our data. For example, we can describe a numerical variable as discrete or continuous. A numerical variable is discrete if it can take on only specific values—typically, but not always, an integer value—between its limits; a continuous variable can take on any possible value within its limits. For the numerical data in Table 2.1.1, year, number yellow, and total M&Ms are discrete in that each is limited to integer values. The numerical variables weight (oz) and % red, on the other hand, are continuous variables. Note that weight is a continuous variable even if the device we use to measure weight yields discrete values.

2.2: Using R to Organize and Manipulate Data

The data in Table 2.2.1 should remind you of a data frame, a way of organizing data in R that we introduced in Chapter 1. Here we will learn how to create a data frame that holds the data in Table 2.2.1 and learn how we can make use of the data frame.

Creating a Data Frame

To create a data frame we begin by creating vectors for each of the variables. Note that `letters` is a constant in R that contains the 26 lower case letters of the Roman alphabet: here we are using just the first six letters for the bag ids.

```
bag_id = letters[1:6]
year = c(2006, 2006, 2000, 2000, 1994, 1994)
weight = c(1.74, 1.74, 0.80, 0.80, 10.0, 10.0)
type = c("peanut", "peanut", "plain", "plain", "plain", "plain")
number_yellow = c(2, 3, 1, 5, 56, 63)
percent_red = c(27.8, 4.35, 22.7, 20.8, 23.0, 21.9)
total = c(18, 23, 22, 24, 331, 333)
rank = c("sixth", "fourth", "fifth", "third", "second", "first")
```

To create the data frame, we use R's `data.frame()` function, passing to it the names of our vectors, each of which must be of the same length. There is an option within this function to treat variables whose values are character strings as factors—another name for a categorical variable—by using the argument `stringsAsFactors = TRUE`. As the default value for this argument depends on your version of R, it is useful to make your choice explicit by including it in your code, as we do here.

```
mm_data = data.frame(bag_id, year, weight, type, number_yellow, percent_red, total, rank,
stringsAsFactors = TRUE)
mm_data
```

```
bag_id year weight type number_yellow percent_red total rank
```

```
1 a 2006 1.74 peanut 2 27.80 18 sixth
2 b 2006 1.74 peanut 3 4.35 23 fourth
3 c 2000 0.80 plain 1 22.70 22 fifth
4 d 2000 0.80 plain 5 20.80 24 third
5 e 1994 10.00 plain 56 23.00 331 second
6 f 1994 10.00 plain 63 21.90 333 first
```

If we examine the structure of this data set using R's `str()` function, we see that `bag_id`, `type`, and `rank` are factors and `year`, `weight`, `number_yellow`, `percent_red`, and `total` are numerical variables, assignments that are consistent with our earlier analysis of the data.

```
str(mm_data)

'data.frame': 6 obs. of 8 variables:
 $ bag_id : Factor w/ 6 levels "a","b","c","d",...: 1 2 3 4 5 6
 $ year   : num 2006 2006 2000 2000 1994 ...
 $ weight : num 1.74 1.74 0.8 0.8 10 10
```

```
$ type : Factor w/ 2 levels "peanut","plain": 1 1 2 2 2 2
$ number_yellow: num 2 3 1 5 56 63
$ percent_red : num 27.8 4.35 22.7 20.8 23 21.9
$ total : num 18 23 22 24 331 333
$ rank : Factor w/ 6 levels "fifth","first",...: 5 3 1 6 4 2
```

Finally, we can use the function `as.factor()` to have R treat a numerical variable as a categorical variable, as we do here for year. Why we might wish to do this is a topic we will return to in later chapters.

```
mm_year_as_factor = data.frame(bag_id, as.factor(year), percent_red, total)
str(mm_year_as_factor)
'data.frame': 6 obs. of 4 variables:
 $ bag_id : Factor w/ 6 levels "a","b","c","d",...: 1 2 3 4 5 6
 $ as.factor.year.: Factor w/ 3 levels "1994","2000",...: 3 3 2 2 1 1
 $ percent_red : num 27.8 4.35 22.7 20.8 23 21.9
 $ total : num 18 23 22 24 331 33
```

Creating a New Data Frame by Subsetting an Existing Data Frame

In Chapter 1.2 we learned how to retrieve individual rows or columns from a data frame and assign them to a new object. Here we learn how to use R's more flexible `subset()` function to accomplish the same thing. Here, for example, we retrieve only the data for plain M&Ms.

```
plain_mm = subset(mm_data, type == "plain")
plain_mm
bag_id year weight type number_yellow percent_red total rank
3 c 2000 0.8 plain 1 22.7 22 fifth
4 d 2000 0.8 plain 5 20.8 24 third
5 e 1994 10.0 plain 56 23.0 331 second
6 f 1994 10.0 plain 63 21.9 333 first
```

Note that `type == "plain"` uses a relational operator to choose only those rows in which the variable `type` has the value `plain`. Here is a list of relational operators:

Table 2.2.2. Relational Operators in R.

operator	usage	meaning
<	<code>x < y</code>	x is less than y
>	<code>x > y</code>	x is greater than y
<=	<code>x <= y</code>	x is less than or equal to y
>=	<code>x >= y</code>	x is greater than or equal to y

operator	usage	meaning
==	x == y	x is exactly equal to y
!=	x != y	x is not equal to y

We can string variables together using the logical & operator.

```
mm_plain10 = subset(mm_data, (weight == 10.0 & type == "plain"))
```

```
mm_plain10
```

```
bag_id year weight type number_yellow percent_red total rank
```

```
5 e 1994 10 plain 56 23.0 331 second
```

```
6 f 1994 10 plain 63 21.9 333 first
```

We also can narrow the number of variables returned using the `subset()` function's `select` argument. In this example we exclude samples collected before the year 2000 and return only the year, the number of yellow M&Ms, and the percentage of red M&Ms.

```
mm_20xx = subset(mm_data, year >= 2000, select = c(year, number_yellow, percent_red))
```

```
mm_20xx
```

```
year number_yellow percent_red
```

```
1 2006 2 27.80
```

```
2 2006 3 4.35
```

```
3 2000 1 22.70
```

```
4 2000 5 20.80
```

2.3: Exercises

1. In Exercise 1 of Chapter 1 you created a data frame with the following information about the first 18 elements.

- name
- symbol
- atomic number
- atomic weight
- phase (gas, liquid, solid)
- group number (1–18)
- row number
- atomic radius (in picometers)
- electronegativity
- first ionization potential (in electron volts)

(a) Setting aside name and symbol, which of the remaining variables are categorical or numerical?

(b) For those variables that are categorical, which are nominal and which are ordinal?

(c) For those variables that are numerical, which are ratio and which are interval?

(d) For those variables that are numerical, which are discrete and which are continuous?

2. Use this [link](#) to download and save the spreadsheet `marlybone_2018.csv`. The data in this file gives the daily average level of NOX (the combined concentrations of NO and of NO₂) in $\mu\text{g}/\text{m}^3$ and the daily average temperature in $^{\circ}\text{C}$ as recorded in 2018 at a roadside monitoring station located on Marylebone Road in Westminster, which is near Regents Park, Madame Tussaud's Wax Museum, and Baker Street, the "home" of Sherlock Holmes. The data is made available by London Air, a website managed by Kings College in London that reports results from the continuous monitoring of air quality at hundreds of sites spread throughout the greater London area. As in most long-term monitoring project, some data is missing for various reasons, such as equipment failure; these values appear in the spreadsheet as empty cells. If you wish, you can visit the London Air web site [here](#).

(a) Use the `read.csv()` function to bring the data into R as a data frame and examine the dataset's structure using the `head()` function.

(b) Add a new column to the data frame that contains the running day number (January 1st is day 1 and December 31st is day 365).

(c) Use the `subset()` function to create separate data frames for each month.

(d) Save all of your data frames in a single `.RData` file so that it is available to you when working problems in other chapters.

3. Use this [link](#) to access a case study on data analysis and complete the five investigations included in Part I: Ways to Describe Data.

3: VISUALIZING DATA

The old saying that "a picture is worth a 1000 words" may not be universally true, but it true when it comes to the analysis of data. A good visualization of data, for example, allows us to see patterns and relationships that are less evident when we look at data arranged in a table, and it provides a powerful way to tell our data's story. One of R's significant strengths as a statistical programming language is the ease with which we can generate useful visualizations.

[3.1: Types of Visualizations](#)

[3.2: Using R to Visualize Data](#)

[3.3: Creating Plots From Scratch in R Using Base Graphics](#)

[3.4: Exercises](#)

3.1: Types of Visualizations

Suppose we want to study the composition of 1.69-oz (47.9-g) packages of plain M&Ms. We obtain 30 bags of M&Ms (ten from each of three stores) and remove the M&Ms from each bag one-by-one, recording the number of blue, brown, green, orange, red, and yellow M&Ms. We also record the number of yellow M&Ms in the first five candies drawn from each bag, and record the actual net weight of the M&Ms in each bag. Table 3.1.1 summarizes the data collected on these samples. The bag id identifies the order in which the bags were opened and analyzed.

Table 3.1.1. Analysis of Plain M&Ms in 47.9 g Bags.

bag	store	blue	brown	green	orange	red	yellow	yellow_first_five	net_weight
1	CVS	3	18	1	5	7	23	2	49.287
2	CVS	3	14	9	7	8	15	0	48.870
3	Target	4	14	5	10	10	16	1	51.250
4	Kroger	3	13	5	4	15	16	0	48.692
5	Kroger	3	16	5	7	8	18	1	48.777
6	Kroger	2	12	6	10	17	7	1	46.405
7	CVS	13	11	2	8	6	17	1	49.693
8	CVS	13	12	7	10	7	8	2	49.391
9	Kroger	6	17	5	4	8	16	1	48.196
10	Kroger	8	13	2	5	10	17	1	47.326
11	Target	9	20	1	4	12	13	3	50.974
12	Target	11	12	0	8	4	23	0	50.081
13	CVS	3	15	4	6	14	13	2	47.841
14	Kroger	4	17	5	6	14	10	2	48.377
15	Kroger	9	13	3	8	14	8	0	47.004
16	CVS	8	15	1	10	9	15	1	50.037
17	CVS	10	11	5	10	7	13	2	48.599
18	Kroger	1	17	6	7	11	14	1	48.625
19	Target	7	17	2	8	4	18	1	48.395
20	Kroger	9	13	1	8	7	22	1	51.730
21	Target	7	17	0	15	4	15	3	50.405
22	CVS	12	14	4	11	9	5	2	47.305
23	Target	9	19	0	5	12	12	0	49.477
24	Target	5	13	3	4	15	16	0	48.027
25	CVS	7	13	0	4	15	16	2	48.212
26	Target	6	15	1	13	10	14	1	51.682
27	CVS	5	17	6	4	8	19	1	50.802
28	Kroger	1	21	6	5	10	14	0	49.055
29	Target	4	12	6	5	13	14	2	46.577

bag	store	blue	brown	green	orange	red	yellow	yellow_first_five	net_weight
30	Target	15	8	9	6	10	8	1	48.317

Having collected our data, we next examine it for possible problems, such as missing values (Did we forget to record the number of brown M&Ms in any of our samples?), for errors introduced when we recorded the data (Is the decimal point recorded incorrectly for any of the net weights?), or for unusual results (Is it really the case that this bag has only yellow M&M?). We also examine our data to identify interesting observations that we may wish to explore (It appears that most net weights are greater than the net weight listed on the individual packages. Why might this be? Is the difference significant?) When our data set is small we usually can identify possible problems and interesting observations without much difficulty; however, for a large data set, this becomes a challenge. Instead of trying to examine individual values, we can look at our results visually. While it may be difficult to find a single, odd data point when we have to individually review 1000 samples, it often jumps out when we look at the data using one or more of the approaches we will explore in this chapter.

Dot Plots

A dot plot displays data for one variable, with each sample's value plotted on the x-axis. The individual points are organized along the y-axis with the first sample at the bottom and the last sample at the top. Figure 3.1.1 shows a dot plot for the number of brown M&Ms in the 30 bags of M&Ms from Table 3.1.1. The distribution of points appears random as there is no correlation between the sample id and the number of brown M&Ms. We would be surprised if we discovered that the points were arranged from the lower-left to the upper-right as this implies that the order in which we open the bags determines whether they have many or a few brown M&Ms.

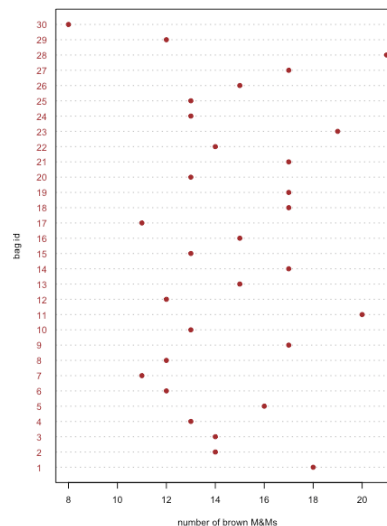


Figure 3.1.1: Dot plot for the brown M&Ms in each of the 30 bags included in Table 3.1.1.

Stripcharts

A dot plot provides a quick way to give us confidence that our data are free from unusual patterns, but at the cost of space because we use the y-axis to include the sample id as a variable. A stripchart uses the same x-axis as a dot plot, but does not use the y-axis to distinguish between samples. Because all samples with the same number of brown M&Ms will appear in the same place—making it impossible to distinguish them from each other—we stack the points vertically to spread them out, as shown in Figure 3.1.2.



Figure 3.1.2: Stripchart for the brown M&Ms in each of the 30 bags included in Table 3.1.1.

Both the dot plot in Figure 3.1.1 and the stripchart in Figure 3.1.2 suggest that there is a smaller density of points at the lower limit and the upper limit of our results. We see, for example, that there is just one bag each with 8, 16, 18, 19, 20, and 21 brown M&Ms, but there are six bags each with 13 and 17 brown M&Ms.

Because a stripchart does not use the y -axis to provide meaningful categorical information, we can easily display several stripcharts at once. Figure 3.1.3 shows this for the data in Table 3.1.1. Instead of stacking the individual points, we jitter them by applying a small, random offset to each point. Among the things we learn from this stripchart are that only brown and yellow M&Ms have counts of greater than 20 and that only blue and green M&Ms have counts of three or fewer M&Ms.

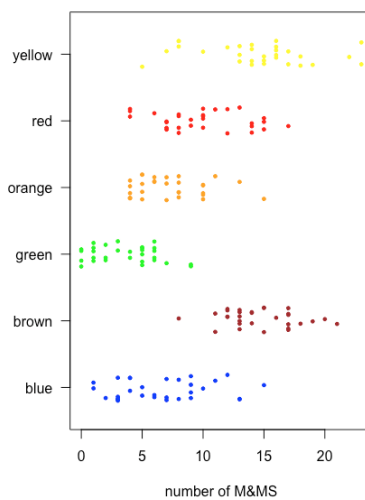


Figure 3.1.3: Stripcharts for each color of M&Ms in each of the 30 bags included in Table 3.1.1.

Box and Whisker Plots

The stripchart in Figure 3.1.3 is easy for us to examine because the number of samples, 30 bags, and the number of M&Ms per bag is sufficiently small that we can see the individual points. As the density of points becomes greater, a stripchart becomes less useful. A box and whisker plot provides a similar view but focuses on the data in terms of the range of values that encompass the middle 50% of the data.

Figure 3.1.4 shows the box and whisker plot for brown M&Ms using the data in Table 3.1.1. The 30 individual samples are superimposed as a stripchart. The central box divides the x-axis into three regions: bags with fewer than 13 brown M&Ms (seven samples), bags with between 13 and 17 brown M&Ms (19 samples), and bags with more than 17 brown M&Ms (four samples). The box's limits are set so that it includes at least the middle 50% of our data. In this case, the box contains 19 of the 30 samples (63%) of the bags, because moving either end of the box toward the middle results in a box that includes less than 50% of the samples. The difference between the box's upper limit (19) and its lower limit (13) is called the interquartile range (IQR). The thick line in the box is the median, or middle value (more on this and the IQR in the next chapter). The dashed lines at either end of the box are called whiskers, and they extend to the largest or the smallest result that is within $\pm 1.5 \times \text{IQR}$ of the box's right or left edge, respectively.

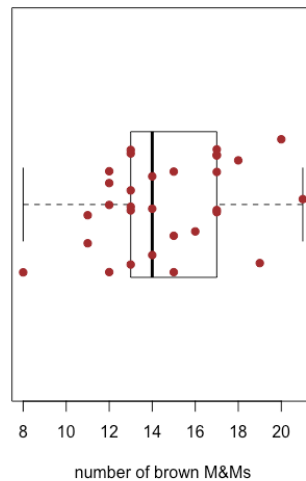


Figure 3.1.4: Box-and-whisker plot for the brown M&Ms in each of the 30 bags included in Table 3.1.1 showing individual samples as a jittered stripchart.

Because a box and whisker plot does not use the y-axis to provide meaningful categorical information, we can easily display several plots in the same frame. Figure 3.1.5 shows this for the data in Table 3.1.1. Note that when a value falls outside of a whisker, as is the case here for yellow M&Ms, it is flagged by displaying it as an open circle.

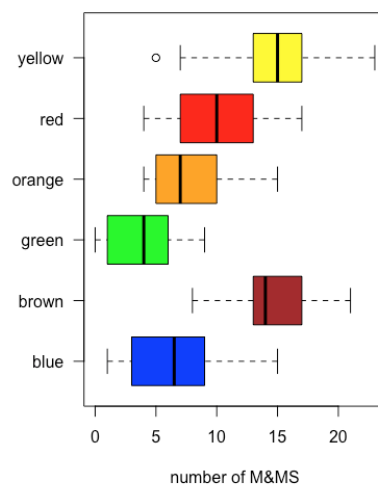


Figure 3.1.5: Box-and-whisker plots for each of the 30 bags included in Table 3.1.1 organized by color.

One use of a box and whisker plot is to examine the distribution of the individual samples, particularly with respect to symmetry. With the exception of the single sample that falls outside of the whiskers, the distribution of yellow M&Ms appears symmetrical:

the median is near the center of the box and the whiskers extend equally in both directions. The distribution of the orange M&Ms is asymmetrical: half of the samples have 4–7 M&Ms (just four possible outcomes) and half have 7–15 M&Ms (nine possible outcomes), suggesting that the distribution is skewed toward higher numbers of orange M&Ms (see Chapter 5 for more information about the distribution of samples).

Figure 3.1.6 shows box-and-whisker plots for yellow M&Ms grouped according to the store where the bags of M&Ms were purchased. Although the box and whisker plots are quite different in terms of the relative sizes of the boxes and the relative length of the whiskers, the dot plots suggest that the distribution of the underlying data is relatively similar in that most bags contain 12–18 yellow M&Ms and just a few bags deviate from these limits. These observations are reassuring because we do not expect the choice of store to affect the composition of bags of M&Ms. If we saw evidence that the choice of store affected our results, then we would look more closely at the bags themselves for evidence of a poorly controlled variable, such as type (Did we accidentally purchase bags of peanut butter M&Ms from one store?) or the product's lot number (Did the manufacturer change the composition of colors between lots?).

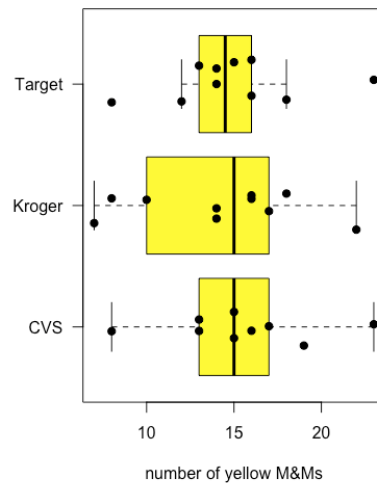


Figure 3.1.6: Box-and-whisker plots for yellow M&Ms for each of the 30 bag in Table 3.1.1 organized by the store where the bags were purchased.

Bar Plots

Although a dot plot, a stripchart and a box-and-whisker plot provide some qualitative evidence of how a variable's values are distributed—we will have more to say about the distribution of data in Chapter 5—they are less useful when we need a more quantitative picture of the distribution. For this we can use a bar plot that displays a count of each discrete outcome. Figure 3.1.7 shows bar plots for orange and for yellow M&Ms using the data in Table 3.1.1.

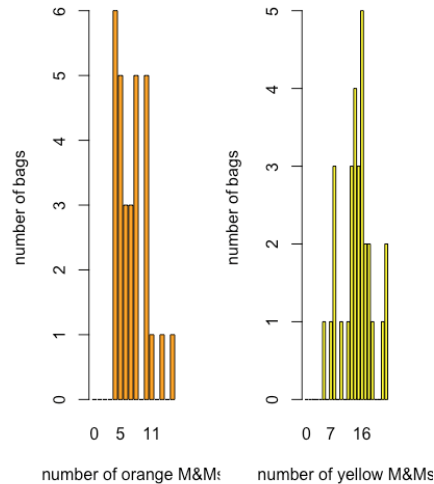


Figure 3.1.7: Bar plots for orange M&Ms and yellow M&Ms using the data in Table 3.1.1.

Here we see that the most common number of orange M&Ms per bag is four, which is also the smallest number of orange M&Ms per bag, and that there is a general decrease in the number of bags as the number of orange M&M per bag increases. For the yellow M&Ms, the most common number of M&Ms per bag is 16, which falls near the middle of the range of yellow M&Ms.

Histograms

A bar plot is a useful way to look at the distribution of discrete results, such as the counts of orange or yellow M&Ms, but it is not useful for continuous data where each result is unique. A histogram, in which we display the number of results that fall within a sequence of equally spaced bins, provides a view that is similar to that of a bar plot but that works with continuous data. Figure 3.1.8, for example, shows a histogram for the net weights of the 30 bags of M&Ms in Table 3.1.1. Individual values are shown by the vertical hash marks at the bottom of the histogram.

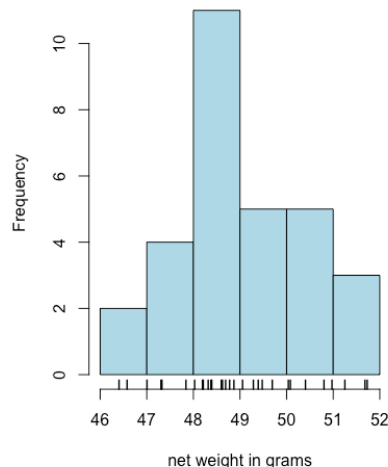


Figure 3.1.8: Histogram of net weights for the data in Table 3.1.1. There are, for example, four bags of M&Ms with net weights between 47 g and 48 g.

This page titled [3.1: Types of Visualizations](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [David Harvey](#).

3.2: Using R to Visualize Data

One of the strengths of R is the ease with which you can plot data and the quality of the plots you can create. R has two pre-installed graphing packages: one is the `graphics` package, which is available to you when you launch R, and the second is the `lattice` package that you can bring into your session by running `library(lattice)` in the console—and there are many additional graphics packages, such as `ggplot2`, developed by others. As our interest in this textbook is making R quickly and easily accessible, we will rely on R's base graphics. See this chapter's resources for a list of other graphing packages.

Note

This section uses the M&M data in Table 1 of Chapter 3.1. You can download a copy of the data as a .csv spreadsheet using this [link](#), and save it in your working directory.

Bringing Your Data Into R

Before we can create a visualization, we need to make our data available to R. The code below uses the `read.csv()` function to read in the file `MandM.csv` as a data frame with the name `mm_data`. The text "`MandM.csv`" assumes the file is located in your working directory.

```
mm_data = read.csv("MandM.csv")
```

Creating a Dot Plot Using R

To create a dot plot in R we use the function `dotchart(x, ...)` where `x` is the object that holds our data, typically a vector or a single column from a data frame, and `...` is a list of optional arguments that affects what we see. In the example below, `pch` sets the plotting symbol (19 is an solid circle), `col` is the color assigned to the plotting symbol, `labels` identifies the samples by name along the y-axis, `xlab` assigns a label to the x-axis, `ylab` assigns a label to the y-axis, and `cex` controls the size of the labels and points. See the last section of this chapter for a more general introduction to creating and displaying plots using R's base graphics.

```
dotchart(mm_data$brown, pch = 19, col = "brown", labels = mm_data$bag, xlab = "number of brown M&Ms", ylab = "bag id", cex = 0.5)
```

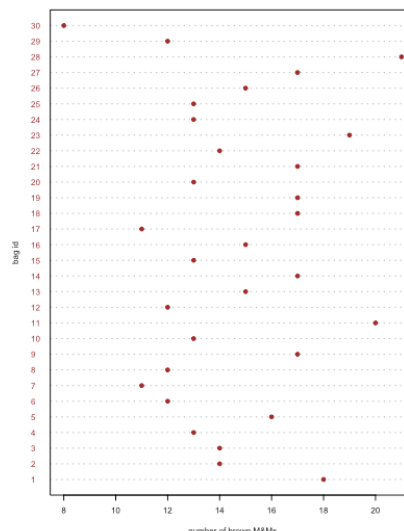


Figure 3.2.1: Example of a dot plot created using R's `dotchart()` function.

Creating a Stripchart Using R

To create a stripchart in R we use the function `stripchart(x, ...)` where `x` is the object that holds our data, typically a vector or a column from a data frame, and `...` is a list of optional arguments that affects what we see. In the example below, `pch` sets the plotting symbol (19 is an solid circle), `col` is the color assigned to the plotting symbol, `method` defines how points with the same value for `x` are displayed on the y-axis, in this case stacking them one above the other by an amount defined by an `offset`, and `cex` controls the size of the individual data points.

```
stripchart(mm_data$brown, pch = 19, col = "brown", method = "stack", offset = 0.5, cex = 0.6,
xlab = "number of brown M&Ms")
```



Figure 3.2.2: Example of a stripchart created using R's `stripchart()` function.

Because a stripchart does not use the y-axis to provide information, we can easily display several stripcharts at once, as shown in the following example, where we use `mm_data[3:8]` to identify the data for each stripchart and `col` to assign a color to each stripchart. Instead of stacking the individual points, they are jittered by applying a small, random offset to each point using `jitter`. The parameter `las` forces the labels to be displayed horizontally (`las = 0` aligns labels parallel to the axis, `las = 1` aligns labels horizontally, `las = 2` aligns labels perpendicular to the axis, and `las = 4` aligns labels vertically).

```
stripchart(mm_data[3:8], pch = 19, cex = 0.5, xlab = "number of M&MS", col = c("blue", "brown",
"green", "orange", "red", "yellow"), method = "jitter", jitter = 0.2, las = 1)
```

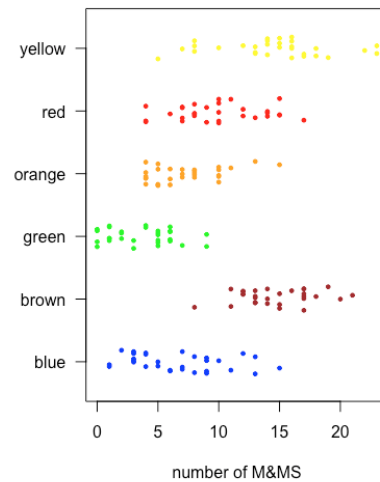


Figure 3.2.3: Example of a stripchart for multiple sets of data created using R's `stripchart()` function.

Creating a Box-and-Whisker Plot Using R

To create a box-and-whisker plot in R we use the function `boxplot(x, ...)` where `x` is the object that holds our data, typically a vector or a column from a data frame, and `...` is a list of optional arguments that affects what we see. In the example below, the option `horizontal = TRUE` overrides the default, which is to display a vertical boxplot, and `range` specifies the length of the whisker as a multiple of the IQR. In this example, we also show the individual values using `stripchart()` with the option `add = TRUE` to overlay the stripchart on the boxplot.

```
boxplot(mm_data$brown, horizontal = TRUE, range = 1.5, xlab = "number of brown M&Ms")
```

```
stripchart(mm_data$brown, method = "jitter", jitter = 0.2, add = TRUE, col = "brown", pch = 19)
```

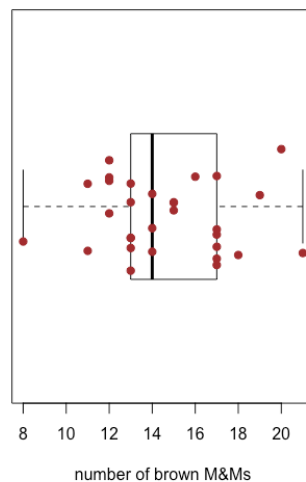


Figure 3.2.4: Example of a box-and-whisker plot created using R's `boxplot()` function. A stripchart of the data is overlaid on the box-and-whisker plot using the `stripchart()` function.

Because a box and whisker plot does not use the y-axis to provide information, we can easily display several plots at once, as shown in the following example, where we use `mm_data[3:8]` to identify the data for each plot and `col` to assign a color to each plot.

```
boxplot(mm_data[3:8], xlab = "number of M&MS", las = 1, horizontal = TRUE, col = c("blue",
"brown", "green", "orange", "red", "yellow"))
```

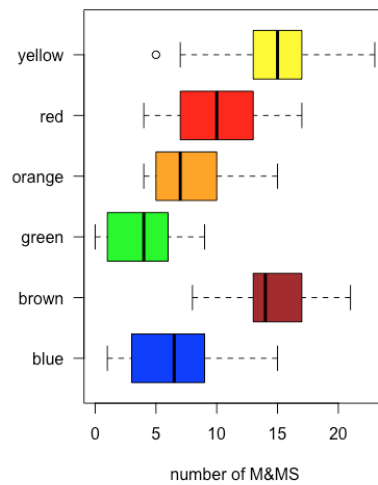


Figure 3.2.5: Example of box-and-whisker plots for multiple sets of data created using R's `boxplot()` function.

In the example below, the code `mm_data$yellow ~ mm_data$store` is a formula, which takes the general form of y as a function of x ; in this case, it uses the data in the column named `store` to divide the data into three groups. The option `outline = FALSE` in the `boxplot()` function suppresses the function's default to plot an open circle for each sample that lies outside of the whiskers; by doing this we avoid plotting these points twice.

```
boxplot(mm_data$yellow ~ mm_data$store, horizontal = TRUE, las = 1, col = "yellow", outline =
FALSE, xlab = "number of yellow M&Ms")stripchart(mm_data$yellow ~ mm_data$store, add = TRUE,
pch = 19, method = "jitter", jitter = 0.2)
```

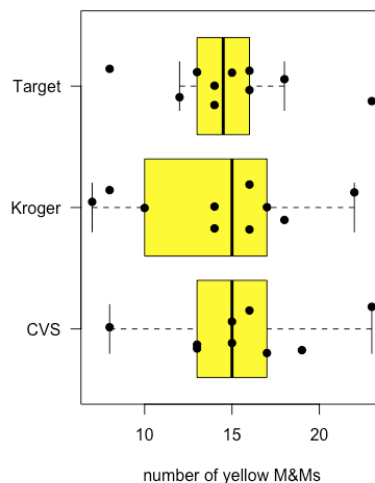


Figure 3.2.6: Example of using a formula to subset the data into three groups based on the store where the samples were purchased.

Note

See Chapter 8.5 for a discussion of the use of formulas in R.

Creating a Bar Plot Using R

To create a bar plot in R we use the function `barplot(x, ...)` where `x` is the object that holds our data, typically a vector or a column from a data frame and `...` is a list of optional arguments that affects what we see. Unlike the previous plots, we cannot pass to `barplot()` our raw data that consists of the number of orange M&Ms in each bag. Instead, we have to provide the data in the form of a table that gives the number of bags that contain 0, 1, 2, ... up to the maximum number of orange M&Ms in any bag; we accomplish this using the `tabulate()` function. Because `tabulate()` only counts the frequency of positive integers, it will ignore any bags that do not have any orange M&Ms; adding one to each count by using `mm_data$orange + 1` ensures they are counted. The argument `names.arg` allows us to provide categorical labels for the x-axis (and correct for the fact that we increased each index by 1).

```
orange_table = tabulate(mm_data$orange + 1)
barplot(orange_table, col = "orange", names.arg = seq(0, max(mm_data$orange), 1), xlab =
"number of orange M&Ms", ylab = "number of bags")
```

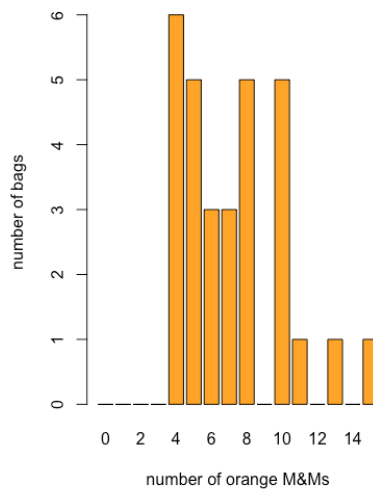


Figure 3.2.7: Example of a bar plot created using R's `barplot()` function.

Creating a Histogram Using R

To create a histogram in R we use the function `hist(x, ...)` where `x` is the object that holds our data, typically a vector or a column from a data frame, and `...` is a list of optional arguments that affects what we see. In the example below, the option `main = NULL` suppresses the placing of a title above the plot, which otherwise is included by default. The option `right = TRUE` means the right-most value of a bin is included in that bin. Finally, although a histogram shows how individual values are distributed, it does not show the individual values themselves. The `rug(x)` function adds tick marks along the x-axis that show each individual value.

```
hist(mm_data$net_weight, col = "lightblue", xlab = "net weight of M&Ms (oz)", right = TRUE,
main = NULL)
```

```
rug(mm_data$net_weight, lwd = 1.5)
```

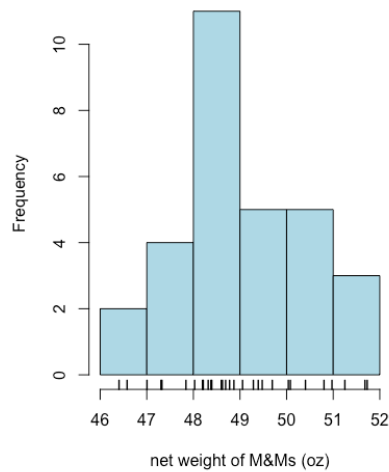


Figure 3.2.8: Example of a histogram created using R's `hist()` and `rug()` functions.

By default, R uses an algorithm to determine how to set the size of bins. As shown in the following example, we can use the option `breaks` to specify the values of `x` where one bin ends and the next bin begins.

```
hist(mm_data$net_weight, col = "lightblue", xlab = "net weight of M&Ms (oz)", breaks = seq(46, 52, 0.5), right = TRUE, main = NULL)
```

```
rug(mm_data$net_weight, lwd = 1.5)
```

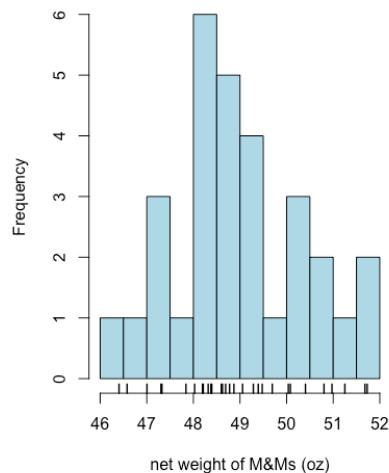


Figure 3.2.9: Example showing how to use the `breaks` command to control the bins used to construct a histogram using R's `hist()` function.

3.3: Creating Plots From Scratch in R Using Base Graphics

As we saw in the last section, the functions to create dot charts, stripcharts, boxplots, barplots, and histograms have arguments that we can use to alter the appearance of the function's output. For example, here is the full list of arguments available when we use `dotchart()` that control what the plot shows.

```
dotchart(x, labels = NULL, groups = NULL, gdata = NULL, cex = par("cex"), pt.cex = cex, pch = 21, gpch = 21, bg = par("bg"), color = par("fg"), gcolor = par("fg"), lcolor = "gray", xlim = range(x[is.finite(x)]), main = NULL, xlab = NULL, ylab = NULL, ...)
```

Each of the arguments has a default value, which means we need not specify the value for an argument unless we wish to change its value, as we did when we set `pch` to 19. The final argument of `...` indicates that we can change any of a long list of graphical parameters that control what we see when we use `dotchart`.

Creating a Simple Scatterplot Using R

One of the most common, and most important, visualizations in analytical chemistry is a scatterplot in which we are interested in the relationship, if any, between two measurement by plotting the values for one variable along the x-axis and the values for the other variable along y-axis. For this exercise, we will use some data from the Puget Sound Data Hoard that gives the mass and the diameter for 816 M&Ms obtained from a 14.0-oz bag of plain M&Ms, a 12.7-oz bag of peanut M&Ms, and a 12.7-oz bag of peanut butter M&Ms. Let's read the data into R and store it in a data frame with the name `psmm_data`. You can download a copy of the data using this [link](#) saving it in your working directory.

```
psmm_data = read.csv("data/PugetSoundM&MData.csv")
```

We might expect that as the diameter of an M&M increases so will the mass of the M&M. We might also expect that the relationship between diameter and mass may depend on whether the M&Ms are plain, peanut, or peanut butter. So that we can access data for each type of M&M, let's use the `which()` function to create vectors that designate the row numbers for each of the three types of M&Ms.

```
pb_id = which(psmm_data$type == "peanut butter")
```

```
plain_id = which(psmm_data$type == "plain")
```

```
peanut_id = which(psmm_data$type == "peanut")
```

Typically we are interested in how one variable affects the other variable. We call the former the independent variable and place it on the x-axis and we call the latter the dependent variable and place it on the y-axis. Here we will use diameter as the independent variable and mass as the dependent variable. To create a scatterplot for the plain M&Ms we use the function `plot(x, y)` where `x` is the data to plot on the x-axis and `y` is the data to plot on the y-axis.

```
plot(x = psmm_data$diameter[plain_id], y = psmm_data$mass[plain_id])
```

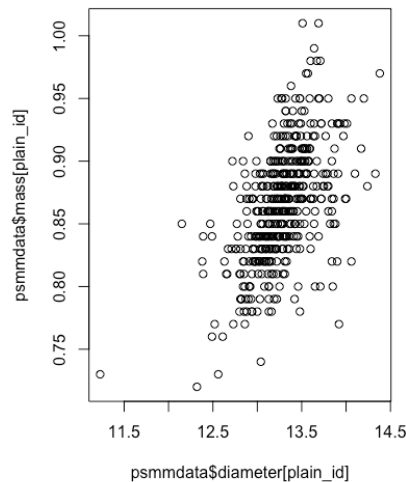


Figure 3.3.1: A simple scatterplot created using R's `plot()` function.

Customizing a Plot Created Using R

Although our scatterplot shows that the mass of a plain M&M increases as its diameter increases, it is not a particularly attractive plot. In addition to specifying x and y , the plot function allows us to pass additional arguments to customize our plot; here are some of these optional arguments:

type = "option". This argument specifies how points are displayed; there are a number of options, but the most useful are "p" for points (this is the default), "l" for lines without points, "b" for both points and lines that do not touch the points, "o" for points and lines that pass through the points, "h" for histogram-like vertical lines, and "s" for stair steps; use "n" if you wish to suppress the points.

pch = number. This argument selects the symbol used to plot the data, with the number assigned to each symbol shown below. The default option is 1, or an open circle. Symbols 15–20 are filled using the color of the symbol's boundary, and symbols 21–25 can take a background color that is different from the symbol's boundary. See later in this document for more details about setting colors. The figure below shows the different options.

```
# code from http://www.sthda.com/english/wiki/r-...available-in-r
oldPar = par()
par(font = 2, mar = c(0.5, 0, 0, 0))
y = rev(c(rep(1, 6), rep(2, 5), rep(3, 5), rep(4, 5), rep(5, 5)))
x = c(rep(1:5, 5), 6)
plot(x, y, pch = 0:25, cex = 1.5, ylim = c(1, 5.5), xlim = c(1, 6.5),
axes = FALSE, xlab = "", ylab = "", bg = "blue")
text(x, y, labels = 0:25, pos = 3)
par(mar = oldPar$mar, font = oldPar$font)
```

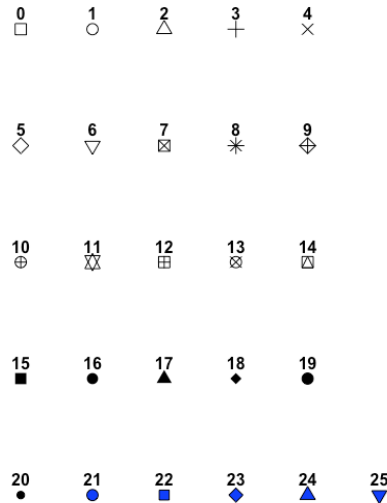


Figure 3.3.2: The 25 `pch` symbols available in R's base graphics.

`lty = number`. This argument specifies the type of line to draw; the options are 1 for a solid line (this is the default), 2 for a dashed line, 3 for a dotted line, 4 for a dot-dash line, 5 for a long-dash line, and 6 for a two-dash line.

`lwd = number`. This argument sets the width of the line. The default is 1 and any other entry simply scales the width relative to the default; thus `lwd = 2` doubles the width and `lwd = 0.5` cuts the width in half.

`bty = "option"`. This argument specifies the type of box to draw around the plot; the options are "o" to draw all four sides (this is the default), "l" to draw on the left side and the bottom side only, "7" to draw on the top side and the right side only, "c" to draw all but the right side, "u" to draw all but the top side, "j" to draw all but the left side, and "n" to omit all four sides.

`axes = logical`. This argument indicates whether the axes are drawn (TRUE) or not drawn (FALSE); the default is TRUE.

`xlim = c(begin, end)`. This argument sets the limits for the x-axis, overriding the default limits set by the `plot()` command.

`ylim = c(begin, end)`. This argument sets the limits for the y-axis, overriding the default limits set by the `plot()` command.

`xlab = "text"`. This argument specifies the label for the x-axis, overriding the default label set by the `plot()` command.

`ylab = "text"`. This argument specifies the label for the y-axis, overriding the default label set by the `plot()` command.

`main = "text"`. This argument specifies the main title, which is placed above the plot, overriding the default title set by the `plot()` command.

`sub = "text"`. This argument specifies the subtitle, which is placed below the plot, overriding the default subtitle set by the `plot()` command.

`cex = number`. This argument controls the relative size of the symbols used to plot points. The default is 1 and any other entry simply scales the size relative to the default; thus `cex = 2` doubles the size and `cex = 0.5` cuts the size in half.

`cex.axis = number`. This argument controls the relative size of the text used for the scale on both axes; see the entry above for `cex` for more details.

`cex.lab = number`. This argument controls the relative size of the text used for the label on both axes; see the entry above for `cex` for more details.

`cex.main = number`. This argument controls the relative size of the text used for the plot's main title; see the entry above for `cex` for more details.

`cex.sub = number`. This argument controls the relative size of the text used for the plot's subtitle; see the entry above for `cex` for more details.

more details.

col = number or “string”. This argument controls the color of the symbols used to plot points. There are 657 available colors, for which the default is “black” or 24. You can see a list of colors (number and text string) by typing `colors()` in the console.

col.axis = number or “string”. This argument controls the color of the text used for the scale on both axes; see the entry above for **col** for more details.

col.lab = number or “string”. This argument controls the color of the text used for the label on both axes; see the entry above for **col** for more details.

col.main = number or “string”. This argument controls the color of the text used for the plot’s main title; see the entry above for **col** for more details.

col.sub = number or “string”. This argument controls the color of the text used for the plot’s subtitle; see the entry above for **col** for more details.

bg = number or “string”. This argument sets the background color for the plot symbols 21–25; see the entries above for **pch** and for **col** for more details.

Let’s use some of these arguments to improve our scatterplot by adding some color to and adjusting the size of the symbols used to plot the data, and by adding a title and some more informative labels for the two axes.

```
plot(x = psmm_data$diameter[plain_id], y = psmm_data$mass[plain_id], xlab = "diameter of M&Ms",
     ylab = "mass of M&Ms", main = "Diameter and Mass of Plain M&Ms", pch = 19, cex = 0.5, col =
     "blue")
```

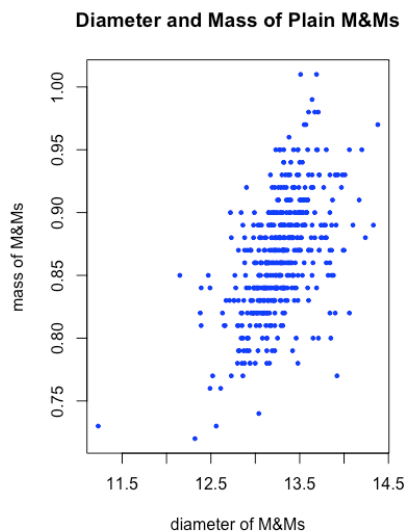


Figure 3.3.3: An improved version of the scatterplot from Figure 3.3.1.

Modifying an Existing Plot Created Using R

We can modify an existing plot in a number of useful ways, such as adding a new set of data, adding a reference line, adding a legend, adding text, and adding a set of grid lines; here are some of the things we can do:

points(x, y, ...). This command is identical to the `plot()` command, but overlays the new points on the current plot instead of first erasing the previous plot. Note: the `points()` command can not re-scale the axes; thus, you must ensure that your original plot—created using the `plot()` command—has x-axis and y-axis limits that meet your needs.

abline(h = number, ...). This command adds a horizontal line at `y = number` with the line’s color, type, and size set using the

optional arguments.

abline(v = number, ...). This command adds a vertical line at $x = \text{number}$ with the line's color, type, and size set using the optional arguments.

abline(b = number, a = number, ...). This command adds a diagonal line defined by a slope (b) and a y -intercept (a); the line's color, type, and size are set using the optional arguments. As we will see in Chapter 8, this is a useful command for displaying the results of a linear regression.

legend(location, legend, ...). This command adds a legend to the current plot. The location is specified in one of two ways:

- by giving the x and y coordinates for the legend's upper-left corner using $x = \text{number}$ and $y = \text{number}$
- by using `location = "keyword"` where the keyword is one of "topleft", "top", "topright", "right", "bottomright", "bottom", "bottomleft", or "left"; the optional argument `inset = number` moves the legend in from the margin when using a keyword (it takes a value from 0 to 1 as a fraction of the plot's area; the default is 0)

The legend is added as a vector of character strings (one for each item in the legend), and any accompanying formatting, such as plot symbols, lines, or colors, are passed along as vectors of the same length; look carefully at the example at the end of this section to see how this command works.

text(location, label, ...). This command adds the text given by "`label`" to the current plot. The location is specified by providing values for x and y using $x = \text{number}$ and $y = \text{number}$. By default, the text is centered at its location; to set the text so that it is left-justified (which is easier to work with), add the argument `adj = c(0, NA)`.

grid(col, lty, lwd). This command adds a set of grid lines to the plot using the color, line type, and line width defined by "`col`", "`lty`", and "`lwd`", respectively.

Here is an example of a figure in which we show how the diameter and mass vary as a function of the type of M&Ms, add a legend, add a grid, and add some text that identifies the source of the data. Note the use of the functions `max` and `min` to identify the limits needed to display results for all of the data.

```
# determine minimum and maximum values for diameter and mass so that we can
# set limits for the x-axis and y-axis that will allow plotting of all data

xmax = max(psmm_data$diameter)
xmin = min(psmm_data$diameter)
ymax = max(psmm_data$mass)
ymin = min(psmm_data$mass)

# create the initial plot using data for plain M&Ms, xlim and ylim values
# ensure plot window will allow plotting of all data

plot(x = psmm_data$diameter[plain_id], y = psmm_data$mass[plain_id], xlab = "diameter of M&Ms",
     ylab = "mass of M&Ms", main = "Diameter and Mass of M&Ms", pch = 19, cex = 0.65, col = "red",
     xlim = c(xmin, xmax), ylim = c(ymin, ymax))

# add the data for the peanut and peanut butter M&Ms using points()

points(x = psmm_data$diameter[peanut_id], y = psmm_data$mass[peanut_id], pch = 18, col =
"brown", cex = 0.65)

points(x = psmm_data$diameter[pb_id], y = psmm_data$mass[pb_id], pch = 17, col = "blue", cex =
0.65)
```

```
# add a legend, grid, and explanatory text
```

```
legend(x = "topleft", legend = c("plain", "peanut", "peanut butter"), col = c("red", "brown",  
"blue"), pch = c(19, 18, 17), bty = "n")
```

```
grid(col = "gray")
```

```
text(x = 16.5, y = 1, label = "data from University of Puget Sound Data Hoard", cex = 0.5)
```

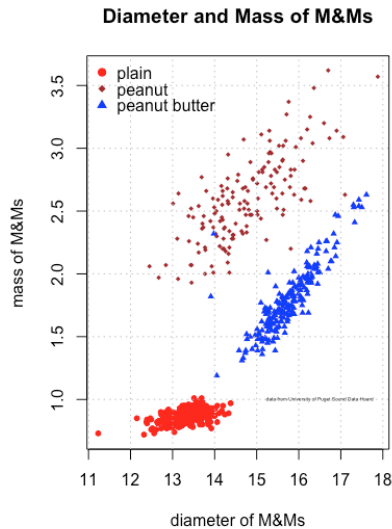


Figure 3.3.4: Example of a more informative scatterplot.

Our new plot shows that the individual M&Ms are reasonably well separated from each other in the space created by the variables diameter and mass, although a few M&Ms encroach into the space occupied by other types of M&Ms. We also see that the distribution of plain M&Ms is much more compact than for peanut and peanut butter M&Ms, which makes sense given the likely variability in the size of individual peanuts and the softer consistency of peanut butter.

3.4: Exercises

1. When copper metal and powdered sulfur are placed in a crucible and ignited, the product is a sulfide with an empirical formula of Cu_xS . The value of x is determined by weighing the Cu and the S before ignition and finding the mass of Cu_xS when the reaction is complete (any excess sulfur leaves as SO_2). The following table shows the Cu/S ratios from 62 such experiments (note that the values are organized from smallest-to-largest by rows). A copy of the data is [available](#) as a .csv file with data organized in a single column.

1.764	1.838	1.865	1.866	1.872	1.877
1.890	1.891	1.891	1.897	1.899	1.900
1.906	1.908	1.910	1.911	1.916	1.919
1.920	1.922	1.927	1.931	1.935	1.936
1.936	1.937	1.939	1.939	1.940	1.941
1.941	1.942	1.943	1.948	1.953	1.955
1.957	1.957	1.957	1.959	1.962	1.963
1.963	1.963	1.966	1.968	1.969	1.973
1.975	1.976	1.977	1.981	1.981	1.988
1.993	1.993	1.995	1.995	1.995	2.017
2.029	2.042				

(a) Construct a boxplot for this data and comment on your results.

(b) Construct a histogram and comment on your results.

2. Mizutani, Yabuki and Asai developed an electrochemical method for analyzing *l*-malate. As part of their study they analyzed a series of beverages using both their method and a standard spectrophotometric procedure based on a clinical kit purchased from Boehringer Scientific. The following table summarizes their results. All values are in ppm.

Sample	Electrode	Spectrophotometric
Apple Juice 1	34.0	33.4
Apple Juice 2	22.6	28.4
Apple Juice 3	29.7	29.5
Apple Juice 4	24.9	24.8
Grape Juice 1	17.8	18.3
Grape Juice 2	14.8	15.4
Mixed Fruit Juice 1	8.6	8.5
Mixed Fruit Juice 2	31.4	31.9
White Wine 1	10.8	11.5
White Wine 2	17.3	17.6
White Wine 3	15.7	15.4
White Wine 4	18.4	18.3

Construct a scatterplot of this data, placing values for the electrochemical method on the x-axis and values for the

spectrophotometric method on the y -axis. Use different symbols for the four types of beverages. The data in this problem are from Mizutani, F.; Yabuki, S.; Asai, M. *Anal. Chim. Acta* **1991**, 245,145–150. A copy of the data is [available](#) as a .csv file.

3. Ten laboratories were asked to determine an analyte's concentration of in three standard test samples. Following are the results, in $\mu\text{g/ml}$.

Laboratory	Sample 1	Sample 2	Sample 3
1	22.6	13.6	16.0
2	23.0	14.2	15.9
3	21.5	13.9	16.9
4	21.9	13.9	16.9
5	21.3	13.5	16.7
6	22.1	13.5	17.4
7	23.1	13.5	17.5
8	21.7	13.5	16.8
9	22.2	12.9	17.2
10	21.7	13.8	16.7

(a) Construct a single plot that contains separate stripcharts for each of the three samples.

(b) Construct a single plot that contains separate boxplots for each of the three samples.

The data in this problem are adapted from Steiner, E. H. "Planning and Analysis of Results of Collaborative Tests," in *Statistical Manual of the Association of Official Analytical Chemists*, Association of Official Analytical Chemists: Washington, D. C., 1975. A copy of the data is [available](#) as a .csv file.

4. Real-time quantitative PCR is an analytical method for determining trace amounts of DNA. During the analysis, each cycle doubles the amount of DNA. A probe species that fluoresces in the presence of DNA is added to the reaction mixture and the increase in fluorescence is monitored during the cycling. The cycle threshold, C_t , is the cycle when the fluorescence exceeds a threshold value. The data in the following table shows C_t values for three samples using real-time quantitative PCR. Each sample was analyzed 18 times.

Sample X		Sample Y		Sample Z	
24.24	25.14	24.41	28.06	22.97	23.43
23.97	24.57	27.21	27.77	22.93	23.66
24.44	24.49	27.02	28.74	22.95	28.79
24.79	24.68	26.81	28.35	23.12	23.77
23.92	24.45	26.64	28.80	23.59	23.98
24.53	24.48	27.63	27.99	23.37	23.56
24.95	24.30	28.42	28.21	24.17	22.80
24.76	24.60	25.16	28.00	23.48	23.29
25.18	24.57	28.53	28.21	23.80	23.86

Use two or more methods to analyze this data visually and write a brief report on your conclusions. The data in this problem is from Burns, M. J.; Nixon, G. J.; Foy, C. A.; Harris, N. *BMC Biotechnol.* **2005**, 5:31 ([open access publication](#)). A copy of the data is [available](#) as a .csv file.

5. The file [problem3_5.csv](#) contains data for 1061 United States pennies organized into three columns: the year the penny

was minted, the penny's mass (to four decimal places), and the location where the penny was minted (D = Denver and P = Philadelphia). Subset the data by year into three groups

- pennies minted before 1982
- pennies minted during 1982
- pennies minted after 1982

Plot separate histograms for the masses of the pennies in each group and comment on your results. The data in this problem was collected by Jordan Katz at Denison University and is available at the Analytical Sciences Digital Library's Active Learning website.

6. Use the element data you created in Exercise 1.3.1 to create several visualizations of your choosing. At least one of your visualizations should be a scatterplot and one should be a boxplot.

7. Use the data set you created in Exercise 2.3.2 on the daily average NOX concentrations and daily average temperatures recorded at a roadside monitoring station located on Marybone Road in Westminster. Use this data to prepare a scatterplot that shows the daily average NOX concentrations for January on the y-axis and the daily average temperature for January on the x-axis. Add to this plot, a second scatterplot that shows the daily average NOX concentrations for July on the y-axis and the daily average temperature for July on the x-axis. Comment on your results.

8. Use this [link](#) to access a case study on data analysis and complete the nine investigations included in Part II: Ways to Visualize Data.

4: SUMMARIZING DATA

In Chapter 3 we used data collected from 30 bags of M&Ms to explore ways to visualize data. Although a good visualization is a powerful tool for quickly examining our data qualitatively, inevitably we will need to be able to describe our data quantitatively as well. In this chapter we will consider ways to summarize our data using one or more statistical measures.

[4.1: Ways to Summarize Data](#)

[4.2: Using R to Summarize Data](#)

[4.3: Exercises](#)

4.1: Ways to Summarize Data

In Chapter 3 we used data collected from 30 bags of M&Ms to explore different ways to visualize data. In this chapter we consider several ways to summarize data using the net weights of the same bags of M&Ms. Here is the raw data.

Table 4.1.1: Net Weights for 30 Bags of M&Ms.

49.287	48.870	51.250	48.692	48.777	46.405
49.693	49.391	48.196	47.326	50.974	50.081
47.841	48.377	47.004	50.037	48.599	48.625
48.395	51.730	50.405	47.305	49.477	48.027
48.212	51.682	50.802	49.055	46.577	48.317

Without completing any calculations, what conclusions can we make by just looking at this data? Here are a few:

- All net weights are greater than 46 g and less than 52 g.
- As we see in Figure 4.1.1, a box-and-whisker plot (overlaid with a stripchart) and a histogram suggest that the distribution of the net weights is reasonably symmetric.
- The absence of any points beyond the whiskers of the box-and-whisker plot suggests that there are no unusually large or unusually small net weights.

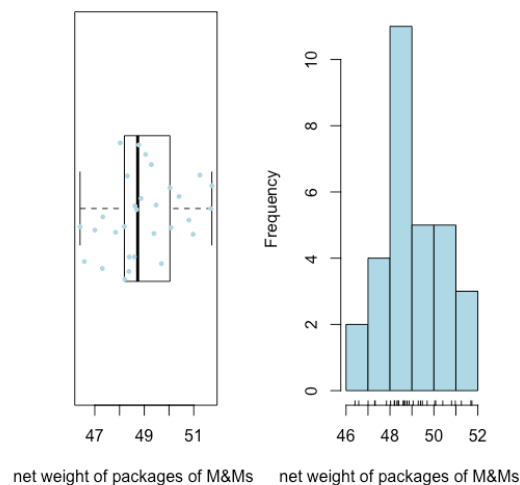


Figure 4.1.1: Two visualizations of the net weights of packages of M&Ms.

Both visualizations provide a good qualitative picture of the data, suggesting that the individual results are scattered around some central value with more results closer to that central value than at distance from it. Neither visualization, however, describes the data quantitatively. What we need is a convenient way to summarize the data by reporting where the data is centered and how varied the individual results are around that center.

Where is the Center?

There are two common ways to report the center of a data set: the mean and the median.

The mean, $\bar{Y}$, is the numerical average obtained by adding together the results for all n observations and dividing by the number of observations

$$\bar{Y} = \frac{\sum_{i=1}^n Y_i}{n} = \frac{49.287 + 48.870 + \dots + 48.317}{30} = 48.980 \text{ g}$$

The median, $\tilde{Y}$, is the middle value after we order our observations from smallest-to-largest, as we show here for our data.

Table 4.1.2: The data from Table 4.1.1 Sorted From Smallest-to-Largest in Value.

46.405	46.577	47.004	47.305	47.326	47.841
48.027	48.196	48.212	48.317	48.377	48.395
48.599	48.625	48.692	48.777	48.870	49.055
49.287	49.391	49.477	49.693	50.037	50.081
50.405	50.802	50.974	51.250	51.682	51.730

If we have an odd number of samples, then the median is simply the middle value, or

$$\tilde{Y} = Y_{\frac{n+1}{2}}$$

where n is the number of samples. If, as is the case here, n is even, then

$$\tilde{Y} = \frac{Y_{\frac{n}{2}} + Y_{\frac{n}{2}+1}}{2} = \frac{48.692 + 48.777}{2} = 48.734 \text{ g}$$

When our data has a symmetrical distribution, as we believe is the case here, then the mean and the median will have similar values.

What is the Variation of the Data About the Center?

There are five common measures of the variation of data about its center: the variance, the standard deviation, the range, the interquartile range, and the median average difference.

The variance, s^2 , is an average squared deviation of the individual observations relative to the mean

$$s^2 = \frac{\sum_{i=1}^n (Y_i - \bar{Y})^2}{n - 1} = \frac{(49.287 - 48.980)^2 + \dots + (48.317 - 48.980)^2}{30 - 1} = 2.052$$

and the standard deviation, s , is the square root of the variance, which gives it the same units as the mean.

$$s = \sqrt{\frac{\sum_{i=1}^n (Y_i - \bar{Y})^2}{n - 1}} = \sqrt{\frac{(49.287 - 48.980)^2 + \dots + (48.317 - 48.980)^2}{30 - 1}} = 1.432$$

The range, w , is the difference between the largest and the smallest value in our data set.

$$w = 51.730 \text{ g} - 46.405 \text{ g} = 5.325 \text{ g}$$

The interquartile range, IQR , is the difference between the median of the bottom 25% of observations and the median of the top 25% of observations; that is, it provides a measure of the range of values that spans the middle 50% of observations. There is no single, standard formula for calculating the IQR , and different algorithms yield slightly different results. We will adopt the algorithm described here:

1. Divide the sorted data set in half; if there is an odd number of values, then remove the median for the complete data set. For our data, the lower half is

Table 4.1.3: The Lower Half of the Data in Table 4.1.2.

46.405	46.577	47.004	47.305	47.326
47.841	48.027	48.196	48.212	48.317

48.377	48.395	48.599	48.625	48.692
--------	--------	--------	--------	--------

and the upper half is

Table 4.1.4: The Upper Half of the Data in Table 4.1.2.

48.777	48.870	49.055	49.287	49.391
49.477	49.693	50.037	50.081	50.405
50.802	50.974	51.250	51.682	51.730

2. Find F_L , the median for the lower half of the data, which for our data is 48.196 g.

3. Find F_U , the median for the upper half of the data, which for our data is 50.037 g.

4. The IQR is the difference between F_U and F_L .

$$F_U - F_L = 50.037 \text{ g} - 48.196 \text{ g} = 1.841 \text{ g}$$

The median absolute deviation, MAD , is the median of the absolute deviations of each observation from the median of all observations. To find the MAD for our set of 30 net weights, we first subtract the median from each sample in Table 4.1.1.

Table 4.1.5: The Results of Subtracting the Median From Each Value in Table 4.1.1.

0.5525	0.1355	2.5155	-0.0425	0.0425	-2.3295
0.9585	0.6565	-0.5385	-1.4085	2.2395	1.3465
-0.8935	-0.3575	-1.7305	1.3025	-0.1355	-0.1095
-0.3395	2.9955	1.6705	-1.4295	0.7425	-0.7075
-0.5225	2.9475	2.0675	0.3205	-2.1575	-0.4175

Next we take the absolute value of each difference and sort them from smallest-to-largest.

Table 4.1.6: The Data in Table 4.1.5 After Taking the Absolute Value.

0.0425	0.0425	0.1095	0.1355	0.1355	0.3205
0.3395	0.3575	0.4175	0.5225	0.5385	0.5525
0.6565	0.7075	0.7425	0.8935	0.9585	1.3025
1.3465	1.4085	1.4295	1.6705	1.7305	2.0675
2.1575	2.2395	2.3295	2.5155	2.9475	2.9955

Finally, we report the median for these sorted values as

$$\frac{0.7425 + 0.8935}{2} = 0.818$$

Robust vs. Non-Robust Measures of The Center and Variation About the Center

A good question to ask is why we might desire more than one way to report the center of our data and the variation in our data about the center. Suppose that the result for the last of our 30 samples was reported as 483.17 instead of 48.317. Whether this is an accidental shifting of the decimal point or a true result is not relevant to us here; what matters is its effect on what we report. Here is a summary of the effect of this one value on each of our ways of summarizing our data.

Table 4.1.7: Effect on Summary Statistics of Changing Last Value in Table 4.1.1 From 48.317 g to 483.17 g.

statistic	original data	new data
mean	48.980	63.475
median	48.734	48.824
variance	2.052	6285.938
standard deviation	1.433	79.280
range	5.325	436.765
<i>IQR</i>	1.841	1.885
<i>MAD</i>	0.818	0.926

Note that the mean, the variance, the standard deviation, and the range are very sensitive to the change in the last result, but the median, the *IQR*, and the *MAD* are not. The median, the *IQR*, and the *MAD* are considered robust statistics because they are less sensitive to an unusual result; the others are, of course, non-robust statistics. Both types of statistics have value to us, a point we will return to from time-to-time.

4.2: Using R to Summarize Data

One of R's strengths is its **Stats** package, which provides access to a rich body of tools for analyzing data. The package is part of R's base installation and is available whenever you use R without the need to use `library()` to make it available. Almost all of the statistical functions we will use in this textbook are included in the **Stats** package.

Bringing Your Data Into R

This section uses the M&M data in Table 1 of Chapter 3.1. You can download a copy of the data as a .csv spreadsheet using this [link](#). Before we can summarize our data, we need to make it available to R. The code below uses the `read.csv` function to read in the data from the file `MandM.csv()` as a data frame. The text "`MandM.csv`" assumes the file is located in your working directory.

```
mm_data = read.csv("MandM.csv")
```

Finding the Central Tendency of Data Using R

To report the mean of a data set we use the function `mean(x)` where `x` is the object that holds our data, typically a vector or a single column from a data frame. An important argument to this, and to many other functions, is how to handle missing or `NA` values. The default is to keep them, which leads to an error when we try to calculate the mean. This is a reasonable default as it requires us to make note of the missing values and to set `na.rm = TRUE` if we wish to remove them from the calculation. As our vector of data is not missing any values, we do not need to include `na.rm = TRUE` here, but we do so to illustrate its importance.

```
mean(mm_data$net_weight, na.rm = TRUE)
```

```
[1] 48.9803
```

To report the median of a data set we use the function `median(x)` where `x` is the object that holds our data, typically a vector or a single column from a data frame.

```
median(mm_data$net_weight, na.rm = TRUE)
```

```
[1] 48.7345
```

Finding the Spread of Data Using R

To report the variance of a data set we use the function `var(x)` where `x` is the object that holds our data, typically a vector or a single column from a data frame.

```
var(mm_data$net_weight, na.rm = TRUE)
```

```
[1] 2.052068
```

To report the standard deviation we use the function `sd(x)` where `x` is the object that holds our data, typically a vector or a single column from a data frame.

```
sd(mm_data$net_weight, na.rm = TRUE)
```

```
[1] 1.432504
```

To report the range we have to be creative as R's `range()` function does not directly report the range. Instead, it returns the minimum as its first value and the maximum as its second value, which we can extract using the bracket operator and then use to compute the range.

```
range(mm_data$net_weight, na.rm = TRUE)[2] - range(mm_data$net_weight, na.rm = TRUE)[1]  
[1] 5.325
```

Another approach for calculating the range is to use R's `max()` and `min()` functions.

```
max(mm_data$net_weight) - min(mm_data$net_weight)  
[1] 5.325
```

To report the interquartile range we use the function `IQR(x)` where `x` is the object that holds our data, typically a vector or a single column from a data frame. The function has nine different algorithms for calculating the *IQR*, identified using `type` as an argument. To obtain an *IQR* equivalent to that generated by R's `boxplot()` function, we use `type = 5` for an even number of values and `type = 7` for an odd number of values.

```
IQR(mm_data$net_weight, na.rm = TRUE, type = 5)  
[1] 1.841
```

To find the median absolute deviation we use the function `mad(x)` where `x` is the object that holds our data, typically a vector or a single column from a data frame. The function includes a scaling constant, the default value for which does not match our description for calculating the *MAD*; the argument `constant = 1` gives a result that is consistent with our description of the *MAD*.

```
mad(mm_data$net_weight, na.rm = TRUE, constant = 1)  
[1] 0.818
```

4.3: Exercises

1. The following masses were recorded for 12 different U.S. quarters (all values given in grams):

5.683	5.549	5.548	5.552
5.620	5.536	5.539	5.684
5.551	5.552	5.554	5.632

Report the mean, median, variance, standard deviation, range, IQR, and MAD for this data.

2. A determination of acetaminophen in 10 separate tablets of Excedrin Extra Strength Pain Reliever gives the following results (in mg). The data in this problem are from Simonian, M. H.; Dinh, S.; Fray, L. A. *Spectroscopy* **1993**, 8(6), 37–47.

224.3	240.4	246.3	239.4	253.1
261.7	229.4	255.5	235.5	249.7

Report the mean, median, variance, standard deviation, range, IQR, and MAD for this data.

3. Salem and Galan developed a new method to determine the amount of morphine hydrochloride in tablets. An analysis of tablets with different nominal dosages gave the following results (in mg/tablet). The data in this problem are from Salem, I. I.; Galan, A. C. *Anal. Chim. Acta* **1993**, 283, 334–337.

100-mg tablets	60-mg tablets	30-mg tablets	10-mg tablets
99.17	54.21	28.51	9.06
94.31	55.62	26.25	8.83
95.92	57.40	25.92	9.08
94.55	57.51	28.62	
93.83	52.59	24.93	

For each dosage, report the mean, median, variance, standard deviation, range, IQR, and MAD for this data.

4. Use the data set you create in Exercise 2.32 for the daily roadside monitoring of NOX concentrations and air temperatures along Marlybone Road. Report the mean, median, variance, standard deviation, range, IQR, and MAD for the NOX concentrations in January. Examine a boxplot of the data and note that two values are flagged. Remove these values and recalculate the mean, median, variance, standard deviation, range, IQR, and MAD for this data. Compare these results to those calculated using all of the data and comment on your results.

5. Use this [link](#) to access a case study on data analysis and complete the three investigations included in Part III: Ways to Summarize Data.

5: THE DISTRIBUTION OF DATA

When we measure something, such as the percentage of yellow M&Ms in a bag of M&Ms, we expect two things:

- that there is an underlying “true” value that our measurements should approximate, and
- that the results of individual measurements will show some variation about that “true” value

Visualizations of data—such as dot plots, stripcharts, boxplot-and-whisker plots, bar plots, histograms, and scatterplots—often suggest there is an underlying structure to our data. For example, we saw in Chapter 3 that the distribution of yellow M&Ms in bags of M&Ms is more or less symmetrical around its median, while the distribution of orange M&Ms was skewed toward higher values. This underlying structure, or distribution, of our data as it effects how we choose to analyze our data. In this chapter we will take a closer look at several ways in which data are distributed.

5.1: Terminology

5.2: Theoretical Models for the Distribution of Data

5.3: The Central Limit Theorem

5.4: Modeling Distributions Using R

5.5: Exercises

5.1: Terminology

Before we consider different types of distributions, let's define some key terms. You may wish, as well, to review the discussion of different types of data in Chapter 2.

Populations and Samples

A population includes every possible measurement we could make on a system, while a sample is the subset of a population on which we actually make measurements. These definitions are fluid. A single bag of M&Ms is a population if we are interested only in that specific bag, but it is but one sample from a box that contains a gross (144) of individual bags. That box, itself, can be a population, or it can be one sample from a much larger production lot. And so on.

Discrete Distributions and Continuous Distributions

In a discrete distribution the possible results take on a limited set of specific values that are independent of how we make our measurements. When we determine the number of yellow M&Ms in a bag, the results are limited to integer values. We may find 13 yellow M&Ms or 24 yellow M&Ms, but we cannot obtain a result of 15.43 yellow M&Ms.

For a continuous distribution the result of a measurement can take on any possible value between a lower limit and an upper limit, even though our measuring device has a limited precision; thus, when we weigh a bag of M&Ms on a three-digit balance and obtain a result of 49.287 g we know that its true mass is greater than 49.2865... g and less than 49.2875... g.

5.2: Theoretical Models for the Distribution of Data

There are four important types of distributions that we will consider in this chapter: the uniform distribution, the binomial distribution, the Poisson distribution, and the normal, or Gaussian, distribution. In Chapter 3 and Chapter 4 we used the analysis of bags of M&Ms to explore ways to visualize data and to summarize data. Here we will use the same data set to explore the distribution of data.

Uniform Distribution

In a uniform distribution, all outcomes are equally probable. Suppose the population of M&Ms has a uniform distribution. If this is the case, then, with six colors, we expect each color to appear with a probability of $1/6$ or 16.7%. Figure 5.2.1 shows a comparison of the theoretical results if we draw 1699 M&Ms—the total number of M&Ms in our sample of 30 bags—from a population with a uniform distribution (on the left) to the actual distribution of the 1699 M&Ms in our sample (on the right). It seems unlikely that the population of M&Ms has a uniform distribution of colors!

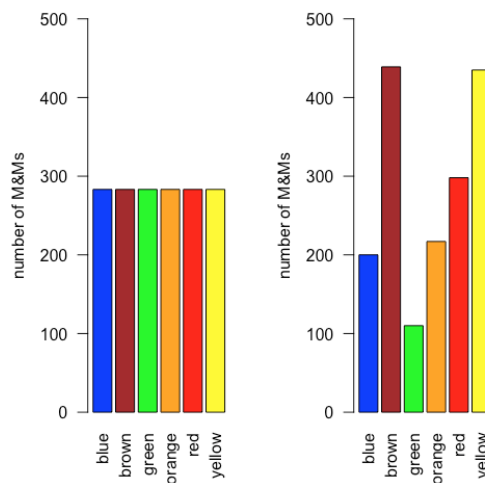


Figure 5.2.1: Comparison of (on the left) a uniform distribution of 1699 M&Ms with (on the right) the actual distribution from the sample in Table 3.1.1.

Binomial Distribution

A binomial distribution shows the probability of obtaining a particular result in a fixed number of trials, where the odds of that result happening in a single trial are known. Mathematically, a binomial distribution is defined by the equation

$$P(X, N) = \frac{N!}{X!(N-X)!} \times p^X \times (1-p)^{N-X}$$

where $P(X, N)$ is the probability that the event happens X times in N trials, and where p is the probability that the event happens in a single trial. The binomial distribution has a theoretical mean, μ , and a theoretical variance, σ^2 , of

$$\mu = Np \quad \sigma^2 = Np(1-p)$$

Figure 5.2.2 compares the expected binomial distribution for drawing 0, 1, 2, 3, 4, or 5 yellow M&Ms in the first five M&Ms—assuming that the probability of drawing a yellow M&M is $435/1699$, the ratio of the number of yellow M&Ms and the total number of M&Ms—to the actual distribution of results. The similarity between the theoretical and the actual results seems evident; in Chapter 6 we will consider ways to test this claim.

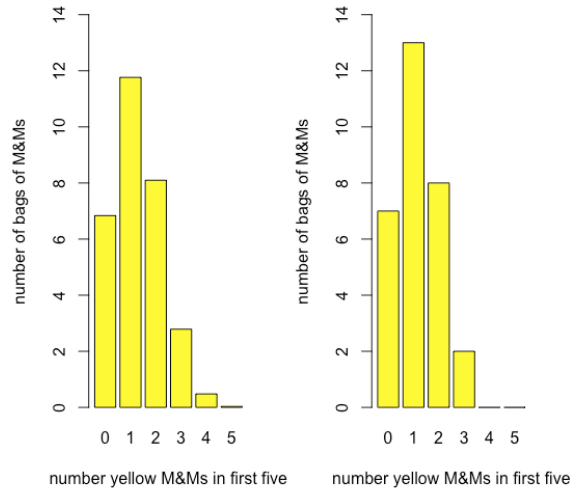


Figure 5.2.2: Comparison of (on the left) the theoretical binomial distribution of yellow M&Ms in the first five selected from a bag of M&Ms and (on the right) the actual distribution of M&Ms.

Poisson Distribution

The binomial distribution is useful if we wish to model the probability of finding a fixed number of yellow M&Ms in a sample of M&Ms of fixed size—such as the first five M&Ms that we draw from a bag—but not the probability of finding a fixed number of yellow M&Ms in a single bag because there is some variability in the total number of M&Ms per bag.

A Poisson distribution gives the probability that a given number of events will occur in a fixed interval in time or space if the event has a known average rate and if each new event is independent of the preceding event. Mathematically a Poisson distribution is defined by the equation

$$P(X, \lambda) = \frac{e^{-\lambda} \lambda^X}{X!}$$

where $P(X, \lambda)$ is the probability that an event happens X times given the event's average rate, λ . The Poisson distribution has a theoretical mean, μ , and a theoretical variance, σ^2 , that are each equal to λ .

The bar plot in Figure 5.2.3 shows the actual distribution of green M&Ms in 35 small bags of M&Ms (as reported by M. A. Xu-Friedman “Illustrating concepts of quantal analysis with an intuitive classroom model,” *Adv. Physiol. Educ.* **2013**, 37, 112–116). Superimposed on the bar plot is the theoretical Poisson distribution based on their reported average rate of 3.4 green M&Ms per bag. The similarity between the theoretical and the actual results seems evident; in Chapter 6 we will consider ways to test this claim.

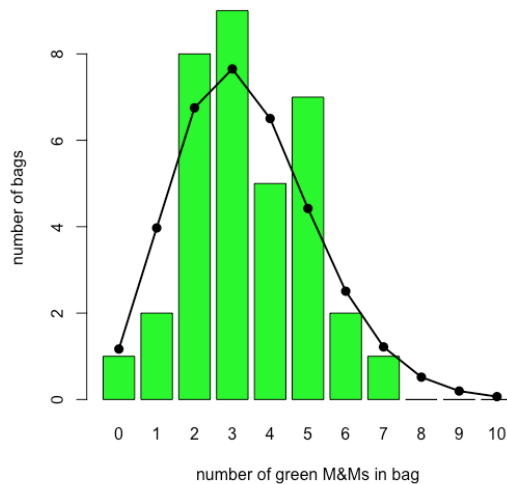


Figure 5.2.3: Comparison of a Poisson distribution for green M&Ms (dots and line) to experimental results (bars). The data are from M. A. Xu-Friedman, "Illustrating concepts of quantal analysis with an intuitive classroom model," Adv. Physiol. Educ. **2013**, 37, 112-116.

Normal Distribution

A uniform distribution, a binomial distribution, and a Poisson distribution predict the probability of a discrete event, such as the probability of finding exactly two green M&Ms in the next bag of M&Ms that we open. Not all of the data we collect is discrete. The net weights of bags of M&Ms is an example of continuous data as the mass of an individual bag is not restricted to a discrete set of allowed values. In many cases we can model continuous data using a normal (or Gaussian) distribution, which gives the probability of obtaining a particular outcome, $P(x)$, from a population with a known mean, μ , and a known variance, σ^2 . Mathematically a normal distribution is defined by the equation

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/(2\sigma^2)}$$

Figure 5.2.4 shows the expected normal distribution for the net weights of our sample of 30 bags of M&Ms if we assume that their mean, $\bar{X}$, of 48.98 g and standard deviation, s , of 1.433 g are good predictors of the population's mean, μ , and standard deviation, σ . Given the small sample of 30 bags, the agreement between the model and the data seems reasonable.

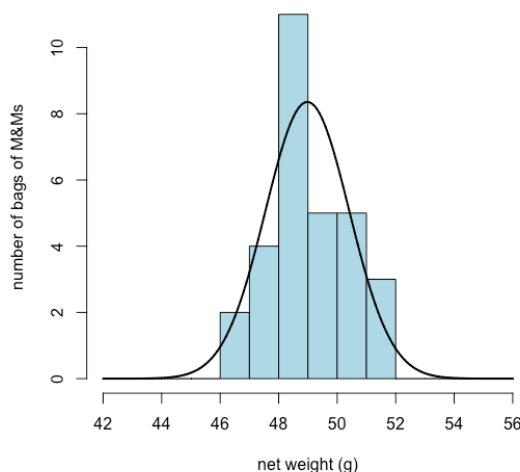


Figure 5.2.4: Comparison of a normal distribution for the net weights of M&Ms (line) to the experimental results (bars).

This page titled [5.2: Theoretical Models for the Distribution of Data](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [David Harvey](#).

5.3: The Central Limit Theorem

Suppose we have a population for which one of its properties has a uniform distribution where every result between 0 and 1 is equally probable. If we analyze 10,000 samples we should not be surprised to find that the distribution of these 10,000 results looks uniform, as shown by the histogram on the left side of Figure 5.3.1. If we collect 1000 pooled samples—each of which consists of 10 individual samples for a total of 10,000 individual samples—and report the average results for these 1000 pooled samples, we see something interesting as their distribution, as shown by the histogram on the right, looks remarkably like a normal distribution. When we draw single samples from a uniform distribution, each possible outcome is equally likely, which is why we see the distribution on the left. When we draw a pooled sample that consists of 10 individual samples, however, the average values are more likely to be near the middle of the distribution's range, as we see on the right, because the pooled sample likely includes values drawn from both the lower half and the upper half of the uniform distribution.

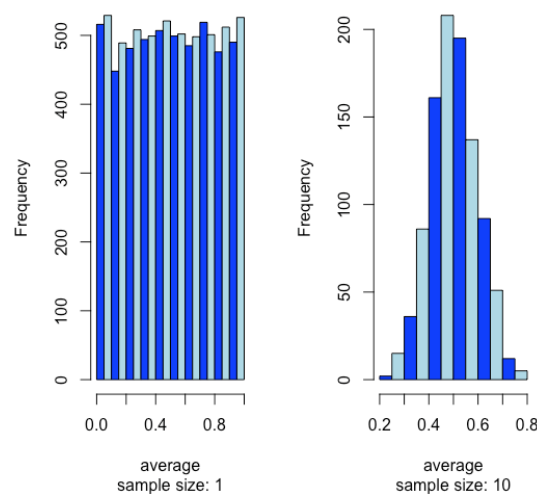


Figure 5.3.1: Distribution of results when analyzing samples of size $n = 1$ (left) and samples of size $n = 10$ (right) drawn from a uniform distribution.

This tendency for a normal distribution to emerge when we pool samples is known as the central limit theorem. As shown in Figure 5.3.2, we see a similar effect with populations that follow a binomial distribution or a Poisson distribution.

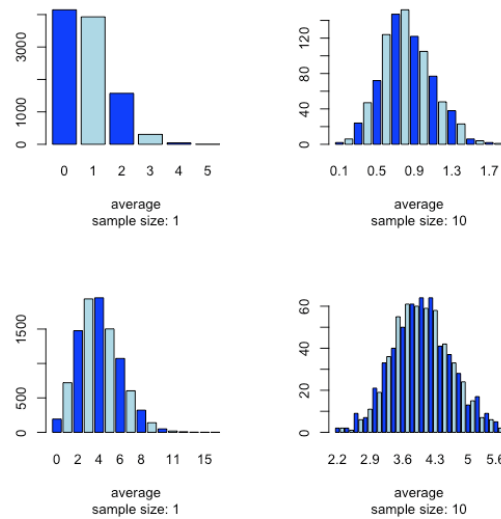


Figure 5.3.2: Distribution of results when analyzing samples of size $n = 1$ (left) and samples of size $n = 10$ (right) drawn from a binomial distribution with $p = 0.167$ (top) and a Poisson distribution with $\lambda = 4$ (bottom).

You might reasonably ask whether the central limit theorem is important as it is unlikely that we will complete 1000 analyses, each of which is the average of 10 individual trials. This is deceiving. When we acquire a sample of soil, for example, it consists of many individual particles each of which is an individual sample of the soil. Our analysis of this sample, therefore, is the mean for a large number of individual soil particles. Because of this, the central limit theorem is relevant.

5.4: Modeling Distributions Using R

The base installation of R includes a variety of functions for working with uniform distributions, binomial distributions, Poisson distributions, and normal distributions. These functions come in four forms that take the general form `xdist` where `dist` is the type of distribution (`unif` for a uniform distribution, `binom` for a binomial distribution, `pois` for a Poisson distribution, and `norm` for a normal distribution), and where `x` defines the information we extract from the distribution. For example, the function `dunif()` returns the probability of obtaining a specific value drawn from a uniform distribution, the function `pbinom()` returns the probability of obtaining a result less than a defined value from a binomial distribution, the function `qpois()` returns the upper boundary that includes a defined percentage of results from a Poisson distribution, and the function `rnorm()` returns results drawn at random from a normal distribution.

Modeling a Uniform Distribution Using R

When you purchase a Class A 10.00-mL volumetric pipet it comes with a tolerance of ± 0.02 mL, which is the manufacturer's way of saying that the pipet's true volume is no less than 9.98 mL and no greater than 10.02 mL. Suppose a manufacturer produces 10,000 pipets, how many might we expect to have a volume between 9.990 mL and 9.992 mL? A uniform distribution is the choice when the manufacturer provides a tolerance range without specifying a level of confidence and when there is no reason to believe that results near the center of the range are more likely than results at the ends of the range.

To simulate a uniform distribution we use R's `runif(n, min, max)` function, which returns `n` random values drawn from a uniform distribution defined by its minimum (`min`) and its maximum (`max`) limits. The result is shown in Figure 5.4.1, where the dots, added using the `points()` function, show the theoretical uniform distribution at the midpoint of each of the histogram's bins.

```
# create vector of volumes for 10000 pipets drawn at random from uniform distribution
pipet = runif(10000, 9.98, 10.02)

# create histogram using 20 bins of size 0.002 mL
pipet_hist = hist(pipet, breaks = seq(9.98, 10.02, 0.002), col = c("blue", "lightblue"),
  ylab = "number of pipets", xlab = "volume of pipet (mL)", main = NULL)

# overlay points showing expected values for uniform distribution
points(pipet_hist$mids, rep(10000/20, 20), pch = 19)
```

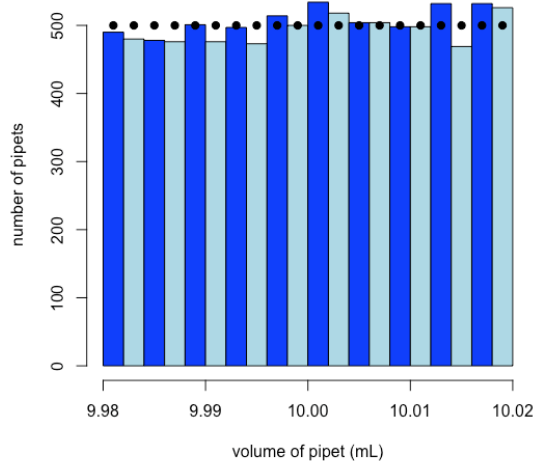


Figure 5.4.1: Uniform distribution of volumes for 10000 10-mL volumetric pipets. The individual bars show the simulated results and the individual dots show the expected results.

Saving the histogram to the object `pipet_hist` allows us to retrieve the number of pipets in each of the histogram's intervals; thus, there are 476 pipets with volumes between 9.990 mL and 9.992 mL, which is the sixth bar from the left edge of Figure 5.4.1.

```
pipet_hist$counts[6]
```

```
[1] 476
```

Modeling a Binomial Distribution Using R

Carbon has two stable, non-radioactive isotopes, ^{12}C and ^{13}C , with relative isotopic abundances of, respectively, 98.89% and 1.11%. Suppose we are working with cholesterol, $\text{C}_{27}\text{H}_{44}\text{O}$, which has 27 atoms of carbon. We can use the binomial distribution to model the expected distribution for the number of atoms ^{13}C in 1000 cholesterol molecules.

To simulate the distribution we use R's `rbinom(n, size, prob)` function, which returns `n` random values drawn from a binomial distribution defined by the `size` of our sample, which is the number of possible carbon atoms, and the isotopic abundance of ^{13}C , which is its `prob` or probability. The result is shown in Figure 5.4.2, where the dots, added using the `points()` function, show the theoretical binomial distribution. These theoretical values are calculated using the `dbinom()` function. The bar plot is assigned to the object `chol_bar` to provide access to the values of `x` when plotting the points.

```
# create vector with 1000 values drawn at random from binomial distribution
```

```
cholesterol = rbinom(1000, 27, 0.0111)
```

```
# create bar plot of results; table(cholesterol) determines the number of cholesterol
```

```
# molecules with 0, 1, 2... atoms of carbon-13; dividing by 1000 gives probability
```

```
chol_bar = barplot(table(cholesterol)/1000, col = "lightblue", ylim = c(0,1), xlab =  
"number of atoms of carbon-13", ylab = "probability")
```

```
# theoretical results for binomial distribution of carbon-13 in cholesterol
```

```
chol_binom = dbinom(seq(0,27,1), 27, 0.0111)
```

```
# overlay theoretical results for binomial distribution
```

```
points(x = chol_bar, y = chol_binom[1:length(chol_bar)], cex = 1.25, pch = 19)
```

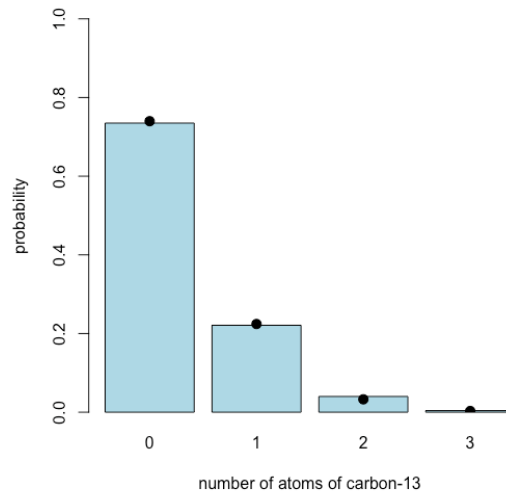


Figure 5.4.2: Distribution of results for carbon-13 atoms in cholesterol. The individual bars show the simulated results and the individual dots show the expected results.

Modeling a Poisson Distribution Using R

One measure of the quality of water in lakes used for recreational purposes is a fecal coliform test. In a typical test a sample of water is passed through a membrane filter, which is then placed on a medium to encourage growth of the bacteria and incubated for 24 hours at 44.5°C. The number of colonies of bacteria is reported. Suppose a lake has a natural background level of 5 colonies per 50 mL of water tested and must be closed for swimming if it exceeds 10 colonies per 50 mL of water tested. We can use a Poisson distribution to determine, over the course of a year of daily testing, the probability that a test will exceed this limit even though the lake's true fecal coliform count remains at its natural background level.

To simulate the distribution we use R's `rpois(n, lambda)` function, which returns `n` random values drawn from a Poisson distribution defined by `lambda` which is its average incidence. Because we are interested in modeling out a year, `n` is set to 365 days. The result is shown in Figure 5.4.3, where the dots, added using the `points()` function, shows the theoretical Poisson distribution. These theoretical values are calculated using the `dpois()` function. The bar plot is assigned to the object `coliform_bar` to provide access to the values of `x` when plotting the points.

```
# create vector of results drawn at random from Poisson distribution
coliforms = rpois(365,5)

# create table of simulated results
coliform_table = table(coliforms)

# create bar plot; ylim ensures there is some space above the plot's highest bar
coliform_bar = barplot(coliform_table, ylim = c(0, 1.2 * max(coliform_table)), col =
"lightblue")

# theoretical results for Poisson distribution
d_coliforms = dpois(seq(0,length(coliform_bar) - 1), 5) * 365

# overlay theoretical results for Poisson distribution
points(coliform_bar, d_coliforms, pch = 19)
```

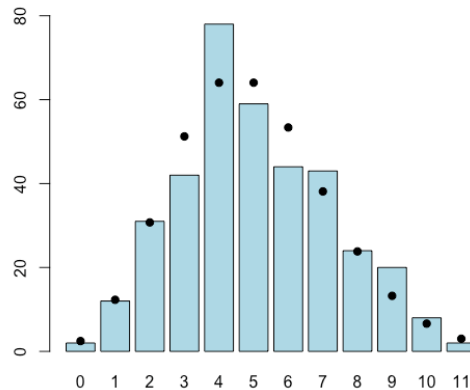


Figure 5.4.3: Distribution of results for fecal coliform test over course of a year. The individual bars show the simulated results and the individual dots show the expected results.

To find the number of times our simulated results exceed the limit of 10 coliforms colonies per 50 mL we use R's `which()` function to identify within `coliforms` the values that are greater than 10

```
coliforms[which(coliforms > 10)]
```

finding that this happen 2 times over the course of a year.

The theoretical probability that a single test will exceed the limit of 10 colonies per 50 mL of water, we use R's `ppois(q, lambda)` function, where `q` is the value we wish to test, which returns the cumulative probability of obtaining a result less than or equal to `q` on any day; over the course of 365 days

```
(1 - ppois(10,5))*365
```

```
[1] 4.998773
```

we expect that on 5 days the fecal coliform count will exceed the limit of 10.

Modeling a Normal Distribution Using R

If we place copper metal and an excess of powdered sulfur in a crucible and ignite it, copper sulfide forms with an empirical formula of Cu_xS . The value of x is determined by weighing the Cu and the S before ignition and finding the mass of Cu_xS when the reaction is complete (any excess sulfur leaves as the gas SO_2). The following are the Cu/S ratios from 62 such experiments, of which just 3 are greater than 2. Because of the central limit theorem, we can use a normal distribution to model the data.

Table 5.4.1: Experimental Cu/S Ratios When Igniting Cu(s) and S(s).

1.764	1.838	1.890	1.891	1.906	1.908
1.920	1.922	1.936	1.937	1.941	1.942
1.957	1.957	1.963	1.963	1.975	1.976
1.993	1.993	2.029	2.042	1.866	1.872
1.891	1.897	1.899	1.910	1.911	1.916
1.927	1.931	1.935	1.939	1.939	1.940
1.943	1.948	1.953	1.957	1.959	1.962

1.966	1.968	1.969	1.977	1.981	1.981
1.995	1.995	1.865	1.995	1.877	1.900
1.919	1.936	1.941	1.955	1.963	1.973
1.988	2.017				

Figure 5.4.4 shows the distribution of the experimental results as a histogram overlaid with the theoretical normal distribution calculated assuming that μ is equal to the mean of the 62 samples and that σ is equal to the standard deviation of the 62 samples. Both the experimental data and theoretical normal distribution suggest that most values of x are between 1.85 and 2.03.

```
# enter the data into a vector with the name cuxs
```

```
cuxs = c(1.764, 1.920, 1.957, 1.993, 1.891, 1.927, 1.943, 1.966, 1.995, 1.919, 1.988, 1.838,
1.922, 1.957, 1.993, 1.897, 1.931, 1.948, 1.968, 1.995, 1.936, 2.017, 1.890, 1.936, 1.963,
2.029, 1.899, 1.935, 1.953, 1.969, 1.865, 1.941, 1.891, 1.937, 1.963, 2.042, 1.910, 1.939,
1.957, 1.977, 1.995, 1.955, 1.906, 1.941, 1.975, 1.866, 1.911, 1.939, 1.959, 1.981, 1.877,
1.963, 1.908, 1.942, 1.976, 1.872, 1.916, 1.940, 1.962, 1.981, 1.900, 1.973)
```

```
# sequence of ratios over which to display experimental results and theoretical distribution
```

```
x = seq(1.7,2.2,0.02)
```

```
# create histogram for experimental results
```

```
cuxs_hist = hist(cuxs, breaks = x, col = c("blue", "lightblue"), xlab = "value for x", ylab =
"frequency", main = NULL)
```

```
# calculate theoretical results for normal distribution using the mean and the standard deviation
```

```
# for the 62 samples as predictors for mu and sigma
```

```
cuxs_theo = dnorm(cuxs_hist$mids, mean = mean(cuxs), sd = sd(cuxs))
```

```
# overlay results for theoretical normal distribution
```

```
points(cuxs_hist$mids, cuxs_theo, pch = 19)
```

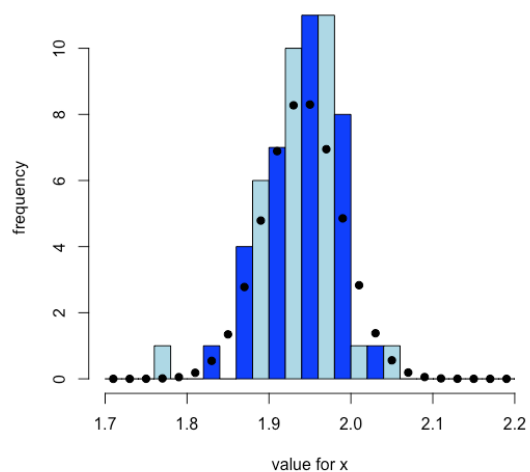


Figure 5.4.4: Distribution of results for ratio of Cu-to-S in preparations of copper sulfide. The individual bars show the simulated results and the individual dots show the expected results.

5.5: Exercises

Behavioral and ecological factors influence dispersion. Uniform patterns of dispersion are generally a result of interactions between individuals like competition and territoriality.

1. In ecology a uniform distribution of an organism may result when the organism exhibits territorial behavior that keeps most organisms. In one study, a portion of a field was divided into a 20×20 grid and a count made of the number of organisms in each unit of the grid giving the results seen below.

number of organisms in plot	frequency
2	58
3	51
4	60
5	64
6	54
7	52
8	61

Create a plot similar to that in 5.4.1 and comment on your results.

2. Chlorine has two isotopes, ^{35}Cl (75.8% abundance) and ^{37}Cl (24.2% abundance). Create a plot similar to that in Figure 5.4.2 for the molecule PCB 77, a chlorinated compound with the formula $\text{C}_{12}\text{H}_6\text{Cl}_4$ and comment on your results.

3. A radioactive decay process has a background level of 3 emissions per minute and follows a Poisson distribution. The number of emissions per minute was monitored for one hour giving the following results

emissions per minute	frequency of event
0	3
1	9
2	13
3	16
4	9
5	5
6	3
7	1
8	1
9	0
10	0

Use this data to create a plot similar to that in Figure 5.4.3 and comment on your results

4. Using the penny data from Exercise 3.4.5, create a plot similar to that in Figure 5.4.4 using all pennies minted after 1982 and comment on your results.

5. Use this [link](#) to access a case study on data analysis and complete the first four investigations included in Part IV: Ways to

Model Data.

This page titled [5.5: Exercises](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [David Harvey](#).

6: UNCERTAINTY OF DATA

In Chapter 5 we examined four ways in which the individual samples we collect and analyze are distributed about a central value: a uniform distribution, a binomial distribution, a Poisson distribution, and a normal distribution. We also learned that regardless of how individual samples are distributed, the distribution of averages for multiple samples often follows a normal distribution. This tendency for a normal distribution to emerge when we report averages for multiple samples is known as the central limit theorem. In this chapter we look more closely at the normal distribution—examining some of its properties—and consider how we can use these properties to say something more meaningful about our data than simply reporting a mean and a standard deviation.

6.1: Properties of a Normal Distribution

6.2: Confidence Intervals

6.3: Using R to Model Properties of a Normal Distribution

6.4: Using R to Find Confidence Intervals

6.5: Exercises

6.1: Properties of a Normal Distribution

Mathematically a normal distribution is defined by the equation

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/(2\sigma^2)}$$

where $P(x)$ is the probability of obtaining a result, x , from a population with a known mean, μ , and a known standard deviation, σ . Figure 6.1.1 shows the normal distribution curves for $\mu = 0$ with standard deviations of 5, 10, and 20.

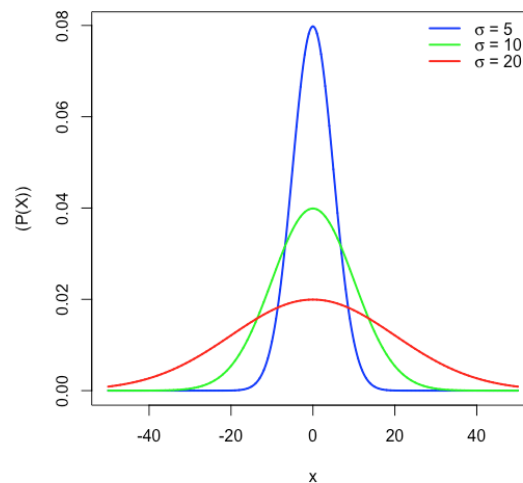


Figure 6.1.1: Three examples of normal distribution curves. Although the height and width are affected by σ , the area under each curve is the same.

Because the equation for a normal distribution depends solely on the population's mean, μ , and its standard deviation, σ , the probability that a sample drawn from a population has a value between any two arbitrary limits is the same for all populations. For example, Figure 6.1.2 shows that 68.26% of all samples drawn from a normally distributed population have values within the range $\mu \pm 1\sigma$, and only 0.14% have values greater than $\mu + 3\sigma$.

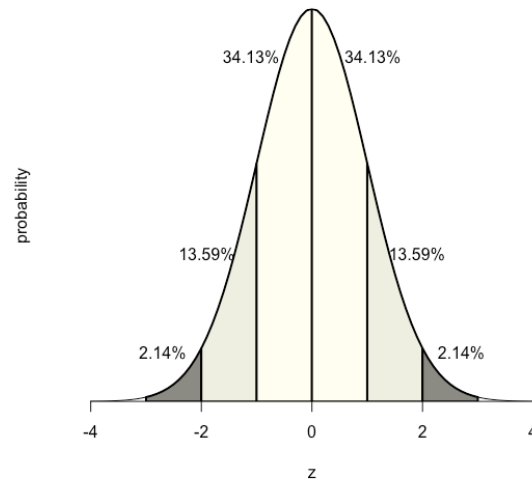


Figure 6.1.2: Normal distribution curve for $\mu = 0$ and $\sigma = 1$ showing area under the curve for various values of z in $\mu \pm z\sigma$.

This feature of a normal distribution—that the area under the curve is the same for all values of σ —allows us to create a probability table (see Appendix 1) based on the relative deviation, z , between a limit, x , and the mean, μ .

$$z = \frac{x - \mu}{\sigma}$$

The value of z gives the area under the curve between that limit and the distribution's closest tail, as shown in Figure 6.1.3.

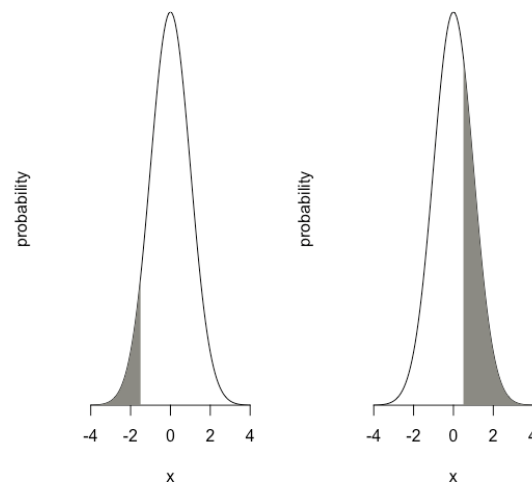


Figure 6.1.3: Normal distribution curve for $\mu = 0$ and $\sigma = 1$ showing (on the left) the area under the curve for $z = -1.5$ and (on the right) for $z = +0.5$.

Example 6.1.1

Suppose we know that μ is 5.5833 ppb Pb and that σ is 0.0558 ppb Pb for a particular standard reference material (SRM). What is the probability that we will obtain a result that is greater than 5.650 ppb if we analyze a single, random sample drawn from the SRM?

Solution

Figure 6.1.4 shows the normal distribution curve given values of 5.5833 ppb Pb for μ and of 0.0558 ppb Pb σ . The shaded area in the figures is the probability of obtaining a sample with a concentration of Pb greater than 5.650 ppm. To determine the probability, we first calculate z

$$z = \frac{x - \mu}{\sigma} = \frac{5.650 - 5.5833}{0.0558} = 1.195$$

Next, we look up the probability in Appendix 1 for this value of z , which is the average of 0.1170 (for $z = 1.19$) and 0.1151 (for $z = 1.20$), or a probability of 0.1160; thus, we expect that 11.60% of samples will provide a result greater than 5.650 ppb Pb.

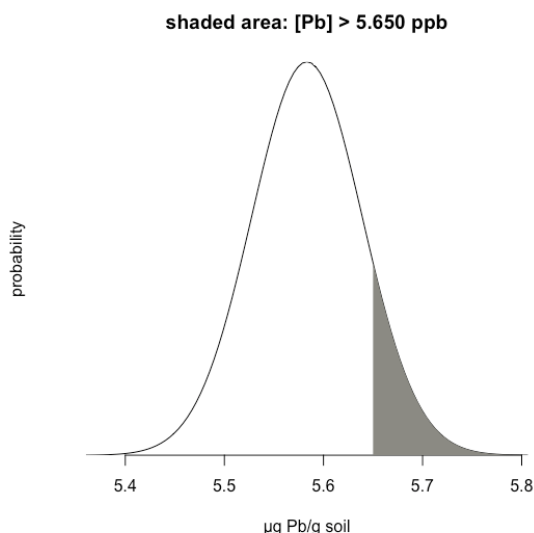


Figure 6.1.4: Normal distribution curve for the amount of lead in a standard reference with $\mu = 5.5833$ ppb and $\sigma = 0.0558$ ppb. The shaded area shows those results for which the concentration of lead exceeds 5.650 ppb.

Example 6.1.2

Example 6.1.1 considers a single limit—the probability that a result exceeds a single value. But what if we want to determine the probability that a sample has between 5.580 g Pb and 5.625 g Pb?

Solution

In this case we are interested in the shaded area shown in Figure 6.1.5. First, we calculate z for the upper limit

$$z = \frac{5.625 - 5.5833}{0.0558} = 0.747$$

and then we calculate z for the lower limit

$$z = \frac{5.580 - 5.5833}{0.0558} = -0.059$$

Then, we look up the probability in Appendix 1 that a result will exceed our upper limit of 5.625, which is 0.2275, or 22.75%, and the probability that a result will be less than our lower limit of 5.580, which is 0.4765, or 47.65%. The total unshaded area is 71.4% of the total area, so the shaded area corresponds to a probability of

$$100.00 - 22.75 - 47.65 = 100.00 - 71.40 = 29.6\%$$

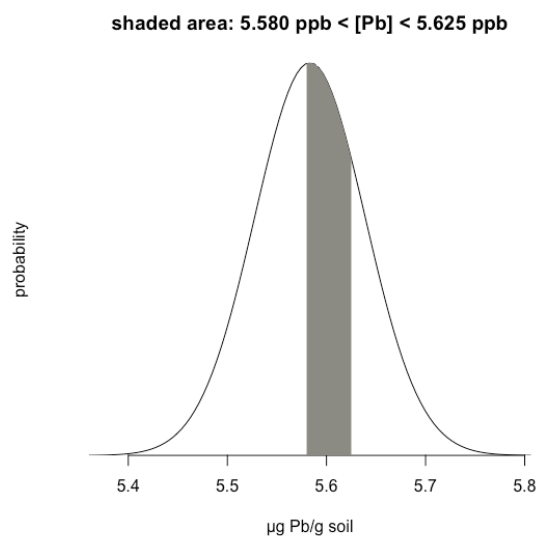


Figure 6.1.5: Normal distribution curve for the amount of lead in a standard reference with $\mu = 5.5833 \text{ ppb}$ and $\sigma = 0.0558 \text{ ppb}$. The shaded area shows those results for which the concentration of lead is more than 5.580 ppb and less than 5.625 ppb.

6.2: Confidence Intervals

In the previous section, we learned how to predict the probability of obtaining a particular outcome if our data are normally distributed with a known μ and a known σ . For example, we estimated that 11.60% of samples drawn at random from a standard reference material will have a concentration of Pb greater than 5.650 ppb given a μ of 5.5833 ppb and a σ of 0.0558 ppb. In essence, we determined how many standard deviations 5.650 is from μ and used this to define the probability given the standard area under a normal distribution curve.

We can look at this in a different way by asking the following question: If we collect a single sample at random from a population with a known μ and a known σ , within what range of values might we reasonably expect to find the sample's result 95% of the time? Rearranging the equation

$$z = \frac{x - \mu}{\sigma}$$

and solving for x gives

$$x = \mu \pm z\sigma = 5.5833 \pm (1.96)(0.0558) = 5.5833 \pm 0.1094$$

where a z of 1.96 corresponds to 95% of the area under the curve; we call this a 95% confidence interval for a single sample.

It generally is a poor idea to draw a conclusion from the result of a single experiment; instead, we usually collect several samples and ask the question this way: If we collect n random samples from a population with a known μ and a known σ , within what range of values might we reasonably expect to find the mean of these samples 95% of the time?

We might reasonably expect that the standard deviation for the mean of several samples is smaller than the standard deviation for a set of individual samples; indeed it is and it is given as

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$$

where $\frac{\sigma}{\sqrt{n}}$ is called the standard error of the mean. For example, if we collect three samples from the standard reference material described above, then we expect that the mean for these three samples will fall within a range

$$\bar{x} = \mu \pm z\sigma_{\bar{x}} = \mu \pm \frac{z\sigma}{\sqrt{n}} = 5.5833 \pm \frac{(1.96)(0.0558)}{\sqrt{3}} = 5.5833 \pm 0.0631$$

that is ± 0.0631 ppb around μ , a range that is smaller than that of ± 0.1094 ppb when we analyze individual samples. Note that the relative value to us of increasing the sample's size diminishes as n increases because of the square root term, as shown in Figure 6.2.1.

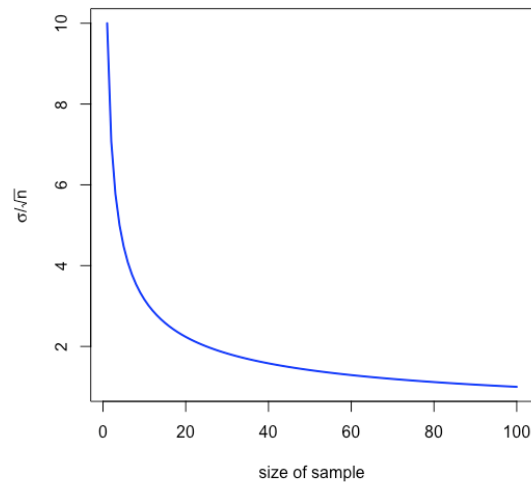


Figure 6.2.1: Plot showing how the standard error of the mean varies with the size of the sample. The value for σ is 10.

Our treatment thus far assumes we know μ and σ for the parent population, but we rarely know these values; instead, we examine samples drawn from the parent population and ask the following question: Given the sample's mean, $\bar{x}$, and its standard deviation, s , what is our best estimate of the population's mean, μ , and its standard deviation, σ .

To make this estimate, we replace the population's standard deviation, σ , with the standard deviation, s , for our samples, replace the population's mean, μ , with the mean, $\bar{x}$, for our samples, replace z with t , where the value of t depends on the number of samples, n

$$\bar{x} = \mu \pm \frac{ts}{\sqrt{n}}$$

and then rearrange the equation to solve for μ .

$$\mu = \bar{x} \pm \frac{ts}{\sqrt{n}}$$

We call this a confidence interval. Values for t are available in tables (see Appendix 2) and depend on the probability level, α , where $(1 - \alpha) \times 100$ is the confidence level, and the degrees of freedom, $n - 1$; note that for any probability level, $t \rightarrow z$ as $n \rightarrow \infty$.

We need to give special attention to what this confidence interval means and to what it does not mean:

- It **does not** mean that there is a 95% probability that the population's mean is in the range $\mu = \bar{x} \pm ts$ because our measurements may be biased or the normal distribution may be inappropriate for our system.
- It **does provide** our best estimate of the population's mean, μ given our analysis of n samples drawn at random from the parent population; a different sample, however, will give a different confidence interval and, therefore, a different estimate for μ .

6.3: Using R to Model Properties of a Normal Distribution

Given a mean and a standard deviation, we can use R's `dnorm()` function to plot the corresponding normal distribution

```
dnorm(x, mean, sd)
```

where `mean` is the value for μ , `sd` is the value for σ , and `x` is a vector of values that spans the range of x-axis values we want to plot.

```
# define the mean and the standard deviation
```

```
mu = 12
sigma = 2
```

```
# create vector for values of x that span a sufficient range of
```

```
# standard deviations on either side of the mean; here we use values
```

```
# for x that are four standard deviations on either side of the mean
```

```
x = seq(4, 20, 0.01)
```

```
# use dnorm() to calculate probabilities for each x
```

```
y = dnorm(x, mean = mu, sd = sigma)
```

```
# plot normal distribution curve
```

```
plot(x, y, type = "l", lwd = 2, col = "blue", ylab = "probability", xlab = "x")
```

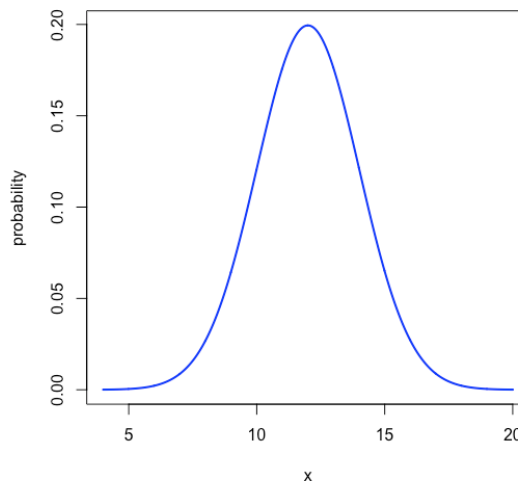


Figure 6.3.1: Plot showing the normal distribution curve for a population with $\mu = 12$ and $\sigma = 2$.

To annotate the normal distribution curve to show an area of interest to us, we use R's `polygon()` function, as illustrated here for the normal distribution curve in Figure 6.3.1, showing the area that includes values between 8 and 15.

```
# define the mean and the standard deviation
```

```
mu = 12
sigma = 2
```

```
# create vector for values of x that span a sufficient range of
# standard deviations on either side of the mean; here we use values
# for x that are four standard deviations on either side of the mean
x = seq(4, 20, 0.01)

# use dnorm() to calculate probabilities for each x
y = dnorm(x, mean = mu, sd = sigma)

# plot normal distribution curve; the options xaxt = "i" and yaxt = "i"
# force the axes to begin and end at the limits of the data

plot(x, y, type = "l", lwd = 2, col = "ivory4", ylab = "probability", xlab = "x", xaxs =
      "i", yaxs = "i")

# create vector for values of x between a lower limit of 8 and an upper limit of 15
lowlim = 8

uplim = 15

dx = seq(lowlim, uplim, 0.01)

# use polygon to fill in area; x and y are vectors of x,y coordinates
# that define the shape that is then filled using the desired color

polygon(x = c(lowlim, dx, uplim), y = c(0, dnorm(dx, mean = 12, sd = 2), 0), border = NA,
      col = "ivory4")
```

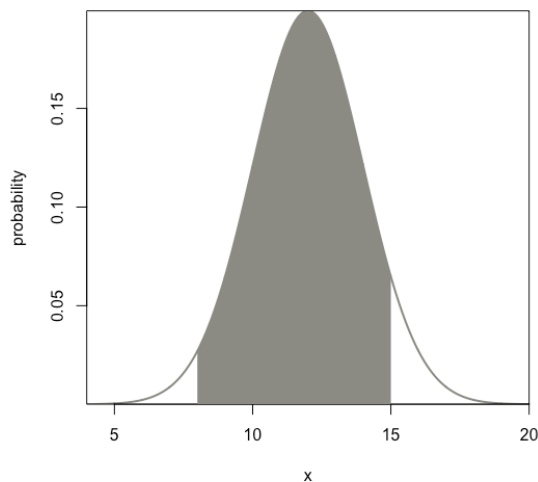


Figure 6.3.2: Plot showing the normal distribution curve for a population with $\mu = 12$ and $\sigma = 2$, and highlighting probability of obtaining a result between 8 and 15.

To find the probability of obtaining a value within the shaded are, we use R's `pnorm()` command

```
pnorm(q, mean, sd, lower.tail)
```

where `q` is a limit of interest, `mean` is the value for μ , `sd` is the value for σ , and `lower.tail` is a logical value that indicates whether we

return the probability for values below the limit (`lower.tail = TRUE`) or for values above the limit (`lower.tail = FALSE`). For example, to find the probability of obtaining a result between 8 and 15, given $\mu = 12$ and $\sigma = 2$, we use the following lines of code.

```
# find probability of obtaining a result greater than 15
prob_greater15 = pnorm(15, mean = 12, sd = 2, lower.tail = FALSE)

# find probability of obtaining a result less than 8
prob_less8 = pnorm(8, mean = 12, sd = 2, lower.tail = TRUE)

# find probability of obtaining a result between 8 and 15
prob_between = 1 - prob_greater15 - prob_less8 # display results

prob_greater15
[1] 0.0668072

prob_less8
[1] 0.02275013

prob_between
[1] 0.9104427
```

Thus, 91.04% of values fall between the limits of 8 and 15.

6.4: Using R to Find Confidence Intervals

The confidence interval for a population's mean, μ , given an experimental mean, $\bar{x}$, for n samples is defined as

$$\mu = \bar{x} \pm \frac{z\sigma}{\sqrt{n}}$$

if we know the population's standard deviation, σ , and as

$$\mu = \bar{x} \pm \frac{ts}{\sqrt{n}}$$

if we assume that the sample's standard deviation, s , is a reasonable predictor of the population's standard deviation. To find values for z we use R's `qnorm()` function, which takes the form

`qnorm(p)`

where p is the probability on one side of the normal distribution curve that a result is not included within the confidence interval. For a 95% confidence interval, $p = 0.05/2 = 0.025$ because the total probability of 0.05 is equally divided between both sides of the normal distribution. To find t we use R's `qt()` function, which takes the form

`qt(p, df)`

where p is defined as above and where df is the degrees of freedom or $n - 1$.

For example, if we have a mean of $\bar{x} = 12$ for 10 samples with a known standard deviation of $\sigma = 2$, then for the 95% confidence interval the value of z and the resulting confidence interval are

```
# for a 95% confidence interval, alpha is 0.05 and the probability, p, on either end of the distribution is 0.025;
```

```
# the value of z is positive on one side of the normal distribution and negative on the other side;
```

```
# as we are interested in just the magnitude, not the sign, we use the abs() function to return the absolute value
```

```
z = qnorm(0.025)
conf_int_pop = abs(z * 2/sqrt(10))
conf_int_pop

[1] 1.23959
```

Adding and subtracting this value from the mean defines the confidence interval, which, in this case is 12 ± 1.2 .

If we have a mean of $\bar{x} = 12$ for 10 samples with an experimental standard deviation of $s = 2$, then for the 95% confidence interval the value of t and the resulting confidence interval are

```
t = qt(p = 0.025, 9)
conf_int_samp = abs(t * 2/sqrt(10))
conf_int_samp

[1] 1.430714
```

Adding and subtracting this value from the mean defines the confidence interval, which, in this case is 12 ± 1.4 .

This page titled [6.4: Using R to Find Confidence Intervals](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [David Harvey](#).

6.5: Exercises

1. Berglund and Wichardt investigated the quantitative determination of Cr in high-alloy steels using a potentiometric titration of Cr(VI). Before the titration, samples of the steel were dissolved in acid and the chromium oxidized to Cr(VI) using peroxydisulfate. Shown here are the results (as %w/w Cr) for the analysis of a reference steel as reported in Berglund, B.; Wichardt, C. *Anal. Chim. Acta* **1990**, 236, 399–410.

16.968	16.922	16.840	16.883
16.887	16.977	16.857	16.728

Calculate the mean, the standard deviation, and the 95% confidence interval about the mean. What does this confidence interval mean?

2. In Exercise 4.3.2 you determined the mean and the variance for 10 separate tablets of Excedrin Extra Strength Pain Reliever gives the following results (in mg). The data in this problem are from Simonian, M. H.; Dinh, S.; Fray, L. A. *Spectroscopy* **1993**, 8(6), 37–47.

224.3	240.4	246.3	239.4	253.1
261.7	229.4	255.5	235.5	249.7

Assuming that $\bar{X}$ and s^2 are good approximations for μ and for σ^2 , and that the population is normally distributed, what percentage of the tablets are expected to contain more than the standard amount of 250 mg acetaminophen per tablet?

3. In Exercise 4.3.3 you determined the mean and the standard deviation for the amount of morphine hydrochloride in each of four different nominal dosages levels using data from Salem, I. I.; Galan, A. C. *Anal. Chim. Acta* 1993, 283, 334–337. All results are in mg/tablet.

100-mg tablets	60-mg tablets	30-mg tablets	10-mg tablets
99.17	54.21	28.51	9.06
94.31	55.62	26.25	8.83
95.92	57.40	25.92	9.08
94.55	57.51	28.62	
93.83	52.59	24.93	

For each dosage level, and assuming that $\bar{X}$ and s^2 are good approximations for μ and for σ^2 , and that the population is normally, what percentage of tablets contain more than the nominal amount of morphine hydrochloride per tablet?

4. Use this [link](#) to access a case study on data analysis and complete the last three investigations included in Part IV: Ways to Model Data and the first three investigations included in Part V: Ways to Draw Conclusions from Data.

7: TESTING THE SIGNIFICANCE OF DATA

A confidence interval is a useful way to report the result of an analysis because it sets limits on the expected result. In the absence of determinate error, or bias, a confidence interval based on a sample's mean indicates the range of values in which we expect to find the population's mean. When we report a 95% confidence interval for the mass of a penny as $3.117 \text{ g} \pm 0.047 \text{ g}$, for example, we are stating that there is only a 5% probability that the penny's expected mass is less than 3.070 g or more than 3.164 g.

Because a confidence interval is a statement of probability, it allows us to consider comparative questions, such as these:

“Are the results for a newly developed method to determine cholesterol in blood significantly different from those obtained using a standard method?”

“Is there a significant variation in the composition of rainwater collected at different sites downwind from a coal-burning utility plant?”

In this chapter we introduce a general approach that uses experimental data to ask and answer such questions, an approach we call significance testing.

The reliability of significance testing recently has received much attention—see Nuzzo, R. “Scientific Method: Statistical Errors,” *Nature*, **2014**, 506, 150–152 for a general discussion of the issues—so it is appropriate to begin this chapter by noting the need to ensure that our data and our research question are compatible so that we do not read more into a statistical analysis than our data allows; see Leek, J. T.; Peng, R. D. “What is the Question?” *Science*, **2015**, 347, 1314–1315 for a useful discussion of six common research questions.

In the context of analytical chemistry, significance testing often accompanies an exploratory data analysis

“Is there a reason to suspect that there is a difference between these two analytical methods when applied to a common sample?”

or an inferential data analysis.

“Is there a reason to suspect that there is a relationship between these two independent measurements?”

A statistically significant result for these types of analytical research questions generally leads to the design of additional experiments that are better suited to making predictions or to explaining an underlying causal relationship. A significance test is the first step toward building a greater understanding of an analytical problem, not the final answer to that problem!

7.1: Significance Testing

7.2: Significance Tests for Normal Distributions

7.3: Analysis of Variance

7.4: Non-Parametric Significance Tests

7.5: Using R for Significance Testing and Analysis of Variance

7.6: Exercises

7.1: Significance Testing

Let's consider the following problem. To determine if a medication is effective in lowering blood glucose concentrations, we collect two sets of blood samples from a patient. We collect one set of samples immediately before we administer the medication, and we collect the second set of samples several hours later. After we analyze the samples, we report their respective means and variances. How do we decide if the medication was successful in lowering the patient's concentration of blood glucose?

One way to answer this question is to construct a normal distribution curve for each sample, and to compare the two curves to each other. Three possible outcomes are shown in Figure 7.1.1. In Figure 7.1.1a, there is a complete separation of the two normal distribution curves, which suggests the two samples are significantly different from each other. In Figure 7.1.1b, the normal distribution curves for the two samples almost completely overlap each other, which suggests the difference between the samples is insignificant. Figure 7.1.1c, however, presents us with a dilemma. Although the means for the two samples seem different, the overlap of their normal distribution curves suggests that a significant number of possible outcomes could belong to either distribution. In this case the best we can do is to make a statement about the probability that the samples are significantly different from each other.

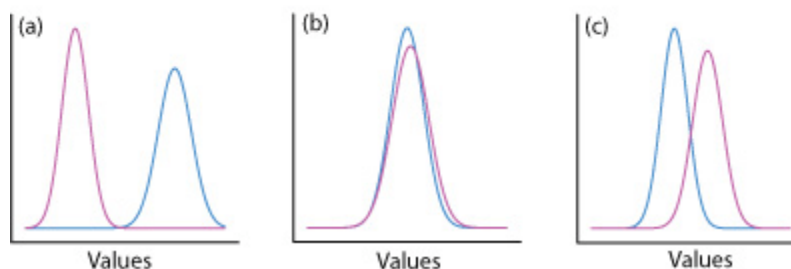


Figure 7.1.1: Three examples of the possible relationships between the normal distribution curves for two samples. In (a) the curves do not overlap, which suggests the samples are significantly different from each other. In (b) the two curves are almost identical, suggesting the samples are indistinguishable. The partial overlap of the curves in (c) means that the best we can do is evaluate the probability that there is a difference between the samples.

The process by which we determine the probability that there is a significant difference between two samples is called significance testing or hypothesis testing. Before we discuss specific examples let's first establish a general approach to conducting and interpreting a significance test.

Constructing a Significance Test

The purpose of a significance test is to determine whether the difference between two or more results is sufficiently large that we are comfortable stating that the difference cannot be explained by indeterminate errors. The first step in constructing a significance test is to state the problem as a yes or no question, such as

"Is this medication effective at lowering a patient's blood glucose levels?"

A null hypothesis and an alternative hypothesis define the two possible answers to our yes or no question. The null hypothesis, H_0 , is that indeterminate errors are sufficient to explain any differences between our results. The alternative hypothesis, H_A , is that the differences in our results are too great to be explained by random error and that they must be determinate in nature. We test the null hypothesis, which we either retain or reject. If we reject the null hypothesis, then we must accept the alternative hypothesis and conclude that the difference is significant.

Failing to reject a null hypothesis is not the same as accepting it. We retain a null hypothesis because we have insufficient evidence to prove it incorrect. It is impossible to prove that a null hypothesis is true. This is an important point and one that is easy to forget. To appreciate this point let's use this data for the mass of 100 circulating United States pennies.

Table 7.1.1. Masses for a Sample of 100 Circulating U. S. Pennies

Penny	Weight (g)	Penny	Weight (g)	Penny	Weight (g)	Penny	Weight (g)
1	3.126	26	3.073	51	3.101	76	3.086
2	3.140	27	3.084	52	3.049	77	3.123
3	3.092	28	3.148	53	3.082	78	3.115
4	3.095	29	3.047	54	3.142	79	3.055
5	3.080	30	3.121	55	3.082	80	3.057
6	3.065	31	3.116	56	3.066	81	3.097
7	3.117	32	3.005	57	3.128	82	3.066
8	3.034	33	3.115	58	3.112	83	3.113
9	3.126	34	3.103	59	3.085	84	3.102
10	3.057	35	3.086	60	3.086	85	3.033
11	3.053	36	3.103	61	3.084	86	3.112
12	3.099	37	3.049	62	3.104	87	3.103
13	3.065	38	2.998	63	3.107	88	3.198
14	3.059	39	3.063	64	3.093	89	3.103
15	3.068	40	3.055	65	3.126	90	3.126
16	3.060	41	3.181	66	3.138	91	3.111
17	3.078	42	3.108	67	3.131	92	3.126
18	3.125	43	3.114	68	3.120	93	3.052
19	3.090	44	3.121	69	3.100	94	3.113
20	3.100	45	3.105	70	3.099	95	3.085
21	3.055	46	3.078	71	3.097	96	3.117
22	3.105	47	3.147	72	3.091	97	3.142
23	3.063	48	3.104	73	3.077	98	3.031
24	3.083	49	3.146	74	3.178	99	3.083
25	3.065	50	3.095	75	3.054	100	3.104

After looking at the data we might propose the following null and alternative hypotheses.

H_0 : The mass of a circulating U.S. penny is between 2.900 g and 3.200 g

H_A : The mass of a circulating U.S. penny may be less than 2.900 g or more than 3.200 g

To test the null hypothesis we find a penny and determine its mass. If the penny's mass is 2.512 g then we can reject the null hypothesis and accept the alternative hypothesis. Suppose that the penny's mass is 3.162 g. Although this result increases our confidence in the null hypothesis, it does not prove that the null hypothesis is correct because the next penny we sample might weigh less than 2.900 g or more than 3.200 g.

After we state the null and the alternative hypotheses, the second step is to choose a confidence level for the analysis. The confidence level defines the probability that we will incorrectly reject the null hypothesis when it is, in fact, true. We can express this as our confidence that we are correct in rejecting the null hypothesis (e.g. 95%), or as the probability that we are incorrect in rejecting the null hypothesis. For the latter, the confidence level is given as α , where

$$\alpha = 1 - \frac{\text{confidence interval (\%)}}{100}$$

For a 95% confidence level, α is 0.05.

The third step is to calculate an appropriate test statistic and to compare it to a critical value. The test statistic's critical value defines a breakpoint between values that lead us to reject or to retain the null hypothesis, which is the fourth, and final, step of a significance test. As we will see in the sections that follow, how we calculate the test statistic depends on what we are comparing.

The four steps for a statistical analysis of data using a significance test:

1. Pose a question, and state the null hypothesis, H_0 , and the alternative hypothesis, H_A .
2. Choose a confidence level for the statistical analysis.
3. Calculate an appropriate test statistic and compare it to a critical value.
4. Either retain the null hypothesis, or reject it and accept the alternative hypothesis.

One-Tailed and Two-tailed Significance Tests

Suppose we want to evaluate the accuracy of a new analytical method. We might use the method to analyze a Standard Reference Material that contains a known concentration of analyte, μ . We analyze the standard several times, obtaining a mean value, $\bar{X}$, for the analyte's concentration. Our null hypothesis is that there is no difference between $\bar{X}$ and μ

$$H_0: \bar{X} = \mu$$

If we conduct the significance test at $\alpha = 0.05$, then we retain the null hypothesis if a 95% confidence interval around $\bar{X}$ contains μ . If the alternative hypothesis is

$$H_A: \bar{X} \neq \mu$$

then we reject the null hypothesis and accept the alternative hypothesis if μ lies in the shaded areas at either end of the sample's probability distribution curve (Figure 7.1.2a). Each of the shaded areas accounts for 2.5% of the area under the probability distribution curve, for a total of 5%. This is a two-tailed significance test because we reject the null hypothesis for values of μ at either extreme of the sample's probability distribution curve.

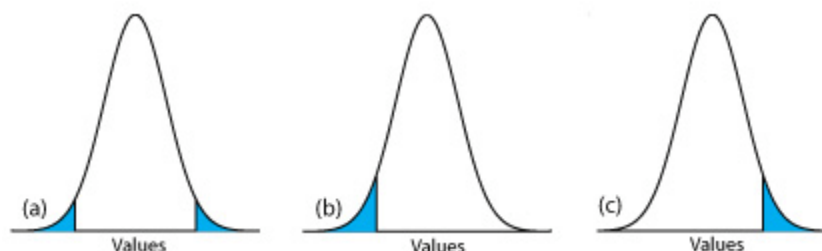


Figure 7.1.2: Examples of (a) two-tailed, and (b, c) one-tailed, significance test of $\bar{X}$ and μ . The probability distribution curves, which are normal distributions, are based on the sample's mean and standard deviation. For $\alpha = 0.05$, the blue areas account for 5% of the area under the curve. If the value of μ falls within the blue areas, then we reject the null hypothesis and accept the alternative hypothesis. We retain the null hypothesis if the value of μ falls within the unshaded area of the curve.

We can write the alternative hypothesis in two additional ways

$$H_A: \bar{X} > \mu$$

$$H_A: \bar{X} < \mu$$

rejecting the null hypothesis if μ falls within the shaded areas shown in Figure 7.1.2b or Figure 7.1.2c, respectively. In each case the shaded area represents 5% of the area under the probability distribution curve. These are examples of a one-tailed significance test.

For a fixed confidence level, a two-tailed significance test is the more conservative test because rejecting the null hypothesis requires a larger difference between the results we are comparing. In most situations we have no particular reason to expect that one result must be larger (or must be smaller) than the other result. This is the case, for example, when we evaluate the accuracy of a new analytical method. A two-tailed significance test, therefore, usually is the appropriate choice.

We reserve a one-tailed significance test for a situation where we specifically are interested in whether one result is larger (or smaller) than the other result. For example, a one-tailed significance test is appropriate if we are evaluating a medication's ability to lower blood glucose levels. In this case we are interested only in whether the glucose levels after we administer the medication are less than the glucose levels before we initiated treatment. If a patient's blood glucose level is greater after we administer the medication, then we know the answer—the medication did not work—and we do not need to conduct a statistical analysis.

Errors in Significance Testing

Because a significance test relies on probability, its interpretation is subject to error. In a significance test, α defines the probability of rejecting a null hypothesis that is true. When we conduct a significance test at $\alpha = 0.05$, there is a 5% probability that we will incorrectly reject the null hypothesis. This is known as a type 1 error, and its risk is always equivalent to α . A type 1 error in a two-tailed or a one-tailed significance tests corresponds to the shaded areas under the probability distribution curves in Figure 7.1.2.

A second type of error occurs when we retain a null hypothesis even though it is false. This is a type 2 error, and the probability of its occurrence is β . Unfortunately, in most cases we cannot calculate or estimate the value for β . The probability of a type 2 error, however, is inversely proportional to the probability of a type 1 error.

Minimizing a type 1 error by decreasing α increases the likelihood of a type 2 error. When we choose a value for α we must compromise between these two types of error. Most of the examples in this text use a 95% confidence level ($\alpha = 0.05$) because this usually is a reasonable compromise between type 1 and type 2 errors for analytical work. It is not unusual, however, to use a more stringent (e.g. $\alpha = 0.01$) or a more lenient (e.g. $\alpha = 0.10$) confidence level when the situation calls for it.

7.2: Significance Tests for Normal Distributions

A normal distribution is the most common distribution for the data we collect. Because the area between any two limits of a normal distribution curve is well defined, it is straightforward to construct and evaluate significance tests.

Note

You can review the properties of a normal distribution in Chapter 5 and Chapter 6.

Comparing $\bar{X}$ to μ

One way to validate a new analytical method is to analyze a sample that contains a known amount of analyte, μ . To judge the method's accuracy we analyze several portions of the sample, determine the average amount of analyte in the sample, $\bar{X}$, and use a significance test to compare $\bar{X}$ to μ . The null hypothesis is that the difference between $\bar{X}$ and μ is explained by indeterminate errors that affect our determination of $\bar{X}$. The alternative hypothesis is that the difference between $\bar{X}$ and μ is too large to be explained by indeterminate error.

$$H_0: \bar{X} = \mu$$

$$H_A: \bar{X} \neq \mu$$

The test statistic is t_{exp} , which we substitute into the confidence interval for μ

$$\mu = \bar{X} \pm \frac{t_{\text{exp}} s}{\sqrt{n}}$$

Rearranging this equation and solving for t_{exp}

$$t_{\text{exp}} = \frac{|\mu - \bar{X}| \sqrt{n}}{s}$$

gives the value for t_{exp} when μ is at either the right edge or the left edge of the sample's confidence interval (Figure 7.2.1a).

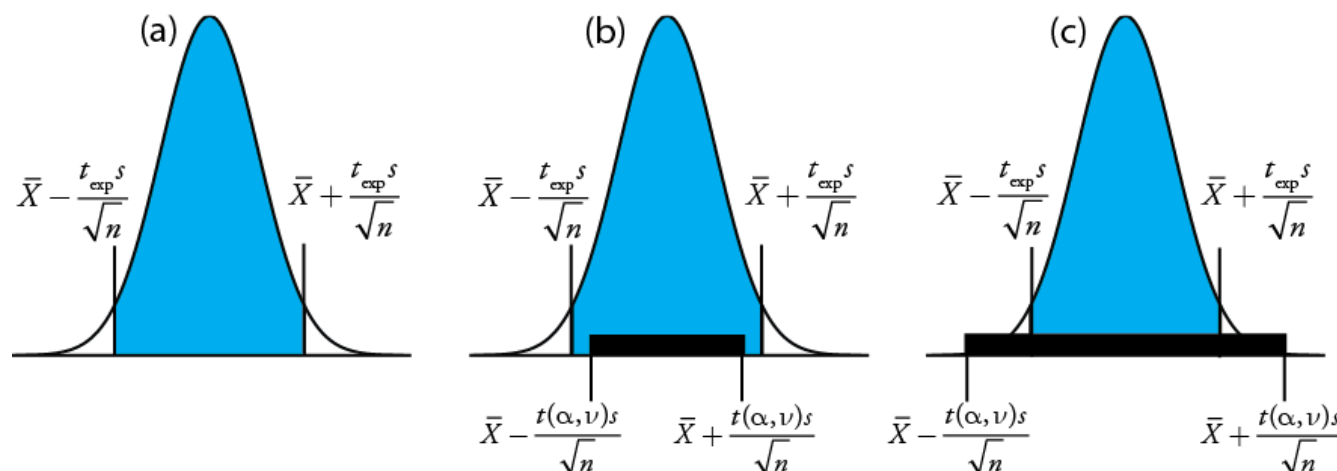


Figure 7.2.1: Relationship between a confidence interval and the result of a significance test. (a) The shaded area under the normal distribution curve shows the sample's confidence interval for μ based on t_{exp} . The solid bars in (b) and (c) show the expected confidence intervals for μ explained by indeterminate error given the choice of α and the available degrees of freedom, ν . For (b) we reject the null hypothesis because portions of the sample's confidence interval fall outside the confidence interval explained by indeterminate error. In the case of (c) we retain the null hypothesis because the confidence interval explained by indeterminate error completely encompasses the sample's confidence interval.

To determine if we should retain or reject the null hypothesis, we compare the value of t_{exp} to a critical value, $t(\alpha, \nu)$, where α is the confidence level and ν is the degrees of freedom for the sample. The critical value $t(\alpha, \nu)$ defines the largest confidence interval explained by indeterminate error. If $t_{\text{exp}} > t(\alpha, \nu)$, then our sample's confidence interval is greater than that explained by indeterminate errors (Figure 7.2.1b). In this case, we reject the null hypothesis and accept the alternative hypothesis. If $t_{\text{exp}} \leq t(\alpha, \nu)$, then our sample's confidence interval is smaller than that explained by indeterminate error, and we retain the null hypothesis (Figure 7.2.1c). Example 7.2.1 provides a typical application of this significance test, which is known as a t -test of $\bar{X}$ to μ . You will find values for $t(\alpha, \nu)$ in Appendix 2.

Example 7.2.1

Before determining the amount of Na_2CO_3 in a sample, you decide to check your procedure by analyzing a standard sample that is 98.76% w/w Na_2CO_3 . Five replicate determinations of the %w/w Na_2CO_3 in the standard gives the following results

98.71% 98.59% 98.62% 98.44% 98.58%

Using $\alpha = 0.05$, is there any evidence that the analysis is giving inaccurate results?

Solution

The mean and standard deviation for the five trials are

$$\bar{X} = 98.59 \quad s = 0.0973$$

Because there is no reason to believe that the results for the standard must be larger or smaller than μ , a two-tailed t -test is appropriate. The null hypothesis and alternative hypothesis are

$$H_0: \bar{X} = \mu \quad H_A: \bar{X} \neq \mu$$

The test statistic, t_{exp} , is

$$t_{\text{exp}} = \frac{|\mu - \bar{X}| \sqrt{n}}{s} = \frac{|98.76 - 98.59| \sqrt{5}}{0.0973} = 3.91$$

The critical value for $t(0.05, 4)$ from Appendix 2 is 2.78. Since t_{exp} is greater than $t(0.05, 4)$, we reject the null hypothesis and accept the alternative hypothesis. At the 95% confidence level the difference between $\bar{X}$ and μ is too large to be explained by indeterminate sources of error, which suggests there is a determinate source of error that affects the analysis.

Note

There is another way to interpret the result of this t -test. Knowing that t_{exp} is 3.91 and that there are 4 degrees of freedom, we use Appendix 2 to estimate the value of α that corresponds to a $t(\alpha, 4)$ of 3.91. From Appendix 2, $t(0.02, 4)$ is 3.75 and $t(0.01, 4)$ is 4.60. Although we can reject the null hypothesis at the 98% confidence level, we cannot reject it at the 99% confidence level. For a discussion of the advantages of this approach, see J. A. C. Sterne and G. D. Smith "Sifting the evidence—what's wrong with significance tests?" *BMJ* **2001**, 322, 226–231.

Earlier we made the point that we must exercise caution when we interpret the result of a statistical analysis. We will keep returning to this point because it is an important one. Having determined that a result is inaccurate, as we did in Example 7.2.1, the next step is to identify and to correct the error. Before we expend time and money on this, however, we first should critically examine our data. For example, the smaller the value of s , the larger the value of t_{exp} . If the standard deviation for our analysis is unrealistically small, then the probability of a type 2 error increases. Including a few additional replicate analyses of the standard and reevaluating the t -test may strengthen our evidence for a determinate error, or it may show us that there is no evidence for a determinate error.

Comparing s^2 to σ^2

If we analyze regularly a particular sample, we may be able to establish an expected variance, σ^2 , for the analysis. This often is the case, for example, in a clinical lab that analyzes hundreds of blood samples each day. A few replicate analyses of a single sample gives a sample variance, s^2 , whose value may or may not differ significantly from σ^2 .

We can use an F -test to evaluate whether a difference between s^2 and σ^2 is significant. The null hypothesis is $H_0: s^2 = \sigma^2$ and the alternative hypothesis is $H_A: s^2 \neq \sigma^2$. The test statistic for evaluating the null hypothesis is F_{exp} , which is given as either

$$F_{\text{exp}} = \frac{s^2}{\sigma^2} \text{ if } s^2 > \sigma^2 \text{ or } F_{\text{exp}} = \frac{\sigma^2}{s^2} \text{ if } \sigma^2 > s^2$$

depending on whether s^2 is larger or smaller than σ^2 . This way of defining F_{exp} ensures that its value is always greater than or equal to one.

If the null hypothesis is true, then F_{exp} should equal one; however, because of indeterminate errors, F_{exp} , usually is greater than one. A critical value, $F(\alpha, \nu_{\text{num}}, \nu_{\text{den}})$, is the largest value of F_{exp} that we can attribute to indeterminate error given the specified significance level, α , and the degrees of freedom for the variance in the numerator, ν_{num} , and the variance in the denominator, ν_{den} . The degrees of freedom for s^2 is $n - 1$, where n is the number of replicates used to determine the sample's variance, and the degrees of freedom for σ^2 is defined as infinity, ∞ . Critical values of F for $\alpha = 0.05$ are listed in Appendix 3 for both one-tailed and two-tailed F -tests.

Example 7.2.2

A manufacturer's process for analyzing aspirin tablets has a known variance of 25. A sample of 10 aspirin tablets is selected and analyzed for the amount of aspirin, yielding the following results in mg aspirin/tablet.

254 249 252 252 249 249 250 247 251 252

Determine whether there is evidence of a significant difference between the sample's variance and the expected variance at $\alpha = 0.05$.

Solution

The variance for the sample of 10 tablets is 4.3. The null hypothesis and alternative hypotheses are

$$H_0: s^2 = \sigma^2 \quad H_A: s^2 \neq \sigma^2$$

and the value for F_{exp} is

$$F_{\text{exp}} = \frac{\sigma^2}{s^2} = \frac{25}{4.3} = 5.8$$

The critical value for $F(0.05, \infty, 9)$ from Appendix 3 is 3.333. Since F_{exp} is greater than $F(0.05, \infty, 9)$, we reject the null hypothesis and accept the alternative hypothesis that there is a significant difference between the sample's variance and the expected variance. One explanation for the difference might be that the aspirin tablets were not selected randomly.

Comparing Variances for Two Samples

We can extend the F -test to compare the variances for two samples, A and B , by rewriting our equation for F_{exp} as

$$F_{\text{exp}} = \frac{s_A^2}{s_B^2}$$

defining A and B so that the value of F_{exp} is greater than or equal to 1.

Example 7.2.3

The table below shows results for two experiments to determine the mass of a circulating U.S. penny. Determine whether there is a difference in the variances of these analyses at $\alpha = 0.05$.

First Experiment		Second Experiment	
Penny	Mass (g)	Penny	Mass (g)
1	3.080	1	3.052
2	3.094	2	3.141
3	3.107	3	3.083
4	3.056	4	3.083
5	3.112	5	3.048
6	3.174		
7	3.198		

Solution

The standard deviations for the two experiments are 0.051 for the first experiment (A) and 0.037 for the second experiment (B). The null and alternative hypotheses are

$$H_0: s_A^2 = s_B^2 \quad H_A: s_A^2 \neq s_B^2$$

and the value of F_{exp} is

$$F_{\text{exp}} = \frac{s_A^2}{s_B^2} = \frac{(0.051)^2}{(0.037)^2} = \frac{0.00260}{0.00137} = 1.90$$

From Appendix 3 the critical value for $F(0.05, 6, 4)$ is 9.197. Because $F_{\text{exp}} < F(0.05, 6, 4)$, we retain the null hypothesis. There is no evidence at $\alpha = 0.05$ to suggest that the difference in variances is significant.

Comparing Means for Two Samples

Three factors influence the result of an analysis: the method, the sample, and the analyst. We can study the influence of these factors by conducting experiments in which we change one factor while holding constant the other factors. For example, to compare two analytical methods we can have the same analyst apply each method to the same sample and then examine the resulting means. In a similar fashion, we can design experiments to compare two analysts or to compare two samples.

Before we consider the significance tests for comparing the means of two samples, we need to understand the difference between unpaired data and paired data. This is a critical distinction and learning to distinguish between these two types of data is important. Here are two simple examples that highlight the difference between unpaired data and paired data. In each example the goal is to compare two balances by weighing pennies.

- Example 1: We collect 10 pennies and weigh each penny on each balance. This is an example of paired data because we use the same 10 pennies to evaluate each balance.
- Example 2: We collect 10 pennies and divide them into two groups of five pennies each. We weigh the pennies in the first group on one balance and we weigh the second group of pennies on the other balance. Note that no penny is weighed on both balances. This is an example of unpaired data because we evaluate each balance using a different sample of pennies.

In both examples the samples of 10 pennies were drawn from the same population; the difference is how we sampled that population. We will learn why this distinction is important when we review the significance test for paired data; first, however, we present the significance test for unpaired data.

Note

One simple test for determining whether data are paired or unpaired is to look at the size of each sample. If the samples are of different size, then the data must be unpaired. The converse is not true. If two samples are of equal size, they may be paired or unpaired.

Unpaired Data

Consider two analyses, A and B , with means of $\bar{X}_A$ and $\bar{X}_B$, and standard deviations of s_A and s_B . The confidence intervals for μ_A and for μ_B are

$$\mu_A = \bar{X}_A \pm \frac{ts_A}{\sqrt{n_A}}$$

$$\mu_B = \bar{X}_B \pm \frac{ts_B}{\sqrt{n_B}}$$

where n_A and n_B are the sample sizes for A and for B . Our null hypothesis, $H_0: \mu_A = \mu_B$, is that any difference between μ_A and μ_B is the result of indeterminate errors that affect the analyses. The alternative hypothesis, $H_A: \mu_A \neq \mu_B$, is that the difference between μ_A and μ_B is too large to be explained by indeterminate error.

To derive an equation for t_{exp} , we assume that μ_A equals μ_B , and combine the equations for the two confidence intervals

$$\bar{X}_A \pm \frac{t_{\text{exp}}s_A}{\sqrt{n_A}} = \bar{X}_B \pm \frac{t_{\text{exp}}s_B}{\sqrt{n_B}}$$

Solving for $|\bar{X}_A - \bar{X}_B|$ and using a propagation of uncertainty, gives

$$|\bar{X}_A - \bar{X}_B| = t_{\text{exp}} \times \sqrt{\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}}$$

Finally, we solve for t_{exp}

$$t_{\text{exp}} = \frac{|\bar{X}_A - \bar{X}_B|}{\sqrt{\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}}}$$

and compare it to a critical value, $t(\alpha, \nu)$, where α is the probability of a type 1 error, and ν is the degrees of freedom.

Thus far our development of this t -test is similar to that for comparing $\bar{X}$ to μ , and yet we do not have enough information to evaluate the t -test. Do you see the problem? With two independent sets of data it is unclear how many degrees of freedom we have.

Suppose that the variances s_A^2 and s_B^2 provide estimates of the same σ^2 . In this case we can replace s_A^2 and s_B^2 with a pooled variance, s_{pool}^2 , that is a better estimate for the variance. Thus, our equation for t_{exp} becomes

$$t_{\text{exp}} = \frac{|\bar{X}_A - \bar{X}_B|}{s_{\text{pool}} \times \sqrt{\frac{1}{n_A} + \frac{1}{n_B}}} = \frac{|\bar{X}_A - \bar{X}_B|}{s_{\text{pool}}} \times \sqrt{\frac{n_A n_B}{n_A + n_B}}$$

where s_{pool} , the pooled standard deviation, is

$$s_{\text{pool}} = \sqrt{\frac{(n_A - 1)s_A^2 + (n_B - 1)s_B^2}{n_A + n_B - 2}}$$

The denominator of this equation shows us that the degrees of freedom for a pooled standard deviation is $n_A + n_B - 2$, which also is the degrees of freedom for the t -test. Note that we lose two degrees of freedom because the calculations for s_A^2 and s_B^2

require the prior calculation of $\bar{X}_A$ and $\bar{X}_B$.

Note

So how do you determine if it is okay to pool the variances? Use an F -test.

If s_A^2 and s_B^2 are significantly different, then we calculate t_{exp} using the following equation. In this case, we find the degrees of freedom using the following imposing equation.

$$\nu = \frac{\left(\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}\right)^2}{\frac{\left(\frac{s_A^2}{n_A}\right)^2}{n_A+1} + \frac{\left(\frac{s_B^2}{n_B}\right)^2}{n_B+1}} - 2$$

Because the degrees of freedom must be an integer, we round to the nearest integer the value of ν obtained from this equation.

Note

The equation above for the degrees of freedom is from [Miller, J.C.](#); Miller, J.N. *Statistics for Analytical Chemistry*, 2nd Ed., Ellis-Horward: Chichester, UK, 1988. In the 6th Edition, the authors note that several different equations have been suggested for the number of degrees of freedom for t when s_A and s_B differ, reflecting the fact that the determination of degrees of freedom an approximation. An alternative equation—which is used by statistical software packages, such as R, Minitab, Excel—is

$$\nu = \frac{\left(\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}\right)^2}{\frac{\left(\frac{s_A^2}{n_A}\right)^2}{n_A-1} + \frac{\left(\frac{s_B^2}{n_B}\right)^2}{n_B-1}} = \frac{\left(\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}\right)^2}{\frac{s_A^4}{n_A^2(n_A-1)} + \frac{s_B^4}{n_B^2(n_B-1)}}$$

For typical problems in analytical chemistry, the calculated degrees of freedom is reasonably insensitive to the choice of equation.

Regardless of whether how we calculate t_{exp} , we reject the null hypothesis if t_{exp} is greater than $t(\alpha, \nu)$ and retain the null hypothesis if t_{exp} is less than or equal to $t(\alpha, \nu)$.

Example 7.2.4

Example 7.2.3 provides results for two experiments to determine the mass of a circulating U.S. penny. Determine whether there is a difference in the means of these analyses at $\alpha = 0.05$.

Solution

First we use an F -test to determine whether we can pool the variances. We completed this analysis in Example 7.2.3, finding no evidence of a significant difference, which means we can pool the standard deviations, obtaining

$$s_{\text{pool}} = \sqrt{\frac{(7-1)(0.051)^2 + (5-1)(0.037)^2}{7+5-2}} = 0.0459$$

with 10 degrees of freedom. To compare the means we use the following null hypothesis and alternative hypotheses

$$H_0: \mu_A = \mu_B \quad H_A: \mu_A \neq \mu_B$$

Because we are using the pooled standard deviation, we calculate t_{exp} as

$$t_{\text{exp}} = \frac{|3.117 - 3.081|}{0.0459} \times \sqrt{\frac{7 \times 5}{7 + 5}} = 1.34$$

The critical value for $t(0.05, 10)$, from Appendix 2, is 2.23. Because t_{exp} is less than $t(0.05, 10)$ we retain the null hypothesis. For $\alpha = 0.05$ we do not have evidence that the two sets of pennies are significantly different.

Example 7.2.5

One method for determining the %w/w Na_2CO_3 in soda ash is to use an acid–base titration. When two analysts analyze the same sample of soda ash they obtain the results shown here.

Analyst A: 86.82% 87.04% 86.93% 87.01% 86.20% 87.00%

Analyst B: 81.01% 86.15% 81.73% 83.19% 80.27% 83.93%

Determine whether the difference in the mean values is significant at $\alpha = 0.05$.

Solution

We begin by reporting the mean and standard deviation for each analyst.

$$\bar{X}_A = 86.83\% \quad s_A = 0.32\%$$

$$\bar{X}_B = 82.71\% \quad s_B = 2.16\%$$

To determine whether we can use a pooled standard deviation, we first complete an F -test using the following null and alternative hypotheses.

$$H_0: s_A^2 = s_B^2 \quad H_A: s_A^2 \neq s_B^2$$

Calculating F_{exp} , we obtain a value of

$$F_{\text{exp}} = \frac{(2.16)^2}{(0.32)^2} = 45.6$$

Because F_{exp} is larger than the critical value of 7.15 for $F(0.05, 5, 5)$ from Appendix 3, we reject the null hypothesis and accept the alternative hypothesis that there is a significant difference between the variances; thus, we cannot calculate a pooled standard deviation.

To compare the means for the two analysts we use the following null and alternative hypotheses.

$$H_0: \bar{X}_A = \bar{X}_B \quad H_A: \bar{X}_A \neq \bar{X}_B$$

Because we cannot pool the standard deviations, we calculate t_{exp} as

$$t_{\text{exp}} = \frac{|86.83 - 82.71|}{\sqrt{\frac{(0.32)^2}{6} + \frac{(2.16)^2}{6}}} = 4.62$$

and calculate the degrees of freedom as

$$\nu = \frac{\left(\frac{(0.32)^2}{6} + \frac{(2.16)^2}{6}\right)^2}{\frac{\left(\frac{(0.32)^2}{6}\right)^2}{6+1} + \frac{\left(\frac{(2.16)^2}{6}\right)^2}{6+1}} - 2 = 5.3 \approx 5$$

From Appendix 2, the critical value for $t(0.05, 5)$ is 2.57. Because t_{exp} is greater than $t(0.05, 5)$ we reject the null hypothesis and accept the alternative hypothesis that the means for the two analysts are significantly different at $\alpha = 0.05$.

Paired Data

Suppose we are evaluating a new method for monitoring blood glucose concentrations in patients. An important part of evaluating a new method is to compare it to an established method. What is the best way to gather data for this study? Because the variation in the blood glucose levels amongst patients is large we may be unable to detect a small, but significant difference between the methods if we use different patients to gather data for each method. Using paired data, in which the we analyze each patient's blood using both methods, prevents a large variance within a population from adversely affecting a t -test of means.

Note

Typical blood glucose levels for most non-diabetic individuals ranges between 80–120 mg/dL (4.4–6.7 mM), rising to as high as 140 mg/dL (7.8 mM) shortly after eating. Higher levels are common for individuals who are pre-diabetic or diabetic.

When we use paired data we first calculate the individual differences, d_i , between each sample's paired results. Using these individual differences, we then calculate the average difference, $\bar{d}$, and the standard deviation of the differences, s_d . The null hypothesis, $H_0: d = 0$, is that there is no difference between the two samples, and the alternative hypothesis, $H_A: d \neq 0$, is that the difference between the two samples is significant.

The test statistic, t_{exp} , is derived from a confidence interval around $\bar{d}$

$$t_{\text{exp}} = \frac{|\bar{d}| \sqrt{n}}{s_d}$$

where n is the number of paired samples. As is true for other forms of the t -test, we compare t_{exp} to $t(\alpha, \nu)$, where the degrees of freedom, ν , is $n - 1$. If t_{exp} is greater than $t(\alpha, \nu)$, then we reject the null hypothesis and accept the alternative hypothesis. We retain the null hypothesis if t_{exp} is less than or equal to $t(\alpha, \nu)$. This is known as a paired t -test.

Example 7.2.6

Marecek et. al. developed a new electrochemical method for the rapid determination of the concentration of the antibiotic monensin in fermentation vats [Marecek, V.; Janchenova, H.; Brezina, M.; Betti, M. *Anal. Chim. Acta* **1991**, *244*, 15–19]. The standard method for the analysis is a test for microbiological activity, which is both difficult to complete and time-consuming. Samples were collected from the fermentation vats at various times during production and analyzed for the concentration of monensin using both methods. The results, in parts per thousand (ppt), are reported in the following table.

Sample	Microbiological	Electrochemical
1	129.5	132.3
2	89.6	91.0
3	76.6	73.6
4	52.2	58.2
5	110.8	104.2
6	50.4	49.9
7	72.4	82.1
8	141.4	154.1
9	75.0	73.4
10	34.1	38.1
11	60.3	60.1

Is there a significant difference between the methods at $\alpha = 0.05$?

Solution

Acquiring samples over an extended period of time introduces a substantial time-dependent change in the concentration of monensin. Because the variation in concentration between samples is so large, we use a paired t -test with the following null and alternative hypotheses.

$$H_0: \bar{d} = 0 \quad H_A: \bar{d} \neq 0$$

Defining the difference between the methods as

$$d_i = (X_{\text{elect}})_i - (X_{\text{micro}})_i$$

we calculate the difference for each sample.

sample	1	2	3	4	5	6	7	8	9	10	11
d_i	2.8	1.4	-3.0	6.0	-6.6	-0.5	9.7	12.7	-1.6	4.0	-0.2

The mean and the standard deviation for the differences are, respectively, 2.25 ppt and 5.63 ppt. The value of t_{exp} is

$$t_{\text{exp}} = \frac{|2.25|\sqrt{11}}{5.63} = 1.33$$

which is smaller than the critical value of 2.23 for $t(0.05, 10)$ from Appendix 2. We retain the null hypothesis and find no evidence for a significant difference in the methods at $\alpha = 0.05$.

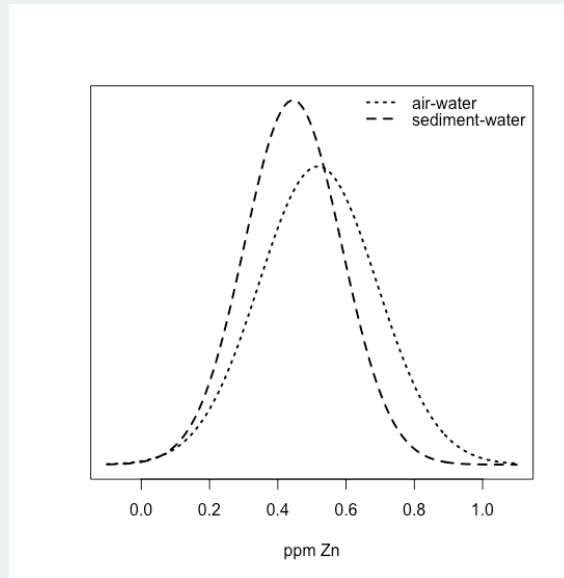
One important requirement for a paired t -test is that the determinate and the indeterminate errors that affect the analysis must be independent of the analyte's concentration. If this is not the case, then a sample with an unusually high concentration of analyte will have an unusually large d_i . Including this sample in the calculation of $\bar{d}$ and s_d gives a biased estimate for the expected mean and standard deviation. This rarely is a problem for samples that span a limited range of analyte concentrations, such as those in Example 7.2.4 or Exercise 7.2.6. When paired data span a wide range of concentrations, however, the magnitude of the determinate and indeterminate sources of error may not be independent of the analyte's concentration; when true, a paired t -test may give misleading results because the paired data with the largest absolute determinate and indeterminate errors will dominate $\bar{d}$. In this situation a regression analysis, which is the subject of the next chapter, is more appropriate method for comparing the data.

Note

The importance of distinguishing between paired and unpaired data is worth examining more closely. The following is data from some work I completed with a colleague in which we were looking at concentration of Zn in Lake Erie at the air-water interface and the sediment-water interface.

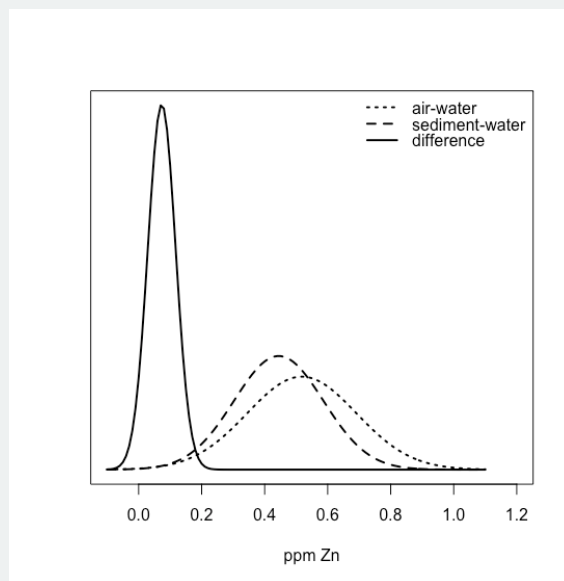
sample site	ppm Zn at air-water interface	ppm Zn at the sediment-water interface
1	0.430	0.415
2	0.266	0.238
3	0.457	0.390
4	0.531	0.410
5	0.707	0.605
6	0.716	0.609

The mean and the standard deviation for the ppm Zn at the air-water interface are 0.5178 ppm and 0.01732 ppm, and the mean and the standard deviation for the ppm Zn at the sediment-water interface are 0.4445 ppm and 0.1418 ppm. We can use these values to draw normal distributions for both by letting the means and the standard deviations for the samples, $\bar{X}$ and s , serve as estimates for the means and the standard deviations for the population, μ and σ . As we see in the following figure



the two distributions overlap strongly, suggesting that a t -test of their means is not likely to find evidence of a difference. And yet, we also see that for each site, the concentration of Zn at the sediment-water interface is less than that at the air-water interface. In this case, the difference between the concentration of Zn at individual sites is sufficiently large that it masks our ability to see the difference between the two interfaces.

If we take the differences between the air-water and sediment-water interfaces, we have values of 0.015, 0.028, 0.067, 0.121, 0.102, and 0.107 ppm Zn, with a mean of 0.07333 ppm Zn and a standard deviation of 0.04410 ppm Zn. Superimposing all three normal distributions



shows clearly that most of the normal distribution for the differences lies above zero, suggesting that a t -test might show evidence that the difference is significant.

Outliers

In chapter 7.1 we examined a data set consisting of the masses of 100 circulating United States penny. Table 7.2.1 provides one more data set. Do you notice anything unusual in this data? Of the 100 pennies included in the earlier table, no penny has a mass of less than 3 g. In this table, however, the mass of one penny is less than 3 g. We might ask whether this penny's mass is so different from the other pennies that it is in error.

Table 7.2.1. Mass (g) for Additional Sample of Circulating U. S. Pennies

3.067	2.514	3.094
3.049	3.048	3.109
3.039	3.079	3.102

A measurement that is not consistent with other measurements is called an outlier. An outlier might exist for many reasons: the outlier might belong to a different population

Is this a Canadian penny?

or the outlier might be a contaminated or an otherwise altered sample

Is the penny damaged or unusually dirty?

or the outlier may result from an error in the analysis

Did we forget to tare the balance?

Regardless of its source, the presence of an outlier compromises any meaningful analysis of our data. There are many significance tests that we can use to identify a potential outlier, three of which we present here.

Dixon's Q-Test

One of the most common significance tests for identifying an outlier is Dixon's Q-test. The null hypothesis is that there are no outliers, and the alternative hypothesis is that there is an outlier. The Q-test compares the gap between the suspected outlier and its nearest numerical neighbor to the range of the entire data set (Figure 7.2.2).

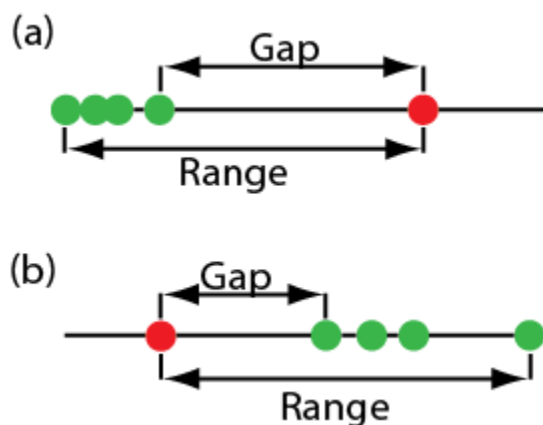


Figure 7.2.2: Dotplots showing the distribution of two data sets containing a possible outlier. In (a) the possible outlier's value is larger than the remaining data, and in (b) the possible outlier's value is smaller than the remaining data.

The test statistic, Q_{exp} , is

$$Q_{exp} = \frac{\text{gap}}{\text{range}} = \frac{|\text{outlier's value} - \text{nearest value}|}{\text{largest value} - \text{smallest value}}$$

This equation is appropriate for evaluating a single outlier. Other forms of Dixon's Q-test allow its extension to detecting multiple

outliers [Rorabacher, D. B. *Anal. Chem.* **1991**, 63, 139–146].

The value of Q_{exp} is compared to a critical value, $Q(\alpha, n)$, where α is the probability that we will reject a valid data point (a type 1 error) and n is the total number of data points. To protect against rejecting a valid data point, usually we apply the more conservative two-tailed Q-test, even though the possible outlier is the smallest or the largest value in the data set. If Q_{exp} is greater than $Q(\alpha, n)$, then we reject the null hypothesis and may exclude the outlier. We retain the possible outlier when Q_{exp} is less than or equal to $Q(\alpha, n)$. Table 7.2.2 provides values for $Q(\alpha, n)$ for a data set that has 3–10 values. A more extensive table is in Appendix 4. Values for $Q(\alpha, n)$ assume an underlying normal distribution.

Table 7.2.2: Dixon's Q-Test

n	$Q(0.05, n)$
3	0.970
4	0.829
5	0.710
6	0.625
7	0.568
8	0.526
9	0.493
10	0.466

Grubb's Test

Although Dixon's Q-test is a common method for evaluating outliers, it is no longer favored by the International Standards Organization (ISO), which recommends the Grubb's test. There are several versions of Grubb's test depending on the number of potential outliers. Here we will consider the case where there is a single suspected outlier.

Note

For details on this recommendation, see International Standards ISO Guide 5752-2 "Accuracy (trueness and precision) of measurement methods and results—Part 2: basic methods for the determination of repeatability and reproducibility of a standard measurement method," 1994.

The test statistic for Grubb's test, G_{exp} , is the distance between the sample's mean, $\bar{X}$, and the potential outlier, X_{out} , in terms of the sample's standard deviation, s .

$$G_{\text{exp}} = \frac{|X_{\text{out}} - \bar{X}|}{s}$$

We compare the value of G_{exp} to a critical value $G(\alpha, n)$, where α is the probability that we will reject a valid data point and n is the number of data points in the sample. If G_{exp} is greater than $G(\alpha, n)$, then we may reject the data point as an outlier, otherwise we retain the data point as part of the sample. Table 7.2.3 provides values for $G(0.05, n)$ for a sample containing 3–10 values. A more extensive table is in Appendix 5. Values for $G(\alpha, n)$ assume an underlying normal distribution.

Table 7.2.3: Grubb's Test

n	$G(0.05, n)$
3	1.115
4	1.481
5	1.715
6	1.887
7	2.020
8	2.126
9	2.215
10	2.290

Chauvenet's Criterion

Our final method for identifying an outlier is Chauvenet's criterion. Unlike Dixon's Q-Test and Grubb's test, you can apply this method to any distribution as long as you know how to calculate the probability for a particular outcome. Chauvenet's criterion states that we can reject a data point if the probability of obtaining the data point's value is less than $(2n^{-1})$, where n is the size of the sample. For example, if $n = 10$, a result with a probability of less than $(2 \times 10)^{-1}$, or 0.05, is considered an outlier.

To calculate a potential outlier's probability we first calculate its standardized deviation, z

$$z = \frac{|X_{\text{out}} - \bar{X}|}{s}$$

where X_{out} is the potential outlier, $\bar{X}$ is the sample's mean and s is the sample's standard deviation. Note that this equation is identical to the equation for G_{exp} in the Grubb's test. For a normal distribution, we can find the probability of obtaining a value of z using the probability table in Appendix 1.

Example 7.2.7

Table 7.2.1 contains the masses for nine circulating United States pennies. One entry, 2.514 g, appears to be an outlier. Determine if this penny is an outlier using a Q-test, Grubb's test, and Chauvenet's criterion. For the Q-test and Grubb's test, let $\alpha = 0.05$.

Solution

For the Q-test the value for Q_{exp} is

$$Q_{\text{exp}} = \frac{|2.514 - 3.039|}{3.109 - 2.514} = 0.882$$

From Table 7.2.2, the critical value for $Q(0.05, 9)$ is 0.493. Because Q_{exp} is greater than $Q(0.05, 9)$, we can assume the penny with a mass of 2.514 g likely is an outlier.

For Grubb's test we first need the mean and the standard deviation, which are 3.011 g and 0.188 g, respectively. The value for G_{exp} is

$$G_{\text{exp}} = \frac{|2.514 - 3.011|}{0.188} = 2.64$$

Using Table 7.2.3, we find that the critical value for $G(0.05, 9)$ is 2.215. Because G_{exp} is greater than $G(0.05, 9)$, we can assume that the penny with a mass of 2.514 g likely is an outlier.

For Chauvenet's criterion, the critical probability is $(2 \times 9)^{-1}$, or 0.0556. The value of z is the same as G_{exp} , or 2.64. Using Appendix 1, the probability for $z = 2.64$ is 0.00415. Because the probability of obtaining a mass of 0.2514 g is less than the critical probability, we can assume the penny with a mass of 2.514 g likely is an outlier.

You should exercise caution when using a significance test for outliers because there is a chance you will reject a valid result. In addition, you should avoid rejecting an outlier if it leads to a precision that is much better than expected based on a propagation of uncertainty. Given these concerns it is not surprising that some statisticians caution against the removal of outliers [Deming, W. E. *Statistical Analysis of Data*; Wiley: New York, 1943 (republished by Dover: New York, 1961); p. 171].

Note

You also can adopt a more stringent requirement for rejecting data. When using the Grubb's test, for example, the ISO 5752 guidelines suggest retaining a value if the probability for rejecting it is greater than $\alpha = 0.05$, and flagging a value as a "straggler" if the probability for rejecting it is between $\alpha = 0.05$ and $\alpha = 0.01$. A "straggler" is retained unless there is compelling reason for its rejection. The guidelines recommend using $\alpha = 0.01$ as the minimum criterion for rejecting a possible outlier.

On the other hand, testing for outliers can provide useful information if we try to understand the source of the suspected outlier. For example, the outlier in Table 7.2.1 represents a significant change in the mass of a penny (an approximately 17% decrease in mass), which is the result of a change in the composition of the U.S. penny. In 1982 the composition of a U.S. penny changed from a brass alloy that was 95% w/w Cu and 5% w/w Zn (with a nominal mass of 3.1 g), to a pure zinc core covered with copper (with a nominal mass of 2.5 g) [Richardson, T. H. *J. Chem. Educ.* **1991**, 68, 310–311]. The pennies in Table 7.2.1, therefore, were drawn from different populations.

7.3: Analysis of Variance

Consider the following data, which shows the stability of a reagent under different conditions for storing samples; all values are percent recoveries, so a result of 100 indicates that the reagent's concentration remains unchanged and that there was no degradation.

trial/treatment	A (total dark)	B (subdued light)	C (full light)
1	101	100	90
2	101	99	92
3	104	101	94

To determine if light has a significant affect on the reagent's stability, we might choose to perform a series of t -tests, comparing all possible mean values; in this case we need three such tests:

- compare A to B
- compare A to C
- compare B to C

Each such test has a probability of a type I error of α_{test} . The total probability of a type I error across k tests, α_{total} , is

$$\alpha_{total} = 1 - (1 - \alpha_{test})^k$$

For three such tests using $\alpha = 0.05$, we have

$$\alpha_{total} = 1 - (1 - 0.05)^3 = 0.143$$

or a 14.3% probability of a type I error. The relationship between the number of conditions, n , and the number of tests, k , is

$$k = \frac{n(n-1)}{2}$$

which means that k grows quickly as n increases, as shown in Figure 7.3.1.

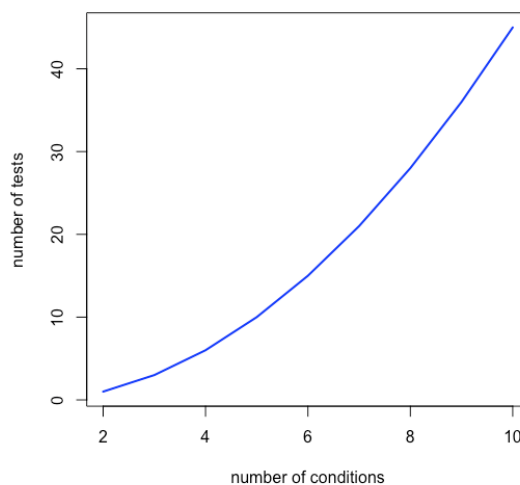


Figure 7.3.1: Plot that shows the growth in the number of tests needed to complete a significance test for every possible pair of conditions.

and that the magnitude of a type I error increases quickly as well, as seen in Figure 7.3.2.

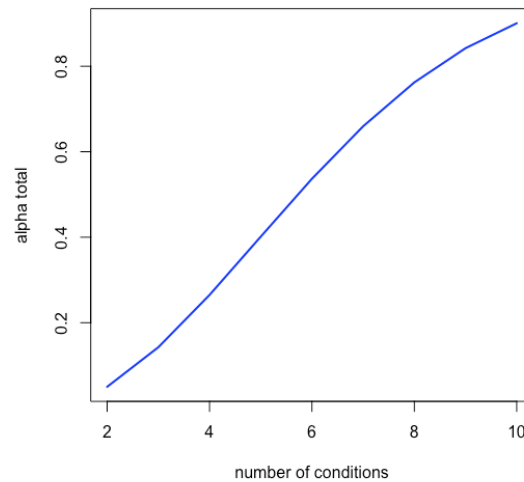


Figure 7.3.2: Plot that shows the increase in α_{total} when we complete a significance test for every possible pair of conditions.

We can compensate for this problem by decreasing α_{test} for each independent test so that α_{total} is equal to our desired probability; thus, for $n = 3$ we have $k = 3$, and to achieve an α_{total} of 0.05 each individual value of α_{test} be

$$\alpha_{test} = 1 - (1 - 0.05)^{1/3} = 0.017$$

Values of α_{test} decrease quickly, as seen in Figure 7.3.3.

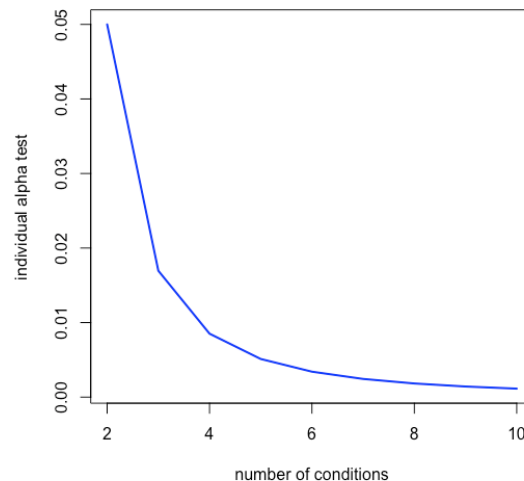


Figure 7.3.2: Plot that shows the value of α_{test} for individual significance tests to achieve an α_{total} of 0.05 based on the number of conditions being compared.

The problem here is that we are searching for a significant difference on a pair-wise basis without any evidence that the overall variation in the data across all conditions (also known as treatments) is sufficiently large that it cannot be explained by experimental uncertainty (that is, random error) only. One way to determine if there is a systematic error in the data set, without identifying the source of the systematic error, is to compare the variation within each treatment to the variation between the treatments. We assume that the variation within each treatment reflects uncertainty in the analytical method (random errors) and that the variation between the treatments includes both the method's uncertainty and any systematic errors in the individual treatments. If the variation between the treatments is significantly greater than the variation within the treatments, then a systematic error seems likely. We call this process an analysis of variance, or ANOVA; for one independent variable (the amount of light in this case), it is a one-way analysis of variance.

The basic details of a one-way ANOVA calculation are as follows:

Step 1: Treat the data as one large data set and calculate its mean and its variance, which we call the global mean, $\bar{\bar{x}}$, and the global variance, $\bar{s}^2$.

$$\bar{\bar{x}} = \frac{\sum_{i=1}^h \sum_{j=1}^{n_i} x_{ij}}{N}$$

$$\bar{s}^2 = \frac{\sum_{i=1}^h \sum_{j=1}^{n_i} (x_{ij} - \bar{\bar{x}})^2}{N - 1}$$

where h is the number of treatments, n_i is the number of replicates for the i^{th} treatment, and N is the total number of measurements.

Step 2: Calculate the within-sample variance, s_w^2 , using the mean for each treatment, $\bar{x}_i$, and the replicates for that treatment.

$$s_w^2 = \frac{\sum_{i=1}^h \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)^2}{N - h}$$

Step 3: Calculate the between-sample variance, s_b^2 , using the means for each treatment and the global mean

$$s_b^2 = \frac{\sum_{i=1}^h \sum_{j=1}^{n_i} (\bar{x}_i - \bar{\bar{x}})^2}{h - 1} = \frac{\sum_{i=1}^h n_i (\bar{x}_i - \bar{\bar{x}})^2}{h - 1}$$

Step 4: If there is a significant difference between the treatments, then s_b^2 should be significantly greater than s_w^2 , which we evaluate using a one-tailed F -test where

$$H_0 : s_b^2 = s_w^2$$

$$H_A : s_b^2 > s_w^2$$

Step 5: If there is a significant difference, then we estimate σ_{rand}^2 and $\sigma_{systematic}^2$ as

$$s_w^2 \approx \sigma_{rand}^2$$

$$s_b^2 \approx \sigma_{rand}^2 + \bar{n} \sigma_{systematic}^2$$

where $\bar{n}$ is the average number of replicates per treatment.

This seems like a lot of work, but we can simplify the calculations by noting that

$$SS_{total} = \sum_{i=1}^h \sum_{j=1}^{n_i} (x_{ij} - \bar{\bar{x}})^2 = \bar{s}^2 (N - 1)$$

$$SS_w = \sum_{i=1}^h \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)^2$$

$$SS_b = \sum_{i=1}^h n_i (\bar{x}_i - \bar{\bar{x}})^2$$

$$SS_{total} = SS_w + SS_b$$

and that SS_{total} and SS_b are relatively easy to calculate, where SS is short for sum-of-squares. Table 7.3.1 gathers these equations together

Table 7.3.1. Summary of Calculations Needed to Complete an Analysis of Variance

source of variance	sum-of-squares	degrees of freedom	variance
between samples	$\sum_{i=1}^h n_i (\bar{x}_i - \bar{\bar{x}})^2$	$h - 1$	$s_b^2 = \frac{SS_b}{h-1}$
within samples	$SS_{total} = SS_w + SS_b$	$N - h$	$s_w^2 = \frac{SS_w}{N-h}$
total	$\bar{s}^2(N - 1)$		

Example 7.3.1

Chemical reagents have a limited shelf-life. To determine the effect of light on a reagent's stability, a freshly prepared solution is stored for one hour under three different light conditions: total dark, subdued light, and full light. At the end of one hour, each solution was analyzed three times, yielding the following percent recoveries; a recovery of 100% means that the measured concentration is the same as the actual concentration. The null hypothesis is that there is no difference between the different treatments, and the alternative hypothesis is that at least one of the treatments yields a result that is significantly different than the other treatments.

trial/condition	A (total dark)	B (subdued light)	C (full light)
1	101	100	90
2	101	99	92
3	104	101	94

Solution

First, we treat the data as one large data set of nine values and calculate the global mean, $\bar{\bar{x}}$, and the global variance, $\bar{s}^2$; these are 98 and 23.75, respectively. We also calculate the mean for each of the three treatments, obtaining a value of 102.0 for treatment A, 100.0 for treatment B, and 92.0 for treatment C.

Next, we calculate the total sum-of-squares, SS_{total}

$$\bar{s}^2(N - 1) = 23.75(9 - 1) = 190.0$$

the between sample sum-of-squares, SS_b

$$SS_b = \sum_{i=1}^h n_i (\bar{x}_i - \bar{\bar{x}})^2 = 3(102.0 - 98.0)^2 + 3(100.0 - 98.0)^2 + 3(92.0 - 98.0)^2 = 168.0$$

and the within sample sum-of-squares, SS_w

$$SS_w = SS_{total} - SS_b = 190.0 - 168.0 = 22.0$$

The variance between the treatments, s_b^2 is

$$\frac{SS_b}{h - 1} = \frac{168}{3 - 1} = 84.0$$

and the variance within the treatments, s_w^2 is

$$\frac{SS_w}{N - h} = \frac{22.0}{9 - 3} = 3.67$$

Finally, we complete an F -test, calculating F_{exp}

$$F_{exp} = \frac{s_b^2}{s_w^2} = \frac{84.0}{3.67} = 22.9$$

and compare it to the critical value for $F(0.05, 2, 6) = 5.143$ from Appendix 3. Because $F_{exp} > F(0.05, 2, 6)$, we reject the null hypothesis and accept the alternative hypothesis that at least one of the treatments yields a result that is significantly different from the other treatments. We can estimate the variance due to random errors as

$$\sigma_{random}^2 = s_w^2 = 3.67$$

and the variance due to systematic errors as

$$\sigma_{systematic}^2 = \frac{\sigma_{random}^2 - s_w^2}{\bar{n}} = \frac{84.0 - 3.67}{3} = 26.8$$

Having found evidence for a significant difference between the treatments, we can use individual t -tests on pairs of treatments to show that the results for treatment C are significantly different from the other two treatments.

7.4: Non-Parametric Significance Tests

The significance tests described in Chapter 7.2 assume that we can treat the individual samples as if they are drawn from a population that is normally distributed. Although often a reasonable assumption, there are times when this is a poor assumption, such as when there is a likely outlier that we are not inclined to remove. Non-parametric significance tests allow us to compare data sets, but without making implicit assumptions about our data's distribution. In this section we will consider two non-parametric tests, the Wilcoxon signed rank test, which we can use in place of a paired t -test, and the Wilcoxon rank sum test, which we can use in place of an unpaired t -test.

Wilcoxon Signed Rank Test

When we use paired data we first calculate the difference, d_i , between each sample's paired values. We then subtract the expected difference from each d_i and then sort these adjusted differences from smallest-to-largest without considering the sign. We then assign each difference a rank (1, 2, 3, ...) and add back its sign. If two or more entries have the same absolute difference, then we average their ranks. Finally, we add together the positive ranks and add together the negative ranks. If there is no difference in the two data sets, then we expect that these two sums should be similar in value. If the smaller of the two ranks is less than a critical value, then there is reason to believe that the two data sets are significantly different from each other; see Appendix 6 for a table of critical values.

Example 7.4.1

Marecek et. al. developed a new electrochemical method for the rapid determination of the concentration of the antibiotic monensin in fermentation vats [Marecek, V.; Janchenova, H.; Brezina, M.; Betti, M. *Anal. Chim. Acta* **1991**, *244*, 15–19]. The standard method for the analysis is a test for microbiological activity, which is both difficult to complete and time-consuming. Samples were collected from the fermentation vats at various times during production and analyzed for the concentration of monensin using both methods. The results, in parts per thousand (ppt), are reported in the following table. This is the same data as in Example 7.2.6.

Sample	Microbiological	Electrochemical
1	129.5	132.3
2	89.6	91.0
3	76.6	73.6
4	52.2	58.2
5	110.8	104.2
6	50.4	49.9
7	72.4	82.1
8	141.4	154.1
9	75.0	73.4
10	34.1	38.1
11	60.3	60.1

Is there a significant difference between the methods at $\alpha = 0.05$?

Solution

Defining the difference between the methods as

$$d_i = (X_{\text{elect}})_i - (X_{\text{micro}})_i$$

we calculate the difference for each sample.

sample	1	2	3	4	5	6	7	8	9	10	11
d_i	2.8	1.4	-3.0	6.0	-6.6	-0.5	9.7	12.7	-1.6	4.0	-0.2

Next, we order the individual differences from smallest-to-largest without considering the sign

d_i	-0.2	-0.5	1.4	-1.6	2.8	-3.0	4.0	6.0	-6.6	9.7	12.7
-------	------	------	-----	------	-----	------	-----	-----	------	-----	------

We then assign each individual difference a rank, retaining the sign; thus

d_i	-1	-2	3	-4	5	-6	7	8	-9	10	11
-------	----	----	---	----	---	----	---	---	----	----	----

The sum of the negative ranks is 22 and the sum of the positive ranks is 44. The critical value for 11 samples and $\alpha = 0.05$ is 10. As the smaller of our two ranks, 22, is greater than 10, there is no evidence to suggest that there is a difference between the two methods.

Wilcoxon Rank Sum Test

The Wilcoxon rank sum test (also known as the Mann-Whitney U test) is used to compare two unpaired data sets. The values in the two data sets are sorted from smallest-to-largest, maintaining sample identity. After sorting, each value is assigned a rank (1, 2, 3, ...), again, maintaining sample identity. If two or more entries have the same absolute difference, then their ranks are averaged. Next, we add up the ranks for each sample. If there is no difference in the two data sets, then we expect that the positive and negative ranks should be similar in value. To account for differences in the size of each sample, we subtract

$$\frac{n_i(n_i + 1)}{2}$$

from each sum where n_i is the size of the sample. If the smaller of the two ranks is less than a critical value, then there is reason to believe that the two data sets are significantly different from each other; see Appendix 7 for a table of critical values.

Example 7.4.2

To compare two production lots of aspirin tablets, you collect samples from each and analyze them, obtaining the following results (in mg aspirin/tablet).

Lot 1: 256, 248, 245, 244, 248, 261

Lot 2: 241, 258, 241, 256, 254

Is there any evidence at $\alpha = 0.05$ that there is a significant difference between these two sets of results?

Solution

First, we sort the results from smallest-to-largest. To distinguish between the two samples, those from Lot 1 are shown in bold.

241, 241, **244, 245, 248, 248**, 254, **256**, 256, 258, **261**

Next we assign ranks, identifying those samples from Lot 1 by underlying them.

1.5, 1.5, 3, 4, 5.5, 5.5, 7, 8.5, 8.5, 10, 11

The sum of the ranks for Lot 1 is 37.5 and the sum of the ranks for Lot 2 is 28.5. After adjusting for the size of each sample, we have

$$37.5 - \frac{6(6+1)}{2} = 16.5$$

for Lot 1 and

$$28.5 - \frac{(5)(5+1)}{2} = 13.5$$

for Lot 2. From Appendix 7, the critical value for $\alpha = 0.05$ is 3. As the smaller of our two ranks, 13.5, is greater than 3, there is no evidence to suggest that there is a difference between the two methods.

7.5: Using R for Significance Testing and Analysis of Variance

The base installation of R has functions for most of the significance tests covered in Chapter 7.2 - Chapter 7.4.

Using R to Compare Variances

The R function for comparing variances is `var.test()` which takes the following form

```
var.test(x, y, ratio = 1, alternative = c("two.sided", "less", "greater"), conf.level = 0.95, ...)
```

where `x` and `y` are numeric vectors that contain the two samples, `ratio` is the expected ratio for the null hypothesis (which defaults to 1), `alternative` is a character string that states the alternative hypothesis (which defaults to `two.sided` or two-tailed), and a `conf.level` that gives the size of the confidence interval, which defaults to 0.95, or 95%, or $\alpha = 0.05$. We can use this function to compare the variances of two samples, s_1^2 vs s_2^2 , but not the variance of a sample and the variance for a population s^2 vs σ^2 .

Let's use R on the data from Example 7.2.3, which considers two sets of United States pennies.

```
# create vectors to store the data

sample1 = c(3.080, 3.094, 3.107, 3.056, 3.112, 3.174, 3.198)

sample2 = c(3.052, 3.141, 3.083, 3.083, 3.048)

# run two-sided variance test with alpha = 0.05 and null hypothesis that variances are equal

var.test(x = sample1, y = sample2, ratio = 1, alternative = "two.sided", conf.level = 0.95)
```

The code above yields the following output

```
F test to compare two variances

data: sample1 and sample2

F = 1.8726, num df = 6, denom df = 4, p-value = 0.5661

alternative hypothesis: true ratio of variances is not equal to 1

95 percent confidence interval:

 0.2036028 11.6609726

sample estimates: ratio of variances

 1.872598
```

Two parts of this output lead us to retain the null hypothesis of equal variances. First, the reported p -value of 0.5661 is larger than our critical value for α of 0.05, and second, the 95% confidence interval for the ratio of the variances, which runs from 0.204 to 11.7 includes the null hypothesis that it is 1.

R does not include a function for comparing s^2 to σ^2 .

Using R to Compare Means

The R function for comparing means is `t.test()` and takes the following form

```
t.test(x, y = NULL, alternative = c("two.sided", "less", "greater"), mu = 0,
      paired = FALSE, var.equal = FALSE, conf.level = 0.95, ...)
```

where `x` is a numeric vector that contains the data for one sample and `y` is an optional vector that contains data for a second sample, `alternative` is a character string that states the alternative hypothesis (which defaults to two-tailed), `mu` is either the population's expected mean or the expected difference in the means of the two samples, `paired` is a logical value that indicates whether the data is paired, `var.equal` is a logical value that indicates whether the variances for two samples are treated as equal or unequal (based on a prior `var.test()`), and `conf.level` gives the size of the confidence interval (which defaults to 0.95, or 95%, or $\alpha = 0.05$).

Using R to Compare $\bar{X}$ to μ

Let's use R on the data from Example 7.2.1, which considers the determination of the %Na₂CO₃ in a standard sample that is known to be 98.76 % w/w Na₂CO₃.

```
# create vector to store the data
na2co3 = c(98.71, 98.59, 98.62, 98.44, 98.58)

# run a two-sided t-test, using mu to define the expected mean; because the default values
# for paired and var.equal are FALSE, we can omit them here
t.test(x = na2co3, alternative = "two.sided", mu = 98.76, conf.level = 0.95)
```

The code above yields the following output

```
One Sample t-test

data: na2co3
t = -3.9522, df = 4, p-value = 0.01679
alternative hypothesis: true mean is not equal to 98.76
95 percent confidence interval:
 98.46717 98.70883
sample estimates:
mean of x
 98.588
```

Two parts of this output lead us to reject the null hypothesis of equal variances. First, the reported *p*-value of 0.01679 is less than our critical value for α of 0.05, and second, the 95% confidence interval for the experimental mean of 98.588, which runs from 98.467 to 98.709, does not include the null hypothesis that it is 98.76.

Using R to Compare Means for Two Samples

When comparing the means for two samples, we have to be careful to consider whether the data is unpaired or paired, and for unpaired data we must determine whether we can pool the variances for the two samples.

Unpaired Data

Let's use R on the data from Example 7.2.4, which considers two sets of United States pennies. This data is unpaired and, as we showed earlier, there is no evidence to suggest that the variances of the two samples are different.

```
# create vectors to store the data

sample1 = c(3.080, 3.094, 3.107, 3.056, 3.112, 3.174, 3.198)

sample2 = c(3.052, 3.141, 3.083, 3.083, 3.048)

# run a two-sided t-test, setting mu to 0 as the null hypothesis is that the means are the
# same, and setting var.equal to TRUE

t.test(x = sample1, y = sample2, alternative = "two.sided", mu = 0, var.equal = TRUE,
       conf.level = 0.95)
```

The code above yields the following output

```
Two Sample t-test

data: sample1 and sample2

t = 1.3345, df = 10, p-value = 0.2116

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

 -0.02403040 0.09580182

sample estimates:

mean of x mean of y

 3.117286 3.081400
```

Two parts of this output lead us to retain the null hypothesis of equal means. First, the reported p -value of 0.2116 is greater than our critical value for α of 0.05, and second, the 95% confidence interval for the difference in the experimental means, which runs from -0.0240 to 0.0958, includes the null hypothesis that it is 0.

Paired Data

Let's use R on the data from Example 7.2.1, which compares two methods for determining the concentration of the antibiotic monensin in fermentation vats.

```
# create vectors to store the data

microbiological = c(129.5, 89.6, 76.6, 52.2, 110.8, 50.4, 72.4, 141.4, 75.0, 34.1, 60.3)

electrochemical = c(132.3, 91.0, 73.6, 58.2, 104.2, 49.9, 82.1, 154.1, 73.4, 38.1, 60.1)

# run a two-tailed t-test, setting mu to 0 as the null hypothesis is that the means are the
# same, and setting paired to TRUE

t.test(x = microbiological, y = electrochemical, alternative = "two.sided", mu = 0, paired =
       TRUE, conf.level = 0.95)
```

The code above yields the following output

Paired t-test

data: microbiological and electrochemical

$t = -1.3225$, $df = 10$, $p\text{-value} = 0.2155$

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-6.028684 1.537775

sample estimates:

mean of the differences

-2.245455

Two parts of this output lead us to retain the null hypothesis of equal means. First, the reported p -value of 0.2155 is greater than our critical value for α of 0.05, and second, the 95% confidence interval for the difference in the experimental mean, which runs from -6.03 to 1.54, includes the null hypothesis that it is 0.

Using R to Detect Outliers

The base installation of R does not include tests for outliers, but the `outliers` package provided functions for Dixon's Q-test and Grubb's test. To install the package, use the following lines of code

```
install.packages("outliers")
```

```
library(outliers)
```

You only need to install the package once, but you must use `library()` to make the package available when you begin a new R session.

Dixon Q-test

The R function for Dixon's Q-test is `dixon.test()` and takes the following form

```
dixon.test(x, type, two.sided)
```

where `x` is a numeric vector with the data we are considering, `type` defines the specific value(s) that we are testing (we will use `type = 10`, which tests for a single outlier on either end of the ranked data), and `two.sided`, which indicates whether we use a one-tailed or two-tailed test (we will use `two.sided = FALSE` as we are interested in whether the smallest value is too small or the largest value is too large).

Let's use R on the data from Example 7.2.7, which considers the masses of a set of United States pennies.

```
penny = c(3.067, 2.514, 3.094, 3.049, 3.048, 3.109, 3.039, 3.079, 3.102)
```

```
dixon.test(x = penny, two.sided = FALSE, type = 10)
```

The code above yields the following output

Dixon test for outliers

data: penny Q = 0.88235, $p\text{-value} < 2.2\text{e-}16$

alternative hypothesis: lowest value 2.514 is an outlier

The reported p -value of less than 2.2×10^{-16} is less than our critical value for α of 0.05, which suggests that the penny with a mass of 2.514 g is drawn from a different population than the other pennies.

Grubb's Test

The R function for the Grubb's test is `grubbs.test()` and takes the following form

```
grubbs.test(x, type, two.sided)
```

where `x` is a numeric vector with the data we are considering, `type` defines the specific value(s) that we are testing (we will use `type = 10`, which tests for a single outlier on either end of the ranked data), and `two.sided`, which indicated whether we use a one-tailed or two-tailed test (we will use `two.sided = FALSE` as we are interested in whether the smallest value is too small or the largest value is too large).

Let's use R on the data from Example 7.2.7, which considers the masses of a set of United States pennies.

```
penny = c(3.067, 2.514, 3.094, 3.049, 3.048, 3.109, 3.039, 3.079, 3.102)
grubbs.test(x = penny, two.sided = FALSE, type = 10)
```

The code above yields the following output

```
Grubbs test for one outlier

data: penny
G = 2.64300, U = 0.01768, p-value = 9.69e-07
alternative hypothesis: lowest value 2.514 is an outlier
```

The reported p -value of 9.69×10^{-7} is less than our critical value for α of 0.05, which suggests that the penny with a mass of 2.514 g is drawn from a different population than the other pennies.

Using R to Complete Non-Parametric Significance Tests

The R function for completing the Wilcoxon signed rank test and the Wilcoxon rank sum test is `wilcox.test()`, which takes the following form

```
wilcox.test(x, y = NULL, alternative = c("two.sided", "less", "greater"), mu = 0,
            paired = FALSE, conf.level = 0.95, ...)
```

where `x` is a numeric vector that contains the data for one sample and `y` is an optional vector that contains data for a second sample, `alternative` is a character string that states the alternative hypothesis (which defaults to two-tailed), `mu` is either the population's expected mean or the expected difference in the means of the two samples, `paired` is a logical value that indicates whether the data is paired, and `conf.level` gives the size of the confidence interval (which defaults to 0.95, or 95%, or $\alpha = 0.05$).

Using R to Complete a Wilcoxon Signed Rank Test

Let's use R on the data from Example 7.3.1, which compares two methods for determining the concentration of the antibiotic monensin in fermentation vats.

```
# create vectors to store the data

microbiological = c(129.5, 89.6, 76.6, 52.2, 110.8, 50.4, 72.4, 141.4, 75.0, 34.1, 60.3)
electrochemical = c(132.3, 91.0, 73.6, 58.2, 104.2, 49.9, 82.1, 154.1, 73.4, 38.1, 60.1)
```

```
# run a two-tailed wilcoxon signed rank test, setting mu to 0 as the null hypothesis is that
# the means are the same and setting paired to TRUE

wilcox.test(x = microbiological, y = electrochemical, alternative = "two.sided", mu = 0, paired =
TRUE, conf.level = 0.95)
```

The code above yields the following output

```
Wilcoxon signed rank test

data: microbiological and electrochemical

V = 22, p-value = 0.3652

alternative hypothesis: true location shift is not equal to 0
```

where the value V is the smaller of the two signed ranks. The reported p -value of 0.3652 is greater than our critical value for α of 0.05, which means we do not have evidence to suggest that there is a difference between the mean values for the two methods.

Using R to Complete a Wilcoxon Rank Sum Test

Let's use R on the data from Example 7.3.2, which compares two methods for determining the amount of aspirin in tablets from two production lots.

```
# create vectors to store the data

lot1 = c(256, 248, 245, 244, 248, 261)

lot2= c(241, 258, 241, 256, 254)

# run a two-tailed wilcoxon signed rank test, setting mu to 0 as the null hypothesis is
# that the means are the same, and setting paired to TRUE

wilcox.test(x = lot1, y = lot2, alternative = "two.sided", mu = 0, paired = FALSE, conf.level
= 0.95)
```

The code above yields the following output

```
Wilcoxon rank sum test with continuity correction

data: lot1 and lot2

W = 16.5, p-value = 0.8541

alternative hypothesis: true location shift is not equal to 0

Warning message:

In wilcox.test.default(x = lot1, y = lot2, alternative = "two.sided", : cannot compute exact
p-value with ties
```

where the value W is the larger of the two ranked sums. The reported p -value of 0.8541 is greater than our critical value for α of 0.05, which means we do not have evidence to suggest that there is a difference between the mean values for the two methods. Note: we can ignore the warning message here as our calculated value for p is very large relative to an α of 0.05.

Using R to Complete an Analysis of Variance

Let's use the data in Example 7.3.1 to show how to complete an analysis of variance in R. First, we need to create individual numerical vectors for each treatment and then combine these vectors into a single numerical vector, which we will call `recovery`, that contains the results for each treatment.

```
a = c(101, 101, 104)
b = c(100, 98, 102)
c = c(90, 92, 94)
recovery = c(a, b, c)
```

We also need to create a vector of character strings that identifies the individual treatments for each element in the vector `recovery`.

```
treatment = c(rep("a", 3), rep("b", 3), rep("c", 3))
```

The R function for completing an analysis of variance is `aov()`, which takes the following form

```
aov(formula, ...)
```

where `formula` is a way of telling R to "explain this variable by using that variable." We will examine formulas in more detail in Chapter 8, but in this case the syntax is `recovery ~ treatment`, which means to model the recovery based on the treatment. In the code below, we assign the output of the `aov()` function to a variable so that we have access to the results of the analysis of variance

```
aov_output = aov(recovery ~ treatment)
```

through the `summary()` function

```
summary(aov_output)

      Df Sum Sq Mean Sq F value Pr(>F)
treatment 2 168 84.00  22.91 0.00155 **
Residuals 6  22  3.67

--- Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that what we earlier called the between variance is identified here as the variance due to the treatments, and that we earlier called the within variance is identified here as the residual variance. As we saw in Example 7.3.1, the value for F_{exp} is significantly greater than the critical value for F at $\alpha = 0.05$.

Having found evidence that there is a significant difference between the treatments, we can use R's `TukeyHSD()` function to identify the source(s) of that difference (HSD stands for Honest Significant Difference), which takes the general form

```
TukeyHSD(x, conf.level = 0.95, ...)
```

where `x` is an object that contains the results of an analysis of variance.

```
TukeyHSD(aov_output)

Tukey multiple comparisons of means
95% family-wise confidence level

Fit: aov(formula = recovery ~ treatment)
```

```
$treatment

      diff lwr upr p adj
b-a  -2  -6.797161 2.797161 0.4554965
c-a -10 -14.797161 -5.202839 0.0016720
c-b  -8 -12.797161 -3.202839 0.0052447
```

The table at the end of the output shows, for each pair of treatments, the difference in their mean values, the lower and the upper values for the confidence interval about the mean, and the value for α , which in R is listed as an adjusted p -value, for which we can reject the null hypothesis that the means are identical. In this case, we can see that the results for treatment C are significantly different from both treatments A and B.

We also can view the results of the TukeyHSD analysis visually by passing it to R's `plot()` function.

```
plot(TukeyHSD(aov_output))
```

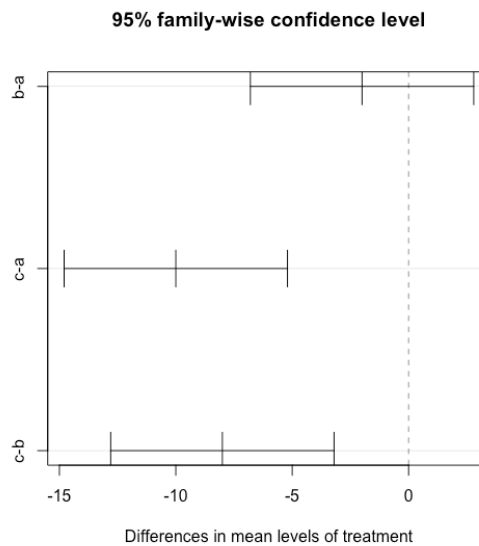


Figure 7.5.1: Plot of the TukeyHSD results. The horizontal segments show the lower boundary and the upper boundary for the confidence interval about the difference between the mean values for each pair of treatments. The vertical dashed line shows a difference of zero. Those pairs of treatments with confidence intervals that do not include a difference of zero are significantly different from each other. Here we see evidence that treatment C is significantly different from both treatments A and B, but no evidence that treatments A and B are significantly different from each other.

7.6: Exercises

1. Use this [link](#) to access a case study on data analysis and complete the last investigation in Part V: Ways to Draw Conclusions from Data.

2. Ketkar and co-workers developed an analytical method to determine trace levels of atmospheric gases. An analysis of a sample that is 40.0 parts per thousand (ppt) 2-chloroethylsulfide gave the following results from Ketkar, S. N.; Dulak, J. G.; Dheandhanou, S.; Fite, W. L. *Anal. Chim. Acta* **1991**, 245, 267-270.

43.3	34.8	31.9
37.8	34.4	31.9
42.1	33.6	35.3

Determine whether there is a significant difference between the experimental mean and the expected value at $\alpha = 0.05$.

3. To test a spectrophotometer's accuracy a solution of 60.06 ppm $K_2Cr_2O_7$ in 5.0 mM H_2SO_4 is prepared and analyzed. This solution has an expected absorbance of 0.640 at 350.0 nm in a 1.0-cm cell when using 5.0 mM H_2SO_4 as a reagent blank. Several aliquots of the solution produce the following absorbance values.

0.639	0.638	0.640	0.639	0.640	0.639	0.638
-------	-------	-------	-------	-------	-------	-------

Determine whether there is a significant difference between the experimental mean and the expected value at $\alpha = 0.01$.

4. Monna and co-workers used radioactive isotopes to date sediments from lakes and estuaries. To verify this method they analyzed a ^{208}Po standard known to have an activity of 77.5 decays/min, obtaining the following results.

77.09	75.37	72.42	76.84	77.84	76.69
78.03	74.96	77.54	76.09	81.12	75.75

Determine whether there is a significant difference between the mean and the expected value at $\alpha = 0.05$. The data in this problem are from Monna, F.; Mathieu, D.; Marques, A. N.; Lancelot, J.; Bernat, M. *Anal. Chim. Acta* **1996**, 330, 107-116.

5. A 2.6540-g sample of an iron ore, which is 53.51% w/w Fe, is dissolved in a small portion of concentrated HCl and diluted to volume in a 250-mL volumetric flask. A spectrophotometric determination of the concentration of Fe in this solution yields results of 5840, 5770, 5650, and 5660 ppm. Determine whether there is a significant difference between the experimental mean and the expected value at $\alpha = 0.05$.

6. Horvat and co-workers used atomic absorption spectroscopy to determine the concentration of Hg in coal fly ash. Of particular interest to the authors was developing an appropriate procedure for digesting samples and releasing the Hg for analysis. As part of their study they tested several reagents for digesting samples. Their results using HNO_3 and using a 1 + 3 mixture of HNO_3 and HCl are shown here. All concentrations are given as ppb Hg sample.

HNO_3 :	161	165	160	167	166	
1 + 3 HNO_3 - HCl:	159	145	140	147	143	156

Determine whether there is a significant difference between these methods at $\alpha = 0.05$. The data in this problem are from Horvat, M.; Lupsina, V.; Pihlar, B. *Anal. Chim. Acta* **1991**, 243, 71-79.

7. Lord Rayleigh, John William Strutt (1842-1919), was one of the most well known scientists of the late nineteenth and early twentieth centuries, publishing over 440 papers and receiving the Nobel Prize in 1904 for the discovery of argon. An important turning point in Rayleigh's discovery of Ar was his experimental measurements of the density of N_2 . Rayleigh approached this

experiment in two ways: first by taking atmospheric air and removing O_2 and H_2 ; and second, by chemically producing N_2 by decomposing nitrogen containing compounds (NO , N_2O , and NH_4NO_3) and again removing O_2 and H_2 . The following table shows his results for the density of N_2 , as published in *Proc. Roy. Soc.* **1894**, LV, 340 (publication 210); all values are the grams of gas at an equivalent volume, pressure, and temperature.

atmospheric origin	chemical origin
2.31017	2.30143
2.30986	2.29890
2.31010	2.29816
2.31001	2.30182
2.31024	2.29869
2.31010	2.29940
2.31028	2.29849
	2.29889

Explain why this data led Rayleigh to look for and to discover Ar. You can read more about this discovery here: Larsen, R. D. J. *Chem. Educ.* **1990**, 67, 925–928.

8. Gács and Ferraroli reported a method for monitoring the concentration of SO_2 in air. They compared their method to the standard method by analyzing urban air samples collected from a single location. Samples were collected by drawing air through a collection solution for 6 min. Shown here is a summary of their results with SO_2 concentrations reported in $\mu L/m^3$.

standard method	new method
21.62	21.54
22.20	20.51
24.27	22.31
23.54	21.30
24.25	24.62
23.09	25.72
21.02	21.54

Using an appropriate statistical test, determine whether there is any significant difference between the standard method and the new method at $\alpha = 0.05$. The data in this problem are from Gács, I.; Ferraroli, R. *Anal. Chim. Acta* **1992**, 269, 177–185.

9. One way to check the accuracy of a spectrophotometer is to measure absorbances for a series of standard dichromate solutions obtained from the National Institute of Standards and Technology. Absorbances are measured at 257 nm and compared to the accepted values. The results obtained when testing a newly purchased spectrophotometer are shown here. Determine if the tested spectrophotometer is accurate at $\alpha = 0.05$.

standard	measured absorbance	expected absorbance
1	0.2872	0.2871
2	0.5773	0.5760
3	0.8674	0.8677
4	1.1623	1.1608
5	1.4559	1.4565

10. Maskarinec and co-workers investigated the stability of volatile organics in environmental water samples. Of particular interest was establishing the proper conditions to maintain the sample's integrity between its collection and its analysis. Two preservatives were investigated—ascorbic acid and sodium bisulfate—and maximum holding times were determined for a number of volatile organics and water matrices. The following table shows results for the holding time (in days) of nine organic compounds in surface water.

compound	Ascorbic Acid	Sodium Bisulfate
methylene chloride	77	62
carbon disulfide	23	54
trichloroethane	52	51
benzene	62	42
1,1,2-trichloroethane	57	53
1,1,2,2-tetrachloroethane	33	85
tetrachloroethene	32	94
chlorobenzene	36	86

Determine whether there is a significant difference in the effectiveness of the two preservatives at $\alpha = 0.10$. The data in this problem are from Maxkarinec, M. P.; Johnson, L. H.; Holladay, S. K.; Moody, R. L.; Bayne, C. K.; Jenkins, R. A. *Environ. Sci. Technol.* **1990**, 24, 1665–1670.

11. Using X-ray diffraction, Karstang and Kvalheim reported a new method to determine the weight percent of kaolinite in complex clay minerals using X-ray diffraction. To test the method, nine samples containing known amounts of kaolinite were prepared and analyzed. The results (as % w/w kaolinite) are shown here.

actual	5.0	10.0	20.0	40.0	50.0	60.0	80.0	90.0	95.0
found	6.8	11.7	19.8	40.5	53.6	61.7	78.9	91.7	94.7

Evaluate the accuracy of the method at $\alpha = 0.05$. The data in this problem are from Karstang, T. V.; Kvalheim, O. M. *Anal. Chem.* **1991**, 63, 767–772.

12. Mizutani, Yabuki and Asai developed an electrochemical method for analyzing *l*-malate. As part of their study they analyzed a series of beverages using both their method and a standard spectrophotometric procedure based on a clinical kit purchased from Boehringer Scientific. The following table summarizes their results. All values are in ppm. The data in this problem are from Mizutani, F.; Yabuki, S.; Asai, M. *Anal. Chim. Acta* **1991**, 245, 145–150.

Sample	Electrode	Spectrophotometric
Apple Juice 1	34.0	33.4
Apple Juice 2	22.6	28.4
Apple Juice 3	29.7	29.5
Apple Juice 4	24.9	24.8
Grape Juice 1	17.8	18.3
Grape Juice 2	14.8	15.4
Mixed Fruit Juice 1	8.6	8.5
Mixed Fruit Juice 2	31.4	31.9
White Wine 1	10.8	11.5
White Wine 2	17.3	17.6
White Wine 3	15.7	15.4
White Wine 4	18.4	18.3

13. Alexiev and colleagues describe an improved photometric method for determining Fe^{3+} based on its ability to catalyze the oxidation of sulphanilic acid by KIO_4 . As part of their study, the concentration of Fe^{3+} in human serum samples was determined by the improved method and the standard method. The results, with concentrations in $\mu\text{mol/L}$, are shown in the following table.

Sample	Improved Method	Standard Method
1	8.25	8.06
2	9.75	8.84
3	9.75	8.36
4	9.75	8.73
5	10.75	13.13
6	11.25	13.65
7	13.88	13.85
8	14.25	13.43

Determine whether there is a significant difference between the two methods at $\alpha = 0.05$. The data in this problem are from Alexiev, A.; Rubino, S.; Deyanova, M.; Stoyanova, A.; Sicilia, D.; Perez Bendito, D. *Anal. Chim. Acta*, **1994**, 295, 211–219.

14. Ten laboratories were asked to determine an analyte's concentration of in three standard test samples. Following are the results, in $\mu\text{g/mL}$.

Laboratory	Sample 1	Sample 2	Sample 3
1	22.6	13.6	16.0
2	23.0	14.2	15.9
3	21.5	13.9	16.9
4	21.9	13.9	16.9
5	21.3	13.5	16.7
6	22.1	13.5	17.4
7	23.1	13.5	17.5
8	21.7	13.5	16.8
9	22.2	12.9	17.2
10	21.7	13.8	16.7

Determine if there are any potential outliers in Sample 1, Sample 2 or Sample 3. Use all three methods—Dixon's Q-test, Grubb's test, and Chauvenet's criterion—and compare the results to each other. For Dixon's Q-test and for the Grubb's test, use a significance level of $\alpha = 0.05$. The data in this problem are adapted from Steiner, E. H. "Planning and Analysis of Results of Collaborative Tests," in *Statistical Manual of the Association of Official Analytical Chemists*, Association of Official Analytical Chemists: Washington, D. C., 1975.

15. Use an appropriate non-parametric test to reanalyze the data in some or all of Exercises 7.6.2 to 7.6.14.

16. The importance of between-laboratory variability on the results of an analytical method are determined by having several laboratories analyze the same sample. In one such study, seven laboratories analyzed a sample of homogenized milk for a selected aflatoxin [data from Massart, D. L.; Vandeginste, B. G. M.; Deming, S. N.; Michotte, Y.; Kaufman, L. *Chemometrics: A Textbook*, Elsevier: Amsterdam, 1988]. The results, in ppb, are summarized below.

lab A	lab B	lab C	lab D	lab E	lab F	lab G
1.6	4.6	1.2	1.5	6.0	6.2	3.3
2.9	2.8	1.9	2.7	3.9	3.8	3.8
3.5	3.0	2.9	3.4	4.3	5.5	5.5
4.5	4.5	1.1	2.0	5.8	4.2	4.9
2.2	3.1	2.9	3.4	4.0	5.3	4.5

(a) Determine if the between-laboratory variability is significantly greater than the within-laboratory variability at $\alpha = 0.05$. If the between-laboratory variability is significant, then determine the source(s) of that variability.

(b) Estimate values for σ_{rand}^2 and for σ_{syst}^2 .

8: CALIBRATING DATA

A calibration curve is one of the most important tools in analytical chemistry as it allows us to determine the concentration of an analyte in a sample by measuring the signal it generates when placed in an instrument, such as a spectrophotometer. To determine the analyte's concentration we must know the relationship between the signal we measure, S , and the analyte's concentration, C_A , which we can write as

$$S = k_A C_A + S_{blank}$$

where k_A is the calibration curve's sensitivity and S_{blank} is the signal in the absence of analyte.

How do we find the best estimate for this relationship between the signal and the concentration of analyte? When a calibration curve is a straight-line, we represent it using the following mathematical model

$$y = \beta_0 + \beta_1 x$$

where y is the analyte's measured signal, S , and x is the analyte's known concentration, C_A , in a series of standard solutions. The constants β_0 and β_1 are, respectively, the calibration curve's expected y -intercept and its expected slope. Because of uncertainty in our measurements, the best we can do is to estimate values for β_0 and β_1 , which we represent as b_0 and b_1 . The goal of a linear regression analysis is to determine the best estimates for b_0 and b_1 .

[8.1: Unweighted Linear Regression With Errors in y](#)

[8.2: Weighted Linear Regression with Errors in y](#)

[8.3: Weighted Linear Regression With Errors in Both x and y](#)

[8.4: Curvilinear, Multivariable, and Multivariate Regression](#)

[8.5: Using R for a Linear Regression Analysis](#)

[8.6: Exercises](#)

8.1: Unweighted Linear Regression With Errors in y

The most common method for completing a linear regression makes three assumptions:

1. the difference between our experimental data and the calculated regression line is the result of indeterminate errors that affect y
2. any indeterminate errors that affect y are normally distributed
3. that indeterminate errors in y are independent of the value of x

Because we assume that the indeterminate errors are the same for all standards, each standard contributes equally in our estimate of the slope and the y -intercept. For this reason the result is considered an unweighted linear regression.

The second assumption generally is true because of the central limit theorem, which we considered in Chapter 5.3. The validity of the two remaining assumptions is less obvious and you should evaluate them before you accept the results of a linear regression. In particular the first assumption is always suspect because there certainly is some indeterminate error in the measurement of x . When we prepare a calibration curve, however, it is not unusual to find that the uncertainty in the signal, S , is significantly greater than the uncertainty in the analyte's concentration, C_A . In such circumstances the first assumption usually is reasonable.

How a Linear Regression Works

To understand the logic of a linear regression consider the example in Figure 8.1.1, which shows three data points and two possible straight-lines that might reasonably explain the data. How do we decide how well these straight-lines fit the data, and how do we determine which, if either, is the best straight-line?

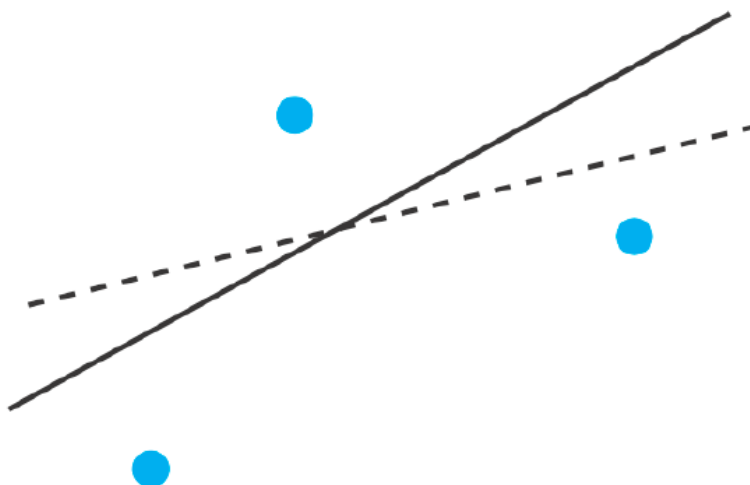


Figure 8.1.1: Illustration showing three data points and two possible straight-lines that might explain the data. The goal of a linear regression is to find the one mathematical model, in this case a straight-line, that best explains the data.

Let's focus on the solid line in Figure 8.1.1. The equation for this line is

$$\hat{y} = b_0 + b_1x$$

where b_0 and b_1 are estimates for the y -intercept and the slope, and $\hat{y}$ is the predicted value of y for any value of x . Because we assume that all uncertainty is the result of indeterminate errors in y , the difference between y and $\hat{y}$ for each value of x is the residual error, r , in our mathematical model.

$$r_i = (y_i - \hat{y}_i)$$

Figure 8.1.2 shows the residual errors for the three data points. The smaller the total residual error, R , which we define as

$$R = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

the better the fit between the straight-line and the data. In a linear regression analysis, we seek values of b_0 and b_1 that give the smallest total residual error.

Note

The reason for squaring the individual residual errors is to prevent a positive residual error from canceling out a negative residual error. You have seen this before in the equations for the sample and population standard deviations introduced in Chapter 4. You also can see from this equation why a linear regression is sometimes called the method of least squares.

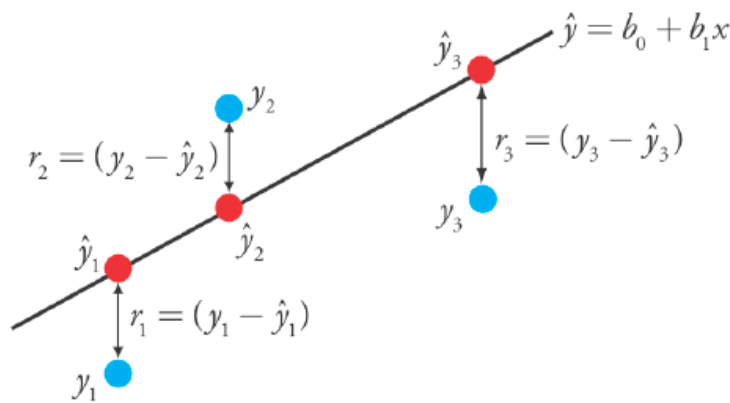


Figure 8.1.2: Illustration that shows the evaluation of a linear regression in which we assume that all uncertainty is the result of indeterminate errors in y . The points in blue, y , are the original data and the points in red, $\hat{y}_i$, are the predicted values from the regression equation, $\hat{y} = b_0 + b_1x$. The smaller the total residual error, the better the fit of the straight-line to the data.

Finding the Slope and y -Intercept for the Regression Model

Although we will not formally develop the mathematical equations for a linear regression analysis, you can find the derivations in many standard statistical texts [See, for example, Draper, N. R.; Smith, H. Applied Regression Analysis, 3rd ed.; Wiley: New York, 1998]. The resulting equation for the slope, b_1 , is

$$b_1 = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}$$

and the equation for the y -intercept, b_0 , is

$$b_0 = \frac{\sum_{i=1}^n y_i - b_1 \sum_{i=1}^n x_i}{n}$$

Although these equations appear formidable, it is necessary only to evaluate the following four summations

$$\sum_{i=1}^n x_i \quad \sum_{i=1}^n y_i \quad \sum_{i=1}^n x_i y_i \quad \sum_{i=1}^n x_i^2$$

Many calculators, spreadsheets, and other statistical software packages are capable of performing a linear regression analysis

based on this model; see Section 8.5 for details on completing a linear regression analysis using R. For illustrative purposes the necessary calculations are shown in detail in the following example.

Example 8.1.1

Using the calibration data in the following table, determine the relationship between the signal, y_i , and the analyte's concentration, x_i , using an unweighted linear regression.

Solution

We begin by setting up a table to help us organize the calculation.

x_i	y_i	$x_i y_i$	x_i^2
0.000	0.00	0.000	0.000
0.100	12.36	1.236	0.010
0.200	24.83	4.966	0.040
0.300	35.91	10.773	0.090
0.400	48.79	19.516	0.160
0.500	60.42	30.210	0.250

Adding the values in each column gives

$$\sum_{i=1}^n x_i = 1.500 \quad \sum_{i=1}^n y_i = 182.31 \quad \sum_{i=1}^n x_i y_i = 66.701 \quad \sum_{i=1}^n x_i^2 = 0.550$$

Substituting these values into the equations for the slope and the y-intercept gives

$$b_1 = \frac{(6 \times 66.701) - (1.500 \times 182.31)}{(6 \times 0.550) - (1.500)^2} = 120.706 \approx 120.71$$

$$b_0 = \frac{182.31 - (120.706 \times 1.500)}{6} = 0.209 \approx 0.21$$

The relationship between the signal, S , and the analyte's concentration, C_A , therefore, is

$$S = 120.71 \times C_A + 0.21$$

For now we keep two decimal places to match the number of decimal places in the signal. The resulting calibration curve is shown in Figure 8.1.3.

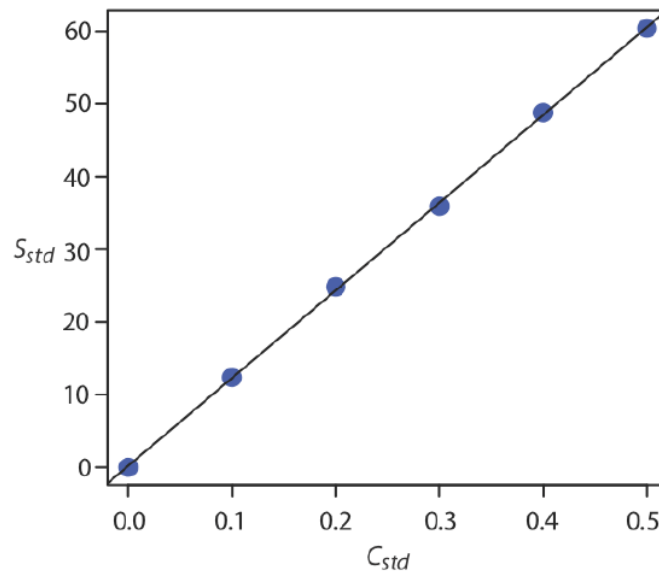


Figure 8.1.3: Calibration curve for the data in Example 8.1.1.

Uncertainty in the Regression Model

As we see in Figure 8.1.3, because of indeterminate errors in the signal, the regression line does not pass through the exact center of each data point. The cumulative deviation of our data from the regression line—the total residual error—is proportional to the uncertainty in the regression. We call this uncertainty the standard deviation about the regression, s_r , which is equal to

$$s_r = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n - 2}}$$

where y_i is the i^{th} experimental value, and $\hat{y}_i$ is the corresponding value predicted by the regression equation $\hat{y} = b_0 + b_1x$. Note that the denominator indicates that our regression analysis has $n - 2$ degrees of freedom—we lose two degree of freedom because we use two parameters, the slope and the y-intercept, to calculate $\hat{y}_i$.

A more useful representation of the uncertainty in our regression analysis is to consider the effect of indeterminate errors on the slope, b_1 , and the y-intercept, b_0 , which we express as standard deviations.

$$s_{b_1} = \sqrt{\frac{ns_r^2}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}} = \sqrt{\frac{s_r^2}{\sum_{i=1}^n (x_i - \bar{x})^2}}$$

$$s_{b_0} = \sqrt{\frac{s_r^2 \sum_{i=1}^n x_i^2}{n \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2}} = \sqrt{\frac{s_r^2 \sum_{i=1}^n x_i^2}{n \sum_{i=1}^n (x_i - \bar{x})^2}}$$

We use these standard deviations to establish confidence intervals for the expected slope, β_1 , and the expected y-intercept, β_0

$$\beta_1 = b_1 \pm ts_{b_1}$$

$$\beta_0 = b_0 \pm ts_{b_0}$$

where we select t for a significance level of α and for $n - 2$ degrees of freedom. Note that these equations do not contain the factor of $(\sqrt{n})^{-1}$ seen in the confidence intervals for μ in Chapter 6.2; this is because the confidence interval here is based on a single regression line.

Example 8.1.2

Calculate the 95% confidence intervals for the slope and y -intercept from Example 8.1.1.

Solution

We begin by calculating the standard deviation about the regression. To do this we must calculate the predicted signals, $\hat{y}_i$, using the slope and the y -intercept from Example 8.1.1, and the squares of the residual error, $(y_i - \hat{y}_i)^2$. Using the last standard as an example, we find that the predicted signal is

$$\hat{y}_6 = b_0 + b_1 x_6 = 0.209 + (120.706 \times 0.500) = 60.562$$

and that the square of the residual error is

$$(y_i - \hat{y}_i)^2 = (60.42 - 60.562)^2 = 0.2016 \approx 0.202$$

The following table displays the results for all six solutions.

x_i	y_i	$\hat{y}_i$	$(y_i - \hat{y}_i)^2$
0.000	0.00	0.209	0.0437
0.100	12.36	12.280	0.0064
0.200	24.83	24.350	0.2304
0.300	35.91	36.421	0.2611
0.400	48.79	48.491	0.0894
0.500	60.42	60.562	0.0202

Adding together the data in the last column gives the numerator in the equation for the standard deviation about the regression; thus

$$s_r = \sqrt{\frac{0.6512}{6 - 2}} = 0.4035$$

Next we calculate the standard deviations for the slope and the y -intercept. The values for the summation terms are from Example 8.1.1.

$$s_{b_1} = \sqrt{\frac{6 \times (0.4035)^2}{(6 \times 0.550) - (1.500)^2}} = 0.965$$

$$s_{b_0} = \sqrt{\frac{(0.4035)^2 \times 0.550}{(6 \times 0.550) - (1.500)^2}} = 0.292$$

Finally, the 95% confidence intervals ($\alpha = 0.05$, 4 degrees of freedom) for the slope and y -intercept are

$$\beta_1 = b_1 \pm t s_{b_1} = 120.706 \pm (2.78 \times 0.965) = 120.7 \pm 2.7$$

$$\beta_0 = b_0 \pm t s_{b_0} = 0.209 \pm (2.78 \times 0.292) = 0.2 \pm 0.80$$

where $t(0.05, 4)$ from Appendix 2 is 2.78. The standard deviation about the regression, s_r , suggests that the signal, S_{std} , is precise to one decimal place. For this reason we report the slope and the y -intercept to a single decimal place.

Using the Regression Model to Determine a Value for x Given a Value for y

Once we have our regression equation, it is easy to determine the concentration of analyte in a sample. When we use a normal calibration curve, for example, we measure the signal for our sample, S_{samp} , and calculate the analyte's concentration, C_A , using the regression equation.

$$C_A = \frac{S_{\text{samp}} - b_0}{b_1}$$

What is less obvious is how to report a confidence interval for C_A that expresses the uncertainty in our analysis. To calculate a confidence interval we need to know the standard deviation in the analyte's concentration, s_{C_A} , which is given by the following equation

$$s_{C_A} = \frac{s_r}{b_1} \sqrt{\frac{1}{m} + \frac{1}{n} + \frac{(\bar{S}_{\text{samp}} - \bar{S}_{\text{std}})^2}{(b_1)^2 \sum_{i=1}^n (C_{\text{std}_i} - \bar{C}_{\text{std}})^2}}$$

where m is the number of replicates we use to establish the sample's average signal, S_{samp} , n is the number of calibration standards, $\bar{S}_{\text{std}}$ is the average signal for the calibration standards, and C_{std_i} and $\bar{C}_{\text{std}}$ are the individual and the mean concentrations for the calibration standards. Knowing the value of s_{C_A} , the confidence interval for the analyte's concentration is

$$\mu_{C_A} = C_A \pm t s_{C_A}$$

where μ_{C_A} is the expected value of C_A in the absence of determinate errors, and with the value of t is based on the desired level of confidence and $n - 2$ degrees of freedom.

A close examination of these equations should convince you that we can decrease the uncertainty in the predicted concentration of analyte, C_A if we increase the number of standards, n , increase the number of replicate samples that we analyze, m , and if the sample's average signal, $\bar{S}_{\text{samp}}$, is equal to the average signal for the standards, $\bar{S}_{\text{std}}$. When practical, you should plan your calibration curve so that S_{samp} falls in the middle of the calibration curve. For more information about these regression equations see (a) Miller, J. N. *Analyst* **1991**, 116, 3-14; (b) Sharaf, M. A.; Illman, D. L.; Kowalski, B. R. *Chemometrics*, Wiley-Interscience: New York, 1986, pp. 126-127; (c) Analytical Methods Committee "Uncertainties in concentrations estimated from calibration experiments," *AMC Technical Brief*, March 2006.

Note

The equation for the standard deviation in the analyte's concentration is written in terms of a calibration experiment. A more general form of the equation, written in terms of x and y , is given here.

$$s_x = \frac{s_r}{b_1} \sqrt{\frac{1}{m} + \frac{1}{n} + \frac{(\bar{Y} - \bar{y})^2}{(b_1)^2 \sum_{i=1}^n (x_i - \bar{x})^2}}$$

Example 8.1.3

Three replicate analyses for a sample that contains an unknown concentration of analyte, yields values for S_{samp} of 29.32, 29.16 and 29.51 (arbitrary units). Using the results from Example 8.1.1 and Example 8.1.2, determine the analyte's concentration, C_A , and its 95% confidence interval.

Solution

The average signal, $\bar{S}_{\text{samp}}$, is 29.33, which, using the slope and the y-intercept from Example 8.1.1, gives the analyte's concentration as

$$C_A = \frac{\bar{S}_{\text{samp}} - b_0}{b_1} = \frac{29.33 - 0.209}{120.706} = 0.241$$

To calculate the standard deviation for the analyte's concentration we must determine the values for $\bar{S}_{\text{std}}$ and for $\sum_{i=1}^2 (C_{\text{std}_i} - \bar{C}_{\text{std}})^2$. The former is just the average signal for the calibration standards, which, using the data in Table 8.1.1, is 30.385. Calculating $\sum_{i=1}^2 (C_{\text{std}_i} - \bar{C}_{\text{std}})^2$ looks formidable, but we can simplify its calculation by recognizing that this sum-of-squares is the numerator in a standard deviation equation; thus,

$$\sum_{i=1}^n (C_{\text{std}_i} - \bar{C}_{\text{std}})^2 = (s_{C_{\text{std}}})^2 \times (n - 1)$$

where $s_{C_{\text{std}}}$ is the standard deviation for the concentration of analyte in the calibration standards. Using the data in Table 8.1.1 we find that $s_{C_{\text{std}}}$ is 0.1871 and

$$\sum_{i=1}^n (C_{\text{std}_i} - \bar{C}_{\text{std}})^2 = (0.1872)^2 \times (6 - 1) = 0.175$$

Substituting known values into the equation for s_{C_A} gives

$$s_{C_A} = \frac{0.4035}{120.706} \sqrt{\frac{1}{3} + \frac{1}{6} + \frac{(29.33 - 30.385)^2}{(120.706)^2 \times 0.175}} = 0.0024$$

Finally, the 95% confidence interval for 4 degrees of freedom is

$$\mu_{C_A} = C_A \pm ts_{C_A} = 0.241 \pm (2.78 \times 0.0024) = 0.241 \pm 0.007$$

Figure 8.1.4 shows the calibration curve with curves showing the 95% confidence interval for C_A .

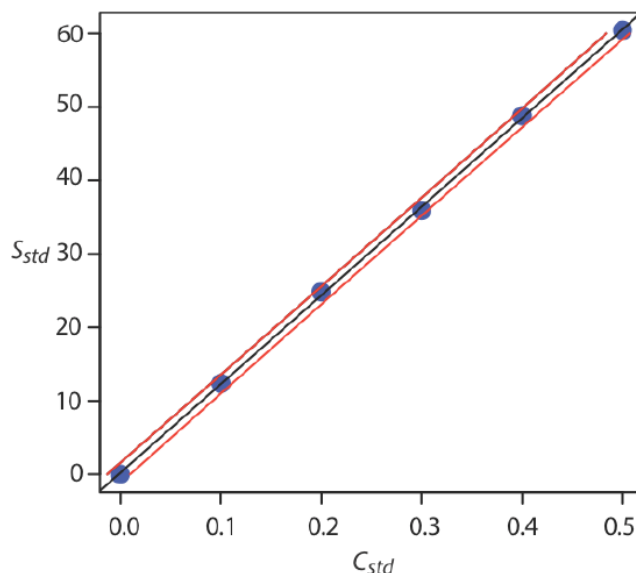


Figure 8.1.4: Example of a normal calibration curve with a superimposed confidence interval for the analyte's concentration. The points in blue are the original data from Table 8.1.1. The black line is the normal calibration curve as determined in Example 8.1.1. The red lines show the 95% confidence interval for C_A assuming a single determination of S_{samp} .

Evaluating a Regression Model

You should never accept the result of a linear regression analysis without evaluating the validity of the model. Perhaps the simplest way to evaluate a regression analysis is to examine the residual errors. As we saw earlier, the residual error for a single calibration standard, r_i , is

$$r_i = (y_i - \hat{y}_i)$$

If the regression model is valid, then the residual errors should be distributed randomly about an average residual error of zero, with no apparent trend toward either smaller or larger residual errors (Figure 8.1.5a). Trends such as those in Figure 8.1.5b and Figure 8.1.5c provide evidence that at least one of the model's assumptions is incorrect. For example, a trend toward larger residual errors at higher concentrations, Figure 8.1.5b, suggests that the indeterminate errors affecting the signal are not independent of the analyte's concentration. In Figure 8.1.5c, the residual errors are not random, which suggests we cannot model the data using a straight-line relationship. Regression methods for the latter two cases are discussed in the following sections.

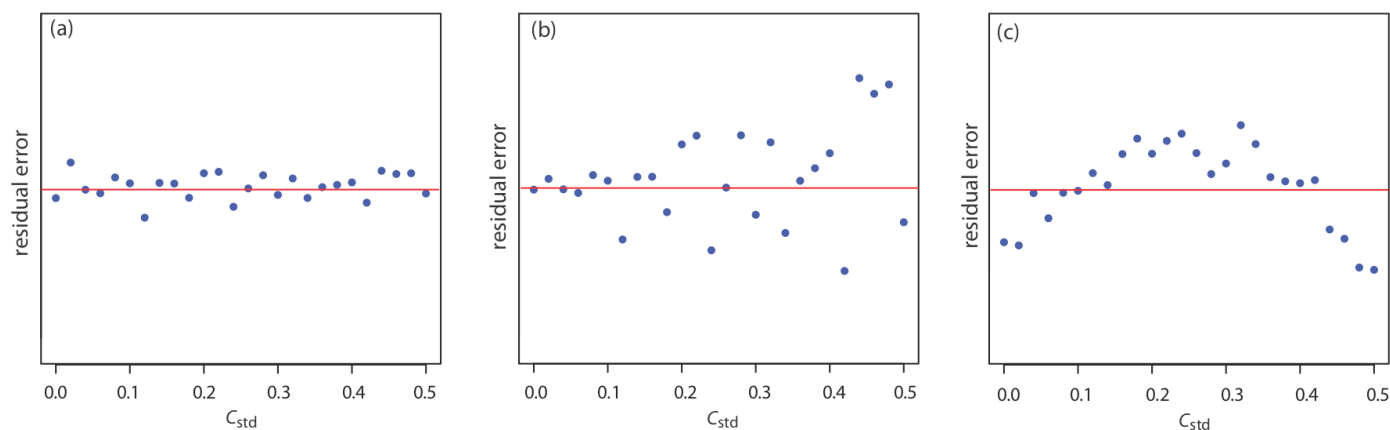


Figure 8.1.5: Plots of the residual error in the signal, S_{std} , as a function of the concentration of analyte, C_{std} , for an unweighted straight-line regression model. The red line shows a residual error of zero. The distribution of the residual errors in (a) indicates that the unweighted linear regression model is appropriate. The increase in the residual errors in (b) for higher concentrations of analyte, suggests that a weighted straight-line regression is more appropriate. For (c), the curved pattern to the residuals suggests that a straight-line model is inappropriate; linear regression using a quadratic model might produce a better fit.

Example 8.1.4

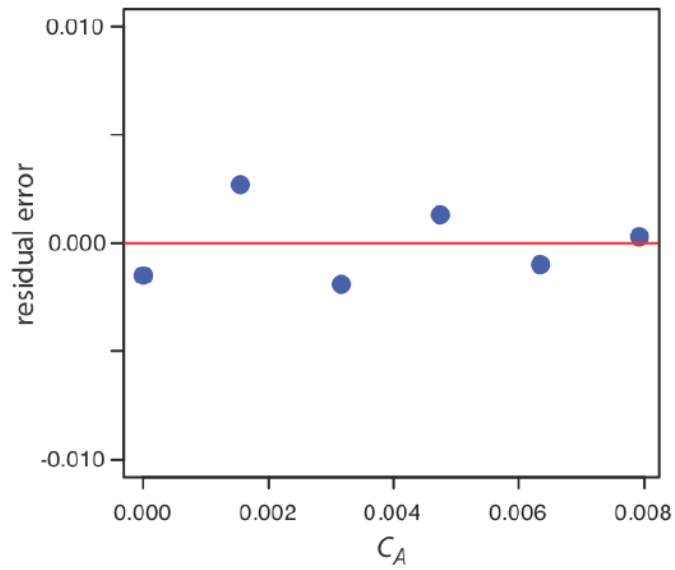
Use your results from Exercise 8.1.1 to construct a residual plot and explain its significance.

Solution

To create a residual plot, we need to calculate the residual error for each standard. The following table contains the relevant information.

x_i	y_i	$\hat{y}_i$	$y_i - \hat{y}_i$
0.000	0.000	0.0015	-0.0015
1.55×10^{-3}	0.050	0.0473	0.0027
3.16×10^{-3}	0.093	0.0949	-0.0019
4.74×10^{-3}	0.143	0.1417	0.0013
6.34×10^{-3}	0.188	0.1890	-0.0010
7.92×10^{-3}	0.236	0.2357	0.0003

The figure below shows a plot of the resulting residual errors. The residual errors appear random, although they do alternate in sign, and they do not show any significant dependence on the analyte's concentration. Taken together, these observations suggest that our regression model is appropriate.



This page titled [8.1: Unweighted Linear Regression With Errors in y](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by LibreTexts.

8.2: Weighted Linear Regression with Errors in y

Our treatment of linear regression to this point assumes that any indeterminate errors that affect y are independent of the value of x . If this assumption is false, then we must include the variance for each value of y in our determination of the y -intercept, b_0 , and the slope, b_1 ; thus

$$b_0 = \frac{\sum_{i=1}^n w_i y_i - b_1 \sum_{i=1}^n w_i x_i}{n}$$

$$b_1 = \frac{n \sum_{i=1}^n w_i x_i y_i - \sum_{i=1}^n w_i x_i \sum_{i=1}^n w_i y_i}{n \sum_{i=1}^n w_i x_i^2 - (\sum_{i=1}^n w_i x_i)^2}$$

where w_i is a weighting factor that accounts for the variance in y_i

$$w_i = \frac{n(s_{y_i})^{-2}}{\sum_{i=1}^n (s_{y_i})^{-2}}$$

and s_{y_i} is the standard deviation for y_i . In a weighted linear regression, each xy -pair's contribution to the regression line is inversely proportional to the precision of y_i ; that is, the more precise the value of y , the greater its contribution to the regression.

Example 8.2.4

Shown here are data for an external standardization in which s_{std} is the standard deviation for three replicate determination of the signal. This is the same data used in the examples in Section 8.1 with additional information about the standard deviations in the signal.

C_{std} (arbitrary units)	S_{std} (arbitrary units)	s_{std}
0.000	0.00	0.02
0.100	12.36	0.02
0.200	24.83	0.07
0.300	35.91	0.13
0.400	48.79	0.22
0.500	60.42	0.33

Determine the calibration curve's equation using a weighted linear regression. As you work through this example, remember that x corresponds to C_{std} , and that y corresponds to S_{std} .

Solution

We begin by setting up a table to aid in calculating the weighting factors.

C_{std} (arbitrary units)	S_{std} (arbitrary units)	s_{std}	$(s_{y_i})^{-2}$	w_i
0.000	0.00	0.02	2500.00	2.8339
0.100	12.36	0.02	2500.00	2.8339
0.200	24.83	0.07	204.08	0.2313
0.300	35.91	0.13	59.17	0.0671
0.400	48.79	0.22	20.66	0.0234
0.500	60.42	0.33	9.18	0.0104

Adding together the values in the fourth column gives

$$\sum_{i=1}^n (s_{y_i})^{-2}$$

which we use to calculate the individual weights in the last column. As a check on your calculations, the sum of the individual weights must equal the number of calibration standards, n . The sum of the entries in the last column is 6.0000, so all is well. After we calculate the individual weights, we use a second table to aid in calculating the four summation terms in the equations for the slope, b_1 , and the y -intercept, b_0 .

x_i	y_i	w_i	$w_i x_i$	$w_i y_i$	$w_i x_i^2$	$w_i x_i y_i$
0.000	0.00	2.8339	0.0000	0.0000	0.0000	0.0000
0.100	12.36	2.8339	0.2834	35.0270	0.0283	3.5027
0.200	24.83	0.2313	0.0463	5.7432	0.0093	1.1486
0.300	35.91	0.0671	0.0201	2.4096	0.0060	0.7229
0.400	48.79	0.0234	0.0094	1.1417	0.0037	0.4567
0.500	60.42	0.0104	0.0052	0.6284	0.0026	0.3142

Adding the values in the last four columns gives

$$\sum_{i=1}^n w_i x_i = 0.3644 \quad \sum_{i=1}^n w_i y_i = 44.9499 \quad \sum_{i=1}^n w_i x_i^2 = 0.0499 \quad \sum_{i=1}^n w_i x_i y_i = 6.1451$$

which gives the estimated slope and the estimated y -intercept as

$$b_1 = \frac{(6 \times 6.1451) - (0.3644 \times 44.9499)}{(6 \times 0.0499) - (0.3644)^2} = 122.985$$

$$b_0 = \frac{44.9499 - (122.985 \times 0.3644)}{6} = 0.0224$$

The calibration equation is

$$S_{std} = 122.98 \times C_{std} + 0.2$$

Figure 8.2.1 shows the calibration curve for the weighted regression determined here and the calibration curve for the unweighted regression in from Section 8.2. Although the two calibration curves are very similar, there are slight differences in the slope and in the y -intercept. Most notably, the y -intercept for the weighted linear regression is closer to the expected value of zero. Because the standard deviation for the signal, S_{std} , is smaller for smaller concentrations of analyte, C_{std} , a weighted linear regression gives more emphasis to these standards, allowing for a better estimate of the y -intercept.

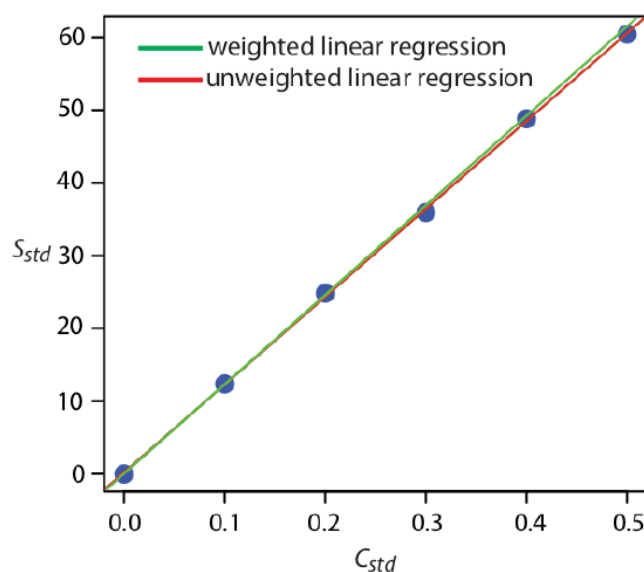


Figure 8.2.1: A comparison of the unweighted and the weighted normal calibration curves. See Example 8.2.1 for details of the unweighted linear regression and Example 8.2.4 for details of the weighted linear regression.

Equations for calculating confidence intervals for the slope, the y-intercept, and the concentration of analyte when using a weighted linear regression are not as easy to define as for an unweighted linear regression [Bonate, P. J. *Anal. Chem.* **1993**, 65, 1367-1372]. The confidence interval for the analyte's concentration, however, is at its optimum value when the analyte's signal is near the weighted centroid, y_c , of the calibration curve.

$$y_c = \frac{1}{n} \sum_{i=1}^n w_i x_i$$

8.3: Weighted Linear Regression With Errors in Both x and y

If we remove our assumption that indeterminate errors affecting a calibration curve are present only in the signal (y), then we also must factor into the regression model the indeterminate errors that affect the analyte's concentration in the calibration standards (x). The solution for the resulting regression line is computationally more involved than that for either the unweighted or weighted regression lines. Although we will not consider the details in this textbook, you should be aware that neglecting the presence of indeterminate errors in x can bias the results of a linear regression.

Note

See, for example, Analytical Methods Committee, "Fitting a linear functional relationship to data with error on both variable," [AMC Technical Brief, March, 2002](#)), as well as this chapter's Additional Resources.

This page titled [8.3: Weighted Linear Regression With Errors in Both \$x\$ and \$y\$](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by LibreTexts.

8.4: Curvilinear, Multivariable, and Multivariate Regression

A straight-line regression model, despite its apparent complexity, is the simplest functional relationship between two variables. What do we do if our calibration curve is curvilinear—that is, if it is a curved-line instead of a straight-line? One approach is to try transforming the data into a straight-line. Logarithms, exponentials, reciprocals, square roots, and trigonometric functions have been used in this way. A plot of $\log(y)$ versus x is a typical example. Such transformations are not without complications, of which the most obvious is that data with a uniform variance in y will not maintain that uniform variance after it is transformed.

Note

It is worth noting here that the term “linear” does not mean a straight-line. A linear function may contain more than one additive term, but each such term has one and only one adjustable multiplicative parameter. The function

$$y = ax + bx^2$$

is an example of a linear function because the terms x and x^2 each include a single multiplicative parameter, a and b , respectively. The function

$$y = x^b$$

is nonlinear because b is not a multiplicative parameter; it is, instead, a power. This is why you can use linear regression to fit a polynomial equation to your data.

Sometimes it is possible to transform a nonlinear function into a linear function. For example, taking the log of both sides of the nonlinear function above gives a linear function.

$$\log(y) = b \log(x)$$

Another approach to developing a linear regression model is to fit a polynomial equation to the data, such as $y = a + bx + cx^2$. You can use linear regression to calculate the parameters a , b , and c , although the equations are different than those for the linear regression of a straight-line. If you cannot fit your data using a single polynomial equation, it may be possible to fit separate polynomial equations to short segments of the calibration curve. The result is a single continuous calibration curve known as a spline function. The use of R for curvilinear regression is included in Chapter 8.5.

Note

For details about curvilinear regression, see (a) Sharaf, M. A.; Illman, D. L.; Kowalski, B. R. *Chemometrics*, Wiley-Interscience: New York, 1986; (b) Deming, S. N.; Morgan, S. L. *Experimental Design: A Chemometric Approach*, Elsevier: Amsterdam, 1987.

The regression models in this chapter apply only to functions that contain a single dependent variable and a single independent variable. One example is the simplest form of Beer's law in which the absorbance, A , of a sample at a single wavelength, λ , depends upon the concentration of a single analyte, C_A

$$A_\lambda = \epsilon_{\lambda,A} b C_A$$

where $\epsilon_{\lambda,A}$ is the analyte's molar absorptivity at the selected wavelength and b is the pathlength through the sample. In the presence of an interferent, I , however, the signal may depend on the concentrations of both the analyte and the interferent

$$A_\lambda = \epsilon_{\lambda,A} b C_A + \epsilon_{\lambda,I} b C_I$$

where $\epsilon_{\lambda,I}$ is the interferent's molar absorptivity and C_I is the interferent's concentration. This is an example of multivariable regression, which is covered in more detail in Chapter 9 when we consider the optimization of experiments where there is a single dependent variable and two or more independent variables.

Note

For more details on Beer's law, see Chapter 10 of *Analytical Chemistry* 2.1.

In multivariate regression we have both multiple dependent variables, such as the absorbance of samples at two or more wavelengths, and multiple independent variables, such as the concentrations of two or more analytes in the samples. As discussed in Chapter 0.2, we can represent this using matrix notation

$$\begin{bmatrix} \dots & \dots & \dots \\ \vdots & A & \vdots \\ \dots & \dots & \dots \end{bmatrix}_{r \times c} = \begin{bmatrix} \dots & \dots & \dots \\ \vdots & \epsilon b & \vdots \\ \dots & \dots & \dots \end{bmatrix}_{r \times n} \times \begin{bmatrix} \dots & \dots & \dots \\ \vdots & C & \vdots \\ \dots & \dots & \dots \end{bmatrix}_{n \times c}$$

where there are r wavelengths, c samples, and n analytes. Each column in the ϵb matrix, for example, holds the ϵb value for a different analyte at one of r wavelengths, and each row in the C matrix is the concentration of one of the n analytes in one of the c samples. We will consider this approach in more detail in Chapter 11.

Note

For a nice discussion of the difference between multivariable regression and multivariate regression, see Hidalgo, B.; Goodman, M. "Multivariate or Multivariable Regression," *Am. J. Public Health*, **2013**, 103, 39-40.

8.5: Using R for a Linear Regression Analysis

In Section 8.1 we used the data in the table below to work through the details of a linear regression analysis where values of x_i are the concentrations of analyte, C_A , in a series of standard solutions, and where values of y_i , are their measured signals, S . Let's use R to model this data using the equation for a straight-line.

$$y = \beta_0 + \beta_1 x$$

Table 8.5.1: Calibration Data From Worked Example in Section 8.1.

x_i	y_i
0.000	0.00
0.100	12.36
0.200	24.83
0.300	35.91
0.400	48.79
0.500	60.42

Entering Data into R

To begin, we create two objects, one that contains the concentration of the standards and one that contains their corresponding signals.

```
conc = c(0, 0.1, 0.2, 0.3, 0.4, 0.5)
```

```
signal = c(0, 12.36, 24.83, 35.91, 48.79, 60.42)
```

Creating a Linear Model in R

A linear model in R is defined using the general syntax

dependent variable ~ independent variable(s)

For example, the syntax for a model with the equation $y = \beta_0 + \beta_1 x$, where β_0 and β_1 are the model's adjustable parameters, is $y \sim x$. Table 8.5.2 provides some additional examples where A and B are independent variables, such as the concentrations of two analytes, and y is a dependent variable, such as a measured signal.

Table 8.5.2: Syntax for Selected Linear Models in R.

model	syntax	comments on model
$y = \beta_a A$	$y \sim 0 + A$	straight-line forced through (0, 0)
$y = \beta_0 + \beta_a A$	$y \sim A$	straight-line with a y-intercept
$y = \beta_0 + \beta_a A + \beta_b B$	$y \sim A + B$	first-order in A and B
$y = \beta_0 + \beta_a A + \beta_b B + \beta_{ab} AB$	$y \sim A * B$	first-order in A and B with AB interaction
$y = \beta_0 + \beta_{ab} AB$	$y \sim A : B$	AB interaction only
$y = \beta_0 + \beta_a A + \beta_{aa} A^2$	'^' allowed only in math mode	second-order polynomial

Note

The last formula in this table, $y \sim A + B$, includes the `I()`, or `AsIs` function. One complication with writing formulas is that they use symbols that have different meanings in formulas than they have in a mathematical equation. For example, take the simple formula $y \sim A + B$ that corresponds to the model $y = \beta_0 + \beta_a A + \beta_b B$. Note that the plus sign here builds a formula that has an intercept and a term for A and a term for B . But what if we wanted to build a model that used the sum of A and B as the variable. Wrapping $A + B$ inside of the `I()` function accomplishes this; thus $y \sim I(A + B)$ builds the model $y = \beta_0 + \beta_{a+b}(A + B)$.

To create our model we use the `lm()` function—where `lm` stands for linear model—assigning the results to an object so that we can access them later.

```
calcurve = lm(signal ~ conc)
```

Evaluating the Linear Regression Model

To evaluate the results of a linear regression we need to examine the data and the regression line, and to review a statistical summary of the model. To examine our data and the regression line, we use the `plot()` function, first introduced in Chapter 3, which takes the following general form

```
plot(x, y, ...)
```

where `x` and `y` are the objects that contain our data and the `...` allow for passing optional arguments to control the plot's style. To overlay the regression curve, we use the `abline()` function

```
abline(object, ...)
```

`object` is the object that contains the results of the linear regression model and the `...` allow for passing optional arguments to control the model's style. Entering the commands

```
plot(conc, signal, pch = 19, col = "blue", cex = 2)
```

```
abline(calcurve, col = "red", lty = 2, lwd = 2)
```

creates the plot shown in Figure 8.5.1.

Note

The `abline()` function works only with a straight-line model.

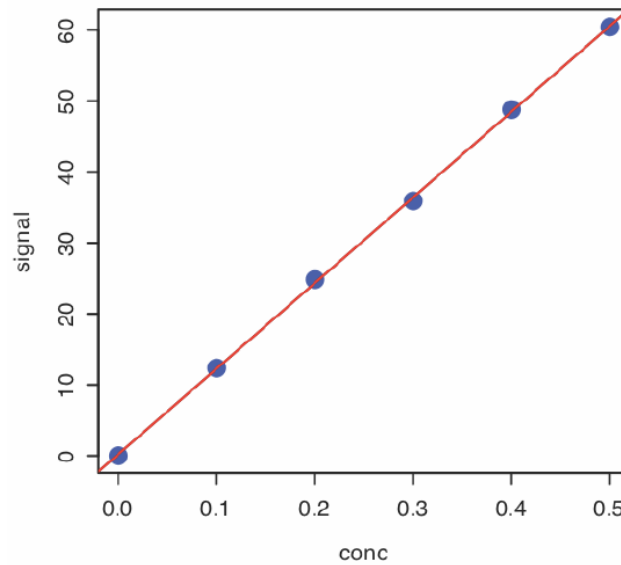


Figure 8.5.1: Example of a regression plot in R showing the data (in blue) and the regression line (in red). You can customize your plot by adjusting the plot command's optional arguments; see Chapter 3.3 for details.

To review a statistical summary of the regression model, we use the `summary()` function.

```
summary(calcurve)
```

The resulting output, which is shown below, contains three sections.

Call:

```
lm(formula = signal ~ conc)
```

Residuals:

```
1 2 3 4 5 6
-0.20857 0.08086 0.48029 -0.51029 0.29914 -0.14143
```

Coefficients:

```
Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.2086 0.2919 0.715 0.514
conc 120.7057 0.9641 125.205 2.44e-08 ***
---
```

```
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.4033 on 4 degrees of freedom
```

```
Multiple R-squared: 0.9997, Adjusted R-squared: 0.9997
```

```
F-statistic: 1.568e+04 on 1 and 4 DF, p-value: 2.441e-08
```

The first section of this summary lists the residual errors. To examine a plot of the residual errors, use the command

```
plot(calcurve, which = 1)
```

which produces the result shown in Figure 8.5.2. Note that R plots the residuals against the predicted (fitted) values of y instead of against the known values of x , as we did in Section 8.1; the choice of how to plot the residuals is not critical. The line in Figure 8.5.2 is a smoothed fit of the residuals.

Note

The reason for including the argument `which = 1` is not immediately obvious. When you use R's `plot()` function on an object created using `lm()`, the default is to create four charts that summarize the model's suitability. The first of these charts is the residual plot; thus, `which = 1` limits the output to this plot.

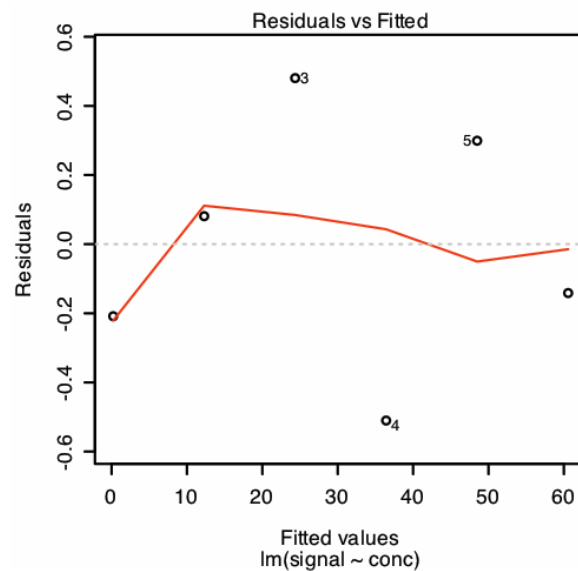


Figure 8.5.2: Example showing R's plot of a regression model's residual error.

The second section of the summary provides estimates for the model's coefficients—the slope, β_1 , and the y -intercept, β_0 —along with their respective standard deviations (*Std. Error*). The column *t value* and the column *Pr(>|t|)* are the p -values for the following t -tests.

$$\text{slope: } H_0: \beta_1 = 0 \quad H_A: \beta_1 \neq 0$$

$$\text{y-intercept: } H_0: \beta_0 = 0 \quad H_A: \beta_0 \neq 0$$

The results of these t -tests provide convincing evidence that the slope is not zero and no evidence that the y -intercept differs significantly from zero.

The last section of the summary provides the standard deviation about the regression (*residual standard error*), the square of the correlation coefficient (*multiple R-squared*), and the result of an F -test on the model's ability to explain the variation in the y values.

The value for F -statistic is the result of an F -test of the following null and alternative hypotheses.

$$H_0: \text{the regression model does not explain the variation in } y$$

$$H_A: \text{the regression model does explain the variation in } y$$

The value in the column for *Significance F* is the probability for retaining the null hypothesis. In this example, the probability is 2.5×10^{-8} , which is strong evidence for rejecting the null hypothesis and accepting the regression model. As is the case with the correlation coefficient, a small value for the probability is a likely outcome for any calibration curve, even when the model is inappropriate. The probability for retaining the null hypothesis for the data in Figure 8.5.3, for example, is 9.0×10^{-5} .

The correlation coefficient is a measure of the extent to which the regression model explains the variation in y . Values of r range

from -1 to $+1$. The closer the correlation coefficient is to $+1$ or to -1 , the better the model is at explaining the data. A correlation coefficient of 0 means there is no relationship between x and y . In developing the calculations for linear regression, we did not consider the correlation coefficient. There is a reason for this. For most straight-line calibration curves the correlation coefficient is very close to $+1$, typically 0.99 or better. There is a tendency, however, to put too much faith in the correlation coefficient's significance, and to assume that an r greater than 0.99 means the linear regression model is appropriate. Figure 8.5.3 provides a useful counterexample. Although the regression line has a correlation coefficient of 0.993 , the data clearly is curvilinear. The take-home lesson is simple: do not fall in love with the correlation coefficient!

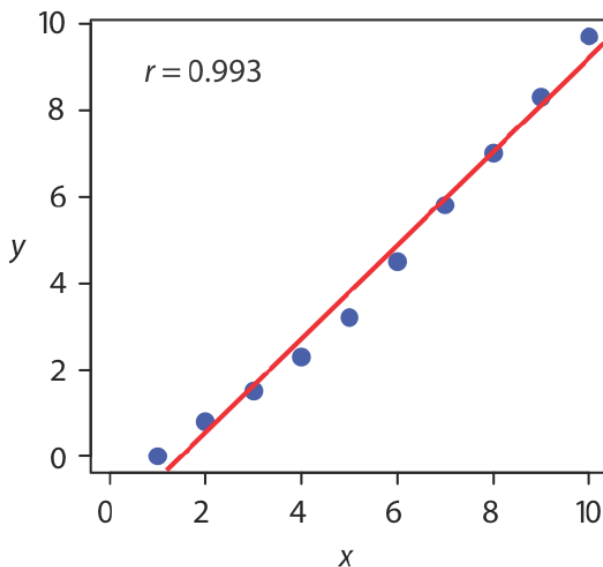


Figure 8.5.3: Example of fitting a straight-line (in red) to curvilinear data (in blue).

Predicting the Uncertainty in x Given y

Although R's base installation does not include a command for predicting the uncertainty in the independent variable, x , given a measured value for the dependent variable, y , the **chemCal** package does. To use this package you need to install it by entering the following command.

```
install.packages("chemCal")
```

Once installed, which you need to do just once, you can access the package's functions by using the **library()** command.

```
library(chemCal)
```

The command for predicting the uncertainty in C_A is **inverse.predict()** and takes the following form for an unweighted linear regression

```
inverse.predict(object, newdata, alpha = value)
```

where **object** is the object that contains the regression model's results, **new-data** is an object that contains one or more replicate values for the dependent variable and **value** is the numerical value for the significance level. Let's use this command to complete the calibration curve example from Section 8.1 in which we determined the concentration of analyte in a sample using three replicate analyses. First, we create an object that contains the replicate measurements of the signal

```
rep_signal = c(29.32, 29.16, 29.51)
```

and then we complete the computation using the following command

```
inverse.predict(calcurve, rep_signal, alpha = 0.05)
```

which yields the results shown here

```
$Prediction  
[1] 0.2412597  
$`Standard Error`  
[1] 0.002363588  
$Confidence  
[1] 0.006562373  
$`Confidence Limits`  
[1] 0.2346974 0.2478221
```

The analyte's concentration, C_A , is given by the value `$Prediction`, and its standard deviation, s_{C_A} , is shown as `$`Standard Error``. The value for `$Confidence` is the confidence interval,  $\pm ts_{C_A}$ , for the analyte's concentration, and `$`Confidence Limits`` provides the lower limit and upper limit for the confidence interval for C_A .

Using R for a Weighted Linear Regression

R's command for an unweighted linear regression also allows for a weighted linear regression if we include an additional argument, `weights`, whose value is an object that contains the weights.

```
lm(y ~ x, weights = object)
```

Let's use this command to complete the weighted linear regression example in Section 8.2 First, we need to create an object that contains the weights, which in R are the reciprocals of the standard deviations in y , $(s_{y_i})^{-2}$. Using the data from the earlier example, we enter

```
syi = c(0.02, 0.02, 0.07, 0.13, 0.22, 0.33)  
w = 1/syi^2
```

to create the object, `w`, that contains the weights. The commands

```
weighted_calcurve = lm(signal ~ conc, weights = w)  
summary(weighted_calcurve)
```

generate the following output.

Call:

```
lm(formula = signal ~ conc, weights = w)
```

Weighted Residuals:

```
1 2 3 4 5 6  
-2.223 2.571 3.676 -7.129 -1.413 -2.864
```

Coefficients:

```
Estimate Std. Error t value Pr(>|t|)  
(Intercept) 0.04446 0.08542 0.52 0.63  
conc 122.64111 0.93590 131.04 2.03e-08 ***
```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.639 on 4 degrees of freedom

Multiple R-squared: 0.9998, Adjusted R-squared: 0.9997

F-statistic: 1.717e+04 on 1 and 4 DF, p-value: 2.034e-08

Any difference between the results shown here and the results in Section 8.2 are the result of round-off errors in our earlier calculations.

Note

You may have noticed that this way of defining weights is different than that shown in Section 8.2 In deriving equations for a weighted linear regression, you can choose to normalize the sum of the weights to equal the number of points, or you can choose not to—the algorithm in R does not normalize the weights.

Using R for a Curvilinear Regression

As we see in this example, we can use R to model data that is not in the form of a straight-line by simply adjusting the linear model.

Example 8.5.1

Use the data below to explore two models for the data in the table below, one using a straight-line, $y = \beta_0 + \beta_1 x$, and one that is a second-order polynomial, $y = \beta_0 + \beta_1 x + \beta_2 x^2$.

x_i	y_i
0.00	0.00
1.00	0.94
2.00	2.15
3.00	3.19
4.00	3.70
5.00	4.21

Solution

First, we create objects to store our data.

```
x = c(0, 1.00, 2.00, 3.00, 4.00, 5.00)
y = c(0, 0.94, 2.15, 3.19, 3.70, 4.21)
```

Next, we build our linear models for a straight-line and for a curvilinear fit to the data

```
straight_line = lm(y ~ x)
curvilinear = lm(y ~ x + I(x^2))
```

and plot the data and both linear models on the same plot. Because `abline()` only works for a straight-line, we use our curvilinear model to calculate sufficient values for x and y that we can use to plot the curvilinear model. Note that the coefficients for this model are stored in `curvilinear$coefficients` with the first value being β_0 , the second value being β_1 , and the third value being β_2 .

```
plot(x, y, pch = 19, col = "blue", ylim = c(0,5), xlab = "x", ylab = "y")
abline(straight_line, lwd = 2, col = "blue", lty = 2)

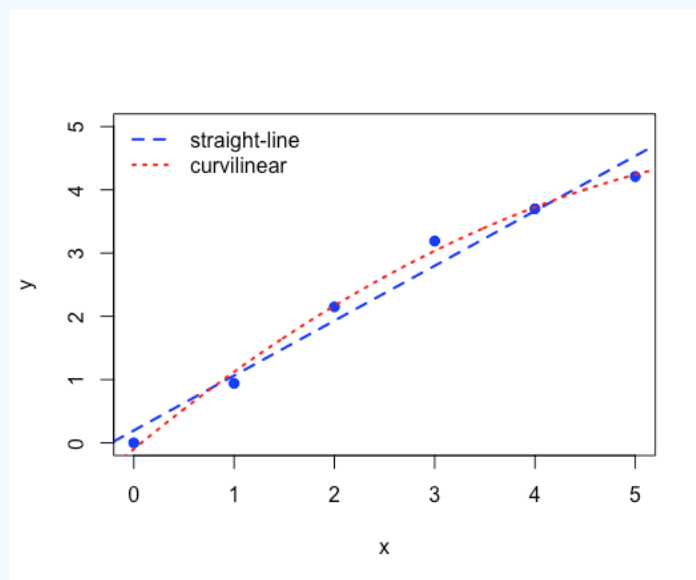
x_seq = seq(-0.5, 5.5, 0.01)

y_seq = curvilinear$coefficients[1] + curvilinear$coefficients[2] * x_seq +
curvilinear$coefficients[3] * x_seq^2

lines(x_seq, y_seq, lwd = 2, col = "red", lty = 3)

legend(x = "topleft", legend = c("straight-line", "curvilinear"), col = c("blue", "red"),
lty = c(2, 3), lwd = 2, bty = "n")
```

The resulting plot is shown here.



8.6: Exercises

1. The following data are for a series of external standards of Cd^{2+} buffered to a pH of 4.6.

$[\text{Cd}^{2+}]$ (nM)	15.4	30.4	44.9	59.0	72.7	86.0
S_{spike} (nA)	4.8	11.4	18.2	26.6	32.3	37.7

(a) Use a linear regression analysis to determine the equation for the calibration curve and report confidence intervals for the slope and the y -intercept.

(b) Construct a plot of the residuals and comment on their significance.

At a pH of 3.7 the following data were recorded for the same set of external standards.

$[\text{Cd}^{2+}]$ (nM)	15.4	30.4	44.9	59.0	72.7	86.0
S_{spike} (nA)	15.0	42.7	58.5	77.0	101	118

(c) How much more or less sensitive is this method at the lower pH?

(d) A single sample is buffered to a pH of 3.7 and analyzed for cadmium, yielding a signal of 66.3 nA. Report the concentration of Cd^{2+} in the sample and its 95% confidence interval.

The data in this problem are from Wojciechowski, M.; Balcerzak, J. *Anal. Chim. Acta* **1991**, 249, 433–445.

2. Consider the following three data sets, each of which gives values of y for the same values of x .

x	y_1	y_2	y_3
10.00	8.04	9.14	7.46
8.00	6.95	8.14	6.77
13.00	7.58	8.74	12.74
9.00	8.81	8.77	7.11
11.00	8.33	9.26	7.81
14.00	9.96	8.10	8.84
6.00	7.24	6.13	6.08
4.00	4.26	3.10	5.39
12.00	10.84	9.13	8.15
7.00	4.82	7.26	6.42
5.00	5.68	4.74	5.73

(a) An unweighted linear regression analysis for the three data sets gives nearly identical results. To three significant figures, each data set has a slope of 0.500 and a y -intercept of 3.00. The standard deviations in the slope and the y -intercept are 0.118 and 1.125 for each data set. All three standard deviations about the regression are 1.24. Based on these results for a linear regression analysis, comment on the similarity of the data sets.

(b) Complete a linear regression analysis for each data set and verify that the results from part (a) are correct. Construct a residual plot for each data set. Do these plots change your conclusion from part (a)? Explain.

- (c) Plot each data set along with the regression line and comment on your results.
- (d) Data set 3 appears to contain an outlier. Remove the apparent outlier and reanalyze the data using a linear regression. Comment on your result.
- (e) Briefly comment on the importance of visually examining your data.

These three data sets are taken from Anscombe, F. J. "Graphs in Statistical Analysis," *Amer. Statis.* **1973**, 27, 17-21.

3. Franke and co-workers evaluated a standard additions method for a voltammetric determination of Tl. A summary of their results is tabulated in the following table.

ppm Tl added	Instrument Response (μA)						
0.000	2.53	2.50	2.70	2.63	2.70	2.80	2.52
0.387	8.42	7.96	8.54	8.18	7.70	8.34	7.98
1.851	29.65	28.70	29.05	28.30	29.20	29.95	28.95
5.734	84.8	85.6	86.0	85.2	84.2	86.4	87.8

Use a weighted linear regression to determine the standardization relationship for this data. The data in this problem are from Franke, J. P.; de Zeeuw, R. A.; Hakkert, R. *Anal. Chem.* **1978**, 50, 1374-1380.

9: OPTIMIZING DATA

In the presence of H_2O_2 and H_2SO_4 , a solution of vanadium forms a reddish brown color that is believed to be a compound with the general formula $(\text{VO})_2(\text{SO}_4)_3$. The intensity of the solution's color depends on the concentration of vanadium, which means we can use its absorbance at a wavelength of 450 nm to develop a quantitative method for vanadium. The intensity of the solution's color also depends on the amounts of H_2O_2 and H_2SO_4 that we add to the sample—in particular, a large excess of H_2O_2 decreases the solution's absorbance as it changes from a reddish brown color to a yellowish color [Vogel's *Textbook of Quantitative Inorganic Analysis*, Longman: London, 1978, p. 752.]. Developing a standard method for vanadium based on this reaction requires that we optimize the amount of H_2O_2 and H_2SO_4 added if we want to maximize the absorbance at 450 nm. Using the terminology of statisticians, we call the solution's absorbance the system's response. Hydrogen peroxide and sulfuric acid are factors whose concentrations, or factor levels, determine the system's response. To optimize the method we need to find the best combination of factor levels. Usually we seek a maximum response, as is the case for the quantitative analysis of vanadium as $(\text{VO})_2(\text{SO}_4)_3$. In other situations, such as minimizing an analysis's percent error, we seek a minimum response. How we design experiments to optimize the response is the subject of this chapter.

9.1: Response Surfaces

9.2: Searching Algorithms

9.3: One-Factor-at-a-Time Optimizations

9.4: Simplex Optimization

9.5: Mathematical Models of Response Surfaces

9.6: Using R to Model a Response Surface (Multiple Regression)

9.7: Exercises

9.1: Response Surfaces

One of the most effective ways to think about an optimization is to visualize how a system's response changes when we increase or decrease the levels of one or more of its factors. We call a plot of the system's response as a function of the factor levels a response surface. The simplest response surface has one factor and is drawn in two dimensions by placing the responses on the y-axis and the factor's levels on the x-axis. The calibration curve in Figure 9.1.1 is an example of a one-factor response surface. We also can define the response surface mathematically. The response surface in Figure 9.1.1, for example, is

$$A = 0.008 + 0.0896C_A$$

where A is the absorbance and C_A is the analyte's concentration in ppm.

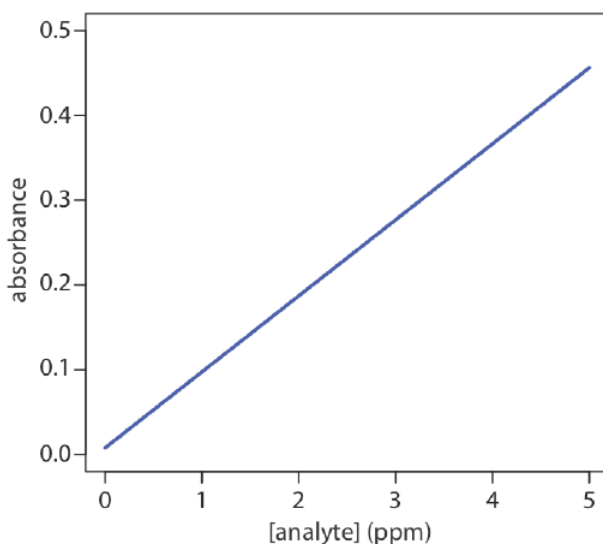


Figure 9.1.1. A calibration curve is an example of a one-factor response surface. The responses (absorbance) are plotted on the y-axis and the factor levels (concentration of analyte) are plotted on the x-axis.

For a two-factor system, such as the quantitative analysis for vanadium described earlier, the response surface is a flat or curved plane in three dimensions. As shown in Figure 9.1.2, we place the response on the z-axis and the factor levels on the x-axis and the y-axis. Figure 9.1.2a shows a pseudo-three dimensional wireframe plot for a system that obeys the equation

$$R = 3.0 - 0.30A + 0.020AB$$

where R is the response, and A and B are the factors. We also can represent a two-factor response surface using the two-dimensional level plot in Figure 9.1.2b, which uses a color gradient to show the response on a two-dimensional grid, or using the two-dimensional contour plot in Figure 9.1.2c, which uses contour lines to display the response surface.

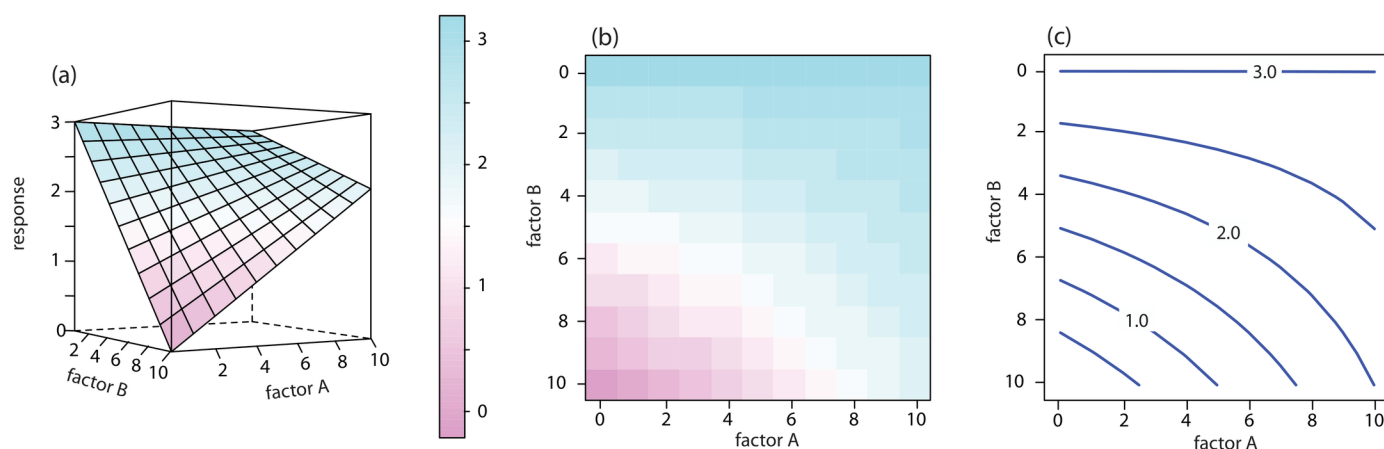


Figure 9.1.2. Three examples of a two-factor response surface displayed as (a) a pseudo-three-dimensional wireframe plot, (b) a two-dimensional level plot, and (c) a two-dimensional contour plot. We call the display in (a) a pseudo-three dimensional response surface because we show the presence of three dimensions on the page's flat, two-dimensional surface.

The response surfaces in Figure 9.1.2 cover a limited range of factor levels ($0 \leq A \leq 10$, $0 \leq B \leq 10$), but we can extend each to more positive or to more negative values because there are no constraints on the factors. Most response surfaces of interest to an analytical chemist have natural constraints imposed by the factors, or have practical limits set by the analyst. The response surface in Figure 9.1.1, for example, has a natural constraint on its factor because the analyte's concentration cannot be less than zero; that is, $C_A \geq 0$.

If we have an equation for the response surface, then it is relatively easy to find the optimum response. Unfortunately, we rarely know any useful details about the response surface. Instead, we must determine the response surface's shape and locate its optimum response by running appropriate experiments. The focus of this chapter is on useful experimental methods for characterizing a response surface. These experimental methods are divided into two broad categories: searching methods, in which an algorithm guides a systematic search for the optimum response, and modeling methods, in which we use a theoretical model or an empirical model of the response surface to predict the optimum response.

9.2: Searching Algorithms

Figure 9.2.1 shows a portion of the South Dakota Badlands, a barren landscape that includes many narrow ridges formed through erosion. Suppose you wish to climb to the highest point on this ridge. Because the shortest path to the summit is not obvious, you might adopt the following simple rule: look around you and take one step in the direction that has the greatest change in elevation, and then repeat until no further step is possible. The route you follow is the result of a systematic search that uses a searching algorithm. Of course there are as many possible routes as there are starting points, three examples of which are shown in Figure 9.2.1. Note that some routes do not reach the highest point—what we call the global optimum. Instead, many routes end at a local optimum from which further movement is impossible.

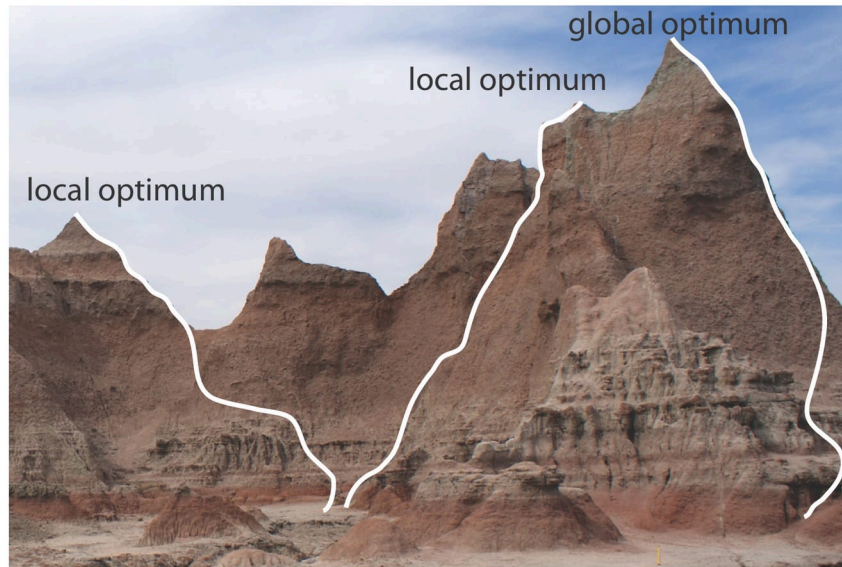


Figure 9.2.1. Finding the highest point on a ridge using a searching algorithm is one useful method for locating the optimum on a response surface. The path on the far right reaches the highest point, or the global optimum. The other two paths reach local optima. Searching algorithms have names: the one described here is the method of steepest ascent.

We can use a systematic searching algorithm to locate the optimum response. We begin by selecting an initial set of factor levels and measure the response. Next, we apply the rules of our searching algorithm to determine a new set of factor levels and measure its response, continuing this process until we reach an optimum response. Before we consider two common searching algorithms, let's consider how we evaluate a searching algorithm.

Effectiveness and Efficiency

A searching algorithm is characterized by its effectiveness and its efficiency. To be effective, a searching algorithm must find the response surface's global optimum, or at least reach a point near the global optimum. A searching algorithm may fail to find the global optimum for several reasons, including a poorly designed algorithm, uncertainty in measuring the response, and the presence of local optima. Let's consider each of these potential problems.

A poorly designed algorithm may prematurely end the search before it reaches the response surface's global optimum. As shown in Figure 9.2.2, when you climb a ridge that slopes up to the northeast, an algorithm is likely to fail if it limits your steps only to the north, south, east, or west. An algorithm that cannot respond to a change in the direction of steepest ascent is not an effective algorithm.

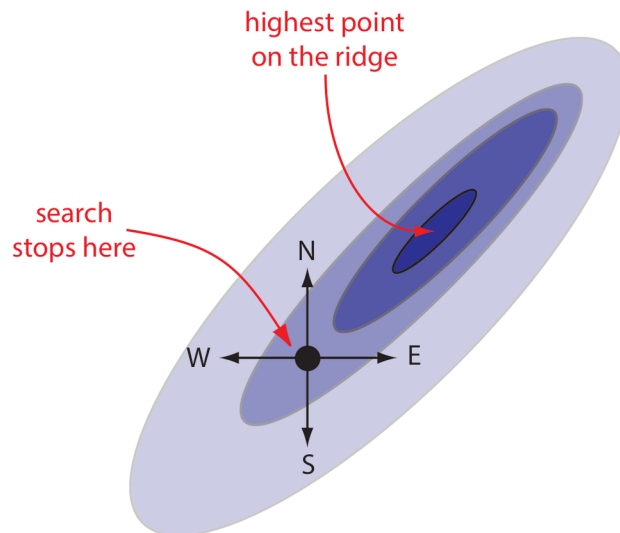


Figure 9.2.2. Example that shows how a poorly designed searching algorithm—limited to moving only north, south, east, or west—can fail to find a response surface's global optimum.

All measurements contain uncertainty, or noise, that affects our ability to characterize the underlying signal. When the noise is greater than the local change in the signal, then a searching algorithm is likely to end before it reaches the global optimum. Figure 9.2.3, which provides a different view of Figure 9.2.1, shows us that the relatively flat terrain leading up to the ridge is heavily weathered and very uneven. Because the variation in local height (the noise) exceeds the slope (the signal), our searching algorithm ends the first time we step up onto a less weathered local surface that is higher than the immediately surrounding surfaces.



Figure 9.2.3. Another view of the ridge in Figure 9.2.1 that shows the weathered terrain leading up to the ridge. The yellow rod at the bottom of the figure, which marks the trail, is about 18 in high.

Finally, a response surface may contain several local optima, only one of which is the global optimum. If we begin the search near a local optimum, our searching algorithm may never reach the global optimum. The ridge in Figure 9.2.1, for example, has many peaks. Only those searches that begin at the far right will reach the highest point on the ridge. Ideally, a searching algorithm should reach the global optimum regardless of where it starts.

A searching algorithm always reaches an optimum. Our problem, of course, is that we do not know if it is the global optimum. One method for evaluating a searching algorithm's effectiveness is to use several sets of initial factor levels, find the optimum response for each, and compare the results. If we arrive at or near the same optimum response after starting from very different

locations on the response surface, then we are more confident that is it the global optimum.

Efficiency is a searching algorithm's second desirable characteristic. An efficient algorithm moves from the initial set of factor levels to the optimum response in as few steps as possible. In seeking the highest point on the ridge in Figure 9.2.3, we can increase the rate at which we approach the optimum by taking larger steps. If the step size is too large, however, the difference between the experimental optimum and the true optimum may be unacceptably large. One solution is to adjust the step size during the search, using larger steps at the beginning and smaller steps as we approach the global optimum.

9.3: One-Factor-at-a-Time Optimizations

A simple algorithm for optimizing the quantitative method for vanadium described earlier is to select initial concentrations for H_2O_2 and H_2SO_4 and measure the absorbance. Next, we optimize one reagent by increasing or decreasing its concentration—holding constant the second reagent's concentration—until the absorbance decreases. We then vary the concentration of the second reagent—maintaining the first reagent's optimum concentration—until we no longer see an increase in the absorbance. We can stop this process, which we call a one-factor-at-a-time optimization, after one cycle or repeat the steps until the absorbance reaches a maximum value or it exceeds an acceptable threshold value.

A one-factor-at-a-time optimization is consistent with a notion that to determine the influence of one factor we must hold constant all other factors. This is an effective, although not necessarily an efficient experimental design when the factors are independent [see Sharaf, M. A.; Illman, D. L.; Kowalski, B. R. *Chemometrics*, Wiley-Interscience: New York, 1986]. Two factors are independent when a change in the level of one factor does not influence the effect of a change in the other factor's level. Table 9.3.1 provides an example of two independent factors.

Table 9.3.1. Example of Two Independent Factors

factor A	factor B	response
A_1	B_1	40
A_2	B_1	80
A_1	B_2	60
A_2	B_2	100

If we hold factor B at level B_1 , changing factor A from level A_1 to level A_2 increases the response from 40 to 80, or a change in response, ΔR of

$$R = 80 - 40 = 40$$

If we hold factor B at level B_2 , we find that we have the same change in response when the level of factor A changes from A_1 to A_2 .

$$R = 100 - 60 = 40$$

We can see this independence visually if we plot the response as a function of factor A 's level, as shown in Figure 9.3.1. The parallel lines show that the level of factor B does not influence factor A 's effect on the response.

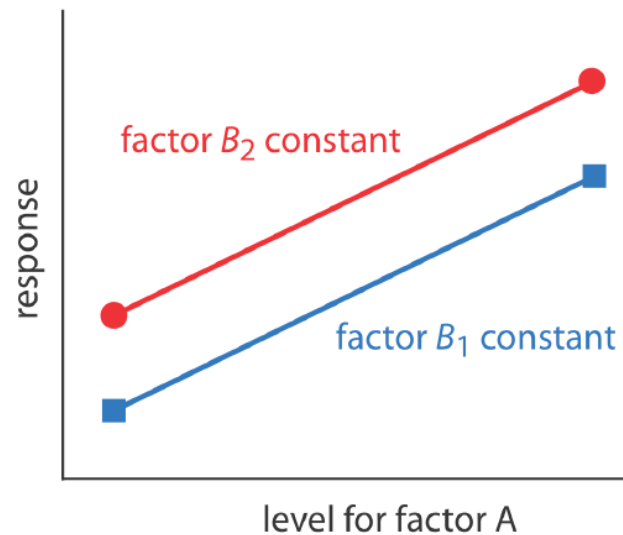


Figure 9.3.1. Factor effect plot for two independent factors. Note that the two lines are parallel, indicating that the level for factor B does not influence how factor A 's level affects the response.

Mathematically, two factors are independent if they do not appear in the same term in the equation that describes the response surface. Figure 9.3.2, for example, shows the resulting pseudo-three-dimensional surface and a contour map for the equation

$$R = 2.0 + 0.12A + 0.48B - 0.03A^2 - 0.03B^2$$

which describes a response surface with independent factors because no term in the equation includes both factor A and factor B .

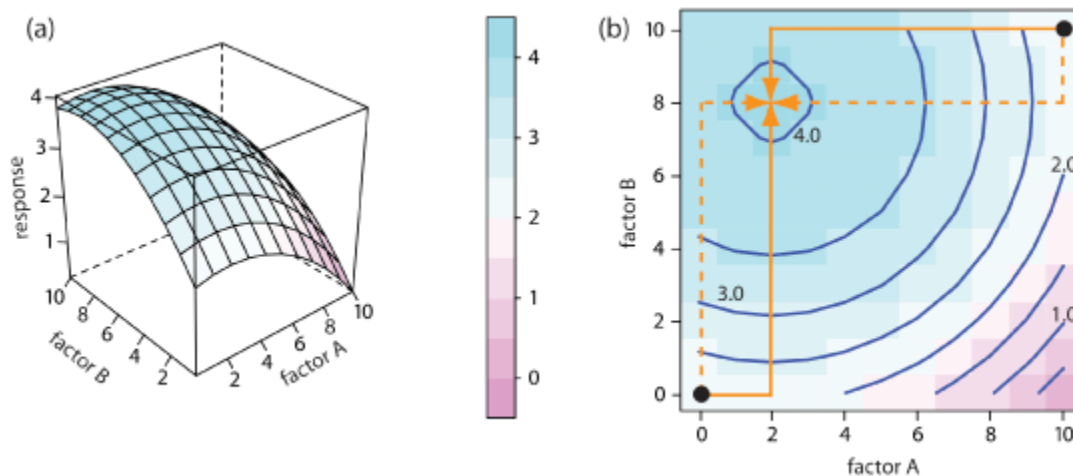


Figure 9.3.2. The response surface for two independent factors based on the equation $R = 2.0 + 0.12A + 0.48B - 0.03A^2 - 0.03B^2$ and displayed as (a) a wireframe, and as (b) an overlaid contour plot and level plot. The orange lines in (b) show the progress of one-factor-at-a-time optimizations beginning from two starting points (•) and optimizing factor A first (solid line) or factor B first (dashed line). All four trials reach the optimum response of (2, 8) in a single cycle.

The easiest way to follow the progress of a searching algorithm is to map its path on a contour plot of the response surface. Positions on the response surface are identified as (a, b) where a and b are the levels for factor A and for factor B . The contour plot in Figure 9.3.2b, for example, shows four one-factor-at-a-time optimizations of the response surface in Figure 9.3.2a. The effectiveness and efficiency of this algorithm when optimizing independent factors is clear—each trial reaches the optimum response at (2, 8) in a single cycle.

Unfortunately, factors often are not independent. Consider, for example, the data in Table 9.3.2

Table 9.3.2. Example of Two Dependent Factors

factor A	factor B	response
A_1	B_1	20
A_2	B_1	80
A_1	B_2	60
A_2	B_2	80

where a change in the level of factor B from level B_1 to level B_2 has a significant effect on the response when factor A is at level A_1

$$R = 60 - 20 = 40$$

but no effect when factor A is at level A_2 .

$$R = 80 - 80 = 0$$

Figure 9.3.3 shows this dependent relationship between the two factors.

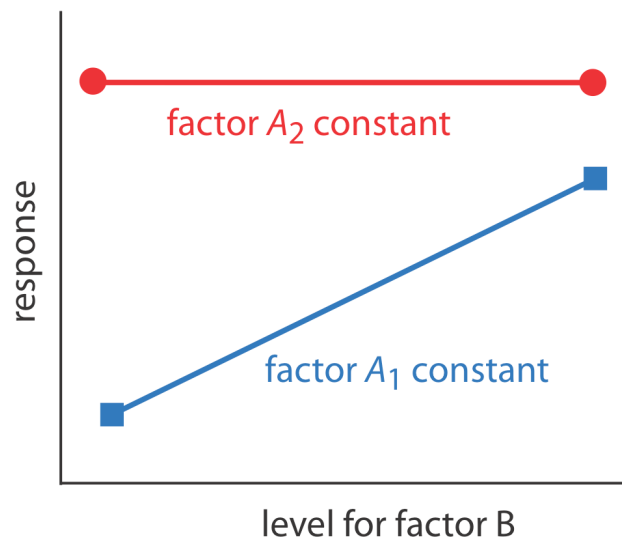


Figure 9.3.3. Factor effect plot for two dependent factors. Note that the two lines are not parallel, indicating that the level for factor A influences how factor B 's level affects the response.

Factors that are dependent are said to interact and the equation for the response surface includes an interaction term that contains both factor A and factor B . The final term in this equation

$$R = 5.5 + 1.5A + 0.6B - 0.15A^2 - 0.245B^2 - 0.0857AB$$

for example, accounts for the interaction between factor A and factor B . Figure 9.3.4 shows the resulting pseudo-three-dimensional surface and a contour map for the response surface defined by this equation. The progress of a one-factor-at-a-time optimization for this response surface is shown in Figure 9.3.4b. Although the optimization for dependent factors is effective, it is less efficient than that for independent factors. In this case it takes four cycles to reach the optimum response of (3, 7) if we begin at (0, 0).

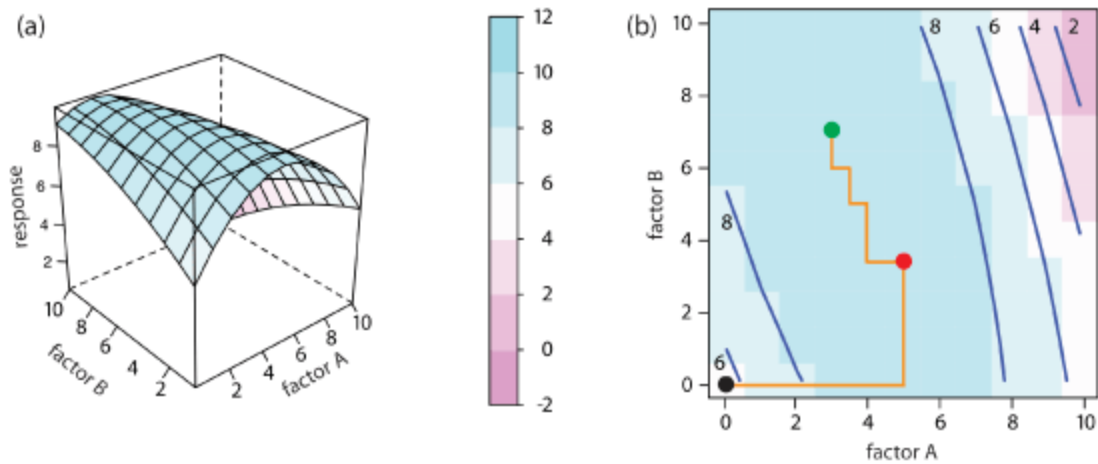


Figure 9.3.4. The response surface for two dependent factors based on equation $R = 5.5 + 1.5A + 0.6B - 0.15A^2 - 0.245B^2 - 0.0857AB$ and displayed as (a) a wireframe, and as (b) an overlaid contour plot and level plot. The orange lines in (b) show the progress of one-factor-at-a-time optimization beginning from the starting point (•) and optimizing factor A first. The red dot (•) marks the end of the first cycle. It takes four cycles to reach the optimum response of (3, 7) as shown by the green dot (•).

9.4: Simplex Optimization

One strategy for improving the efficiency of a searching algorithm is to change more than one factor at a time. A convenient way to accomplish this when there are two factors is to begin with three sets of initial factor levels arranged as the vertices of a triangle (Figure 9.4.1). After measuring the response for each set of factor levels, we identify the combination that gives the worst response and replace it with a new set of factor levels using a set of rules. This process continues until we reach the global optimum or until no further optimization is possible. The set of factor levels is called a **simplex**. In general, for k factors a simplex is a $k + 1$ dimensional geometric figure [see Spendley, W.; Hext, G. R.; Himsworth, F. R. *Technometrics* **1962**, 4, 441–461, and Deming, S. N.; Parker, L. R. *CRC Crit. Rev. Anal. Chem.* **1978** 7(3), 187–202]. Thus, for two factors the simplex is a triangle. For three factors the simplex is a tetrahedron.

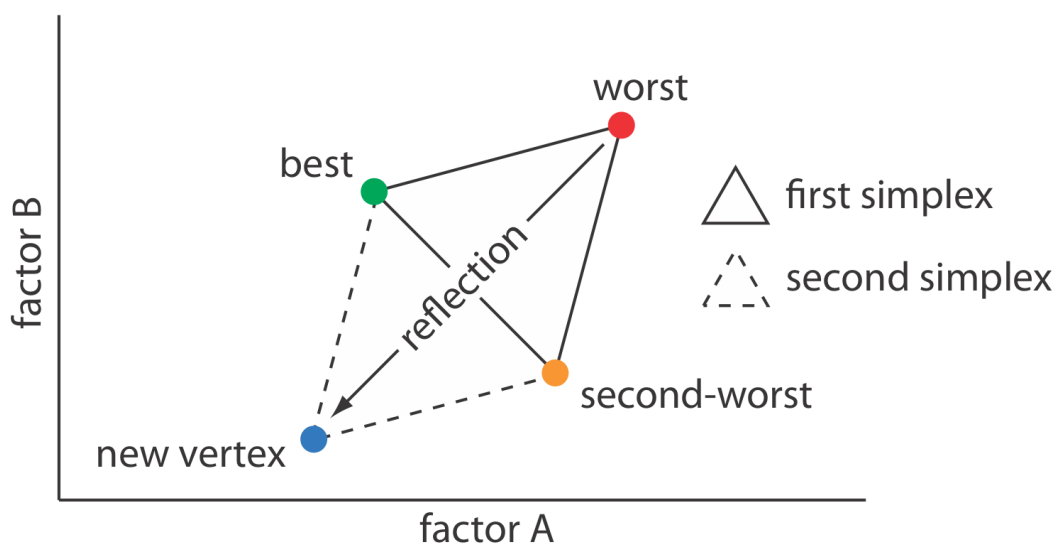


Figure 9.4.1. Example of a two-factor simplex. The original simplex is formed by the green, orange, and red vertices. Replacing the worst vertex with a new vertex moves the simplex to a new position on the response surface.

To place the initial two-factor simplex on the response surface, we choose a starting point (a, b) for the first vertex and place the remaining two vertices at $(a + s_a, b)$ and $(a + 0.5s_a, b + 0.87s_b)$ where s_a and s_b are step sizes for factor A and for factor B [see, for example, Long, D. E. *Anal. Chim. Acta* **1969**, 46, 193–206]. The following set of rules moves the simplex across the response surface in search of the optimum response:

Rule 1. Rank the vertices from best (v_b) to worst (v_w).

Rule 2. Reject the worst vertex (v_w) and replace it with a new vertex (v_n) by reflecting the worst vertex through the midpoint of the remaining vertices. The new vertex's factor levels are twice the average factor levels for the retained vertices minus the factor levels for the worst vertex. For a two-factor optimization, the equations are shown here where v_s is the third vertex.

$$a_{v_n} = 2 \left(\frac{a_{v_b} + a_{v_s}}{2} \right) - a_{v_w}$$

$$b_{v_n} = 2 \left(\frac{b_{v_b} + b_{v_s}}{2} \right) - b_{v_w}$$

Rule 3. If the new vertex has the worst response, then return to the previous vertex and reject the vertex with the second worst response, (v_s) calculating the new vertex's factor levels using rule 2. This rule ensures that the simplex does not return to the previous simplex.

Rule 4. Boundary conditions are a useful way to limit the range of possible factor levels. For example, it may be necessary to limit a factor's concentration for solubility reasons, or to limit the temperature because a reagent is thermally unstable. If the new vertex exceeds a boundary condition, then assign it the worst response and follow rule 3.

Because the size of the simplex remains constant during the search, this algorithm is called a fixed-sized simplex optimization. The following example illustrates the application of these rules.

Example 9.4.1

Find the optimum for the response surface described by the equation

$$R = 5.5 + 1.5A + 0.6B - 0.15A^2 - 0.0254B^2 - 0.0857AB$$

using the fixed-sized simplex searching algorithm. Use (0, 0) for the initial factor levels and set each factor's step size to 1.00.

Solution

Letting $a = 0$, $b = 0$, $s_a = 1.00$, and $s_b = 1.00$ gives the vertices for the initial simplex as

$$\text{vertex 1: } (a, b) = (0, 0)$$

$$\text{vertex 2: } (a + s_a, b) = (1.00, 0)$$

$$\text{vertex 3: } (a + 0.5s_a, b + 0.87s_b) = (0.50, 0.87)$$

The responses for the three vertices are shown in the following table

vertex	a	b	response
v_1	0	0	5.50
v_2	1.00	0	6.85
v_3	0.50	0.87	6.68

with v_1 giving the worst response and v_3 the best response. Following Rule 1, we reject v_1 and replace it with a new vertex; thus

$$a_{v_4} = 2 \left(\frac{1.00 + 0.50}{2} \right) - 0 = 1.50$$

$$b_{v_4} = 2 \left(\frac{0 + 0.87}{2} \right) - 0 = 0.87$$

The following table gives the vertices of the second simplex.

vertex	a	b	response
v_2	1.00	0	6.85
v_3	0.50	0.87	6.68
v_4	1.50	0.87	7.80

with v_3 giving the worst response and v_4 the best response. Following Rule 1, we reject v_3 and replace it with a new vertex; thus

$$a_{v_5} = 2 \left(\frac{1.00 + 1.50}{2} \right) - 0.50 = 2.00$$

$$b_{v_5} = 2 \left(\frac{0 + 0.87}{2} \right) - 0.87 = 0$$

The following table gives the vertices of the third simplex.

vertex	<i>a</i>	<i>b</i>	response
v_2	1.00	0	6.85
v_4	1.50	0.87	7.80
v_5	2.00	0	7.90

The calculation of the remaining vertices is left as an exercise. Figure 9.4.2 shows the progress of the complete optimization. After 29 steps the simplex begins to repeat itself, circling around the optimum response of (3, 7).

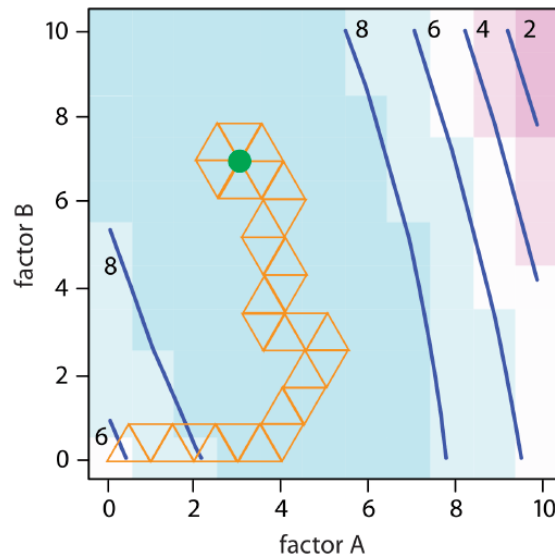


Figure 9.4.2. Progress of the fixed-size simplex optimization in Example 9.4.1. The green dot (•) marks the optimum response of (3,7). Optimization ends when the simplexes begin to circle around a single vertex.

The size of the initial simplex ultimately limits the effectiveness and the efficiency of a fixed-size simplex searching algorithm. We can increase its efficiency by allowing the size of the simplex to expand or to contract in response to the rate at which we approach the optimum. For example, if we find that a new vertex is better than any of the vertices in the preceding simplex, then we expand the simplex further in this direction on the assumption that we are moving directly toward the optimum. Other conditions might cause us to contract the simplex—to make it smaller—to encourage the optimization to move in a different direction. We call this a variable-sized simplex optimization. Consult this chapter's additional resources for further details of the variable-sized simplex optimization.

9.5: Mathematical Models of Response Surfaces

A response surface is described mathematically by an equation that relates the response to its factors. If we measure the response for several combinations of factor levels, then we can use a regression analysis to build a model of the response surface. There are two broad categories of models that we can use for a regression analysis: theoretical models and empirical models.

Theoretical Models of the Response Surface

A theoretical model is derived from the known chemical and physical relationships between the response and its factors. In spectrophotometry, for example, Beer's law is a theoretical model that relates an analyte's absorbance, A , to its concentration, C_A

$$A = \epsilon b C_A$$

where ϵ is the molar absorptivity and b is the pathlength of the electromagnetic radiation passing through the sample. A Beer's law calibration curve, therefore, is a theoretical model of a response surface. In Chapter 8 we learned how to use linear regression to build a mathematical model based on a theoretical relationship.

Empirical Models of the Response Surface

In many cases the underlying theoretical relationship between the response and its factors is unknown. We still can develop a model of the response surface if we make some reasonable assumptions about the underlying relationship between the factors and the response. For example, if we believe that the factors A and B are independent and that each has only a first-order effect on the response, then the following equation is a suitable model.

$$R = \beta_0 + \beta_a A + \beta_b B$$

where R is the response, A and B are the factor levels, and β_0 , β_a , and β_b are adjustable parameters whose values are determined by a linear regression analysis. Other examples of equations include those for dependent factors

$$R = \beta_0 + \beta_a A + \beta_b B + \beta_{ab} AB$$

and those with higher-order terms.

$$R = \beta_0 + \beta_a A + \beta_b B + \beta_{aa} A^2 + \beta_{bb} B^2$$

Each of these equations provides an empirical model of the response surface because it has no rigorous basis in a theoretical understanding of the relationship between the response and its factors. Although an empirical model may provide an excellent description of the response surface over a limited range of factor levels, it has no basis in theory and we cannot reliably extend it to unexplored parts of the response surface.

Factorial Designs

To build an empirical model we measure the response for at least two levels for each factor. For convenience we label these levels as high, H_f , and low, L_f , where f is the factor; thus H_A is the high level for factor A and L_B is the low level for factor B . If our empirical model contains more than one factor, then each factor's high level is paired with both the high level and the low level for all other factors. In the same way, the low level for each factor is paired with the high level and the low level for all other factors. As shown in Figure 9.5.1, this requires 2^k experiments where k is the number of factors. This experimental design is known as a 2^k factorial design.

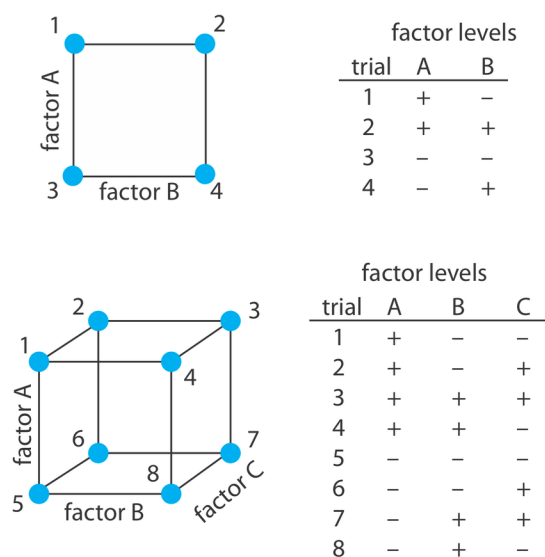


Figure 9.5.1. 2^k factorial designs for (top) $k = 2$, and for (bottom) $k = 3$. A 2^2 factorial design requires four experiments and a 2^3 factorial design requires eight experiments.

Note

Another system of notation is to use a plus sign (+) to indicate a factor's high level and a minus sign (-) to indicate its low level.

Determining the Empirical Model

A 2^2 factorial design requires four experiments and allows for an empirical model with four variables.

With four experiments, we can use a 2^2 factorial design to create an empirical model that includes four variables: an intercept, first-order effects in A and B, and an interaction term between A and B

$$R = \beta_0 + \beta_a A + \beta_b B + \beta_{ab} AB$$

The following example walks us through the calculations needed to find this model.

Example 9.5.1

Suppose we wish to optimize the yield of a synthesis and we expect that the amount of catalyst (factor A with units of mM) and the temperature (factor B with units of °C) are likely important factors. The response, R , is the reaction's yield in mg. We run four experiments and obtain the following responses:

run	A	B	R
1	15	20	145
2	25	20	158
3	15	30	135
4	25	30	150

Determine an equation for a response surface that provides a suitable model for predicting the effect of the catalyst and temperature on the reaction's yield.

Solution

Examining the data we see from runs 1 & 2 and from runs 3 & 4, that increasing factor A while holding factor B constant results in an increase in the response; thus, we expect that higher concentrations of the catalyst have a favorable affect on the reaction's yield. We also see from runs 1 & 3 and from runs 2 & 4, that increasing factor B while holding factor A constant results in a decrease in the response; thus, we expect that an increase in temperature has an unfavorable affect on the reaction's yield. Finally, we also see from runs 1 & 2 and from runs 3 & 4, that ΔR is more positive when factor B is at its higher level; thus, we expect that there is a positive interaction between factors A and B. With four experiments, we are limited to a model that considers an intercept, first-order effects in A and B, and an interaction term between A and B

$$R = \beta_0 + \beta_a A + \beta_b B + \beta_{ab} AB$$

We can work out values for this model's coefficients by solving the following set of simultaneous equations:

$$\beta_0 + 15\beta_a + 20\beta_b + (15)(20)\beta_{ab} = \beta_0 + 15\beta_a + 20\beta_b + 300\beta_{ab} = 145$$

$$\beta_0 + 25\beta_a + 20\beta_b + (25)(20)\beta_{ab} = \beta_0 + 25\beta_a + 20\beta_b + 500\beta_{ab} = 158$$

$$\beta_0 + 15\beta_a + 30\beta_b + (15)(30)\beta_{ab} = \beta_0 + 15\beta_a + 30\beta_b + 450\beta_{ab} = 135$$

$$\beta_0 + 25\beta_a + 30\beta_b + (25)(30)\beta_{ab} = \beta_0 + 25\beta_a + 30\beta_b + 750\beta_{ab} = 150$$

To solve this set of equations, we subtract the first equation from the second equation and subtract the third equation from the fourth equation, leaving us with the following two equations

$$10\beta_a + 200\beta_{ab} = 13$$

$$10\beta_a + 300\beta_{ab} = 15$$

Next, subtracting the first of these equations from the second gives

$$100\beta_{ab} = 2$$

or $\beta_{ab} = 0.02$. Substituting back gives

$$10\beta_a + 200 \times 0.02 = 13$$

or $\beta_a = 0.9$. Subtracting the equation for the first experiment from the equation for the third experiment gives

$$10\beta_b + 150\beta_{ab} = -10$$

Substituting in 0.02 for β_{ab} and solving gives $\beta_b = -1.3$. Finally, substituting in our values for β_a , β_b , and β_{ab} into any of the first four equations gives $\beta_0 = 151.5$. Our final model is

$$R = 151.5 + 0.9A - 1.3B + 0.02AB$$

When we consider how to interpret our empirical equation for the response surface, we need to consider several important limitations:

1. The intercept in our model represents a condition far removed from our experiments: In this case, the intercept gives the reaction's yield in the absence of catalyst and at a temperature of 0°C, either of which we may not be useful conditions. In general, it is never a good idea to extrapolate a model far beyond the conditions used to define the model.
2. The sign for a factor's first-order effects may be misleading if there is a significant interaction between it and other factors. Although our model shows that factor B (the temperature) has a negative first-order effect, the positive interaction between the two factors means there are conditions where an increase in B will increase the reaction's yield.
3. It is difficult to judge the relative importance of two or more factors by examining their coefficients if their scales are not the same. This could present a problem, for example, if we reported the amount of catalyst (factor A) using molar concentrations as these values would be three-orders of magnitude smaller than the reported temperatures.
4. When the number of variables is the same as the number of experiments, as is the case here, then there are no degrees of freedom and we have no simple way to test the model's suitability.

Determining the Empirical Model Using Coded Factor Levels

We can address two of the limitations described above by using coded factor levels in which we assign $+1$ for a high level and -1 for a low level. Defining the upper limit and the lower limit of the factors as $+1$ and -1 does two things for us: it places the intercept at the center of our experiments, which avoids the concern of extrapolating our model; and it places all factors on a common scale, and which makes it easier to compare the relative effects of the factors. Coding also makes it easier to determine the empirical model's equation when we complete calculations by hand.

Example 9.5.2

To explore the effect of temperature on a reaction, we assign 30°C to a coded factor level of -1 and assign a coded level $+1$ to a temperature of 50°C . What temperature corresponds to a coded level of -0.5 and what is the coded level for a temperature of 60°C ?

Solution

The difference between -1 and $+1$ is 2, and the difference between 30°C and 50°C is 20°C ; thus, each unit in coded form is equivalent to 10°C in uncoded form. With this information, it is easy to create a simple scale between the coded and the uncoded values, as shown in Figure 9.5.2. A temperature of 35°C corresponds to a coded level of -0.5 and a coded level of $+2$ corresponds to a temperature of 60°C .

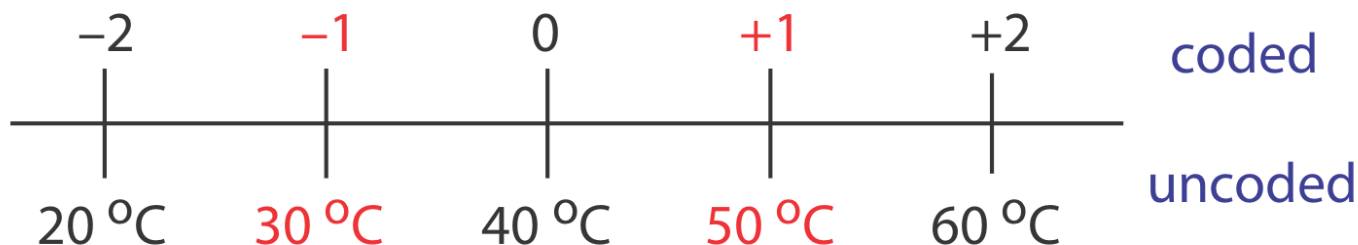


Figure 9.5.2. The relationship between the coded factor levels and the uncoded factor levels for Example 9.5.1. The numbers in red are the values defined in the 2^2 factorial design.

As we see in the following example, factor levels simplify the calculations for an empirical model.

Example 9.5.3

Rework Example 9.5.1 using coded factor levels.

Solution

The table below shows the original factor levels (A and B), their corresponding coded factor levels (A^* and B^*) and A^*B^* , which is the empirical model's interaction term.

run	A	B	A^*	B^*	A^*B^*	R
1	15	20	-1	-1	$+1$	145
2	25	20	$+1$	-1	-1	158
3	15	30	-1	$+1$	-1	135
4	25	30	$+1$	$+1$	$+1$	150

The empirical equation has four unknowns—the four beta terms—and Table 9.5.1 describes the four experiments. We have just enough information to calculate values for β_0 , β_a , β_b , and β_{ab} . When working with the coded factor levels, the values of these parameters are easy to calculate using the following equations, where n is the number of runs.

$$\beta_0 \approx b_0 = \frac{1}{n} \sum_{i=1}^n R_i$$

$$\beta_a \approx b_a = \frac{1}{n} \sum_{i=1}^n A_i^* R_i$$

$$\beta_b \approx b_b = \frac{1}{n} \sum_{i=1}^n B_i^* R_i$$

$$\beta_{ab} \approx b_{ab} = \frac{1}{n} \sum_{i=1}^n A_i^* B_i^* R_i$$

Solving for the estimated parameters using the data in Table 9.5.1

$$b_0 = \frac{145 + 158 + 135 + 150}{4} = 147$$

$$b_a = \frac{-145 + 158 - 135 + 150}{4} = 7$$

$$b_b = \frac{-145 - 11.5 + 135 + 150}{4} = 5.0$$

$$b_{ab} = \frac{145 - 158 - 135 + 150}{4} = 0.5$$

leaves us with the coded empirical model for the response surface.

$$R = 147 + 7A^* - 4.5B^* + 0.5A^*B^*$$

Note

Do you see why the equations for calculating b_0 , b_a , b_b , and b_{ab} work? Take the equation for b_a as an example

$$\beta_a \approx b_a = \frac{1}{n} \sum_{i=1}^n A_i^* R_i$$

where

$$b_a = \frac{-145 + 158 - 135 + 150}{4} = 7$$

The first and the third terms in this equation give the response when A^* is at its low level, and the second and fourth terms in this equation give the response when A^* is at its high level. In the two terms where A^* is at its low level, B^* is at both its low level (first term) and its high level (third term), and in the two terms where A^* is at its high level, B^* is at both its low level (second term) and its high level (fourth term). As a result, the contribution of B^* is removed from the calculation. The same holds true for the effect of A^*B^* , although this is left for you to confirm.

We can transform the coded model into a non-coded model by noting that $A = 20 + 5A^*$ and that $B = 25 + 5B^*$, solving for A^* and B^* , to obtain $A^* = 0.2A - 4$ and $B^* = 0.2B - 5$, and substituting into the coded model and simplifying.

$$R = 147 + 7(0.2A - 4) - 4.5(0.2B - 5) + 0.5(0.2A - 4)(0.2A - 5)$$

$$R = 147 + 1.4A - 28 - 0.9B + 22.5 + 0.02AB - 0.5A - 0.4B + 10$$

$$R = 151.5 + 0.9A - 1.3B + 0.02AB$$

Note that this is the same equation that we derived in Example 9.5.1 using uncoded values for the factors.

Although we can convert this coded model into its uncoded form, there is no need to do so. If we want to know the response for a new set of factor levels, we just convert them into coded form and calculate the response. For example, if A is 23 and B is 22, then $A^* = 0.2 \times 23 - 4 = 0.6$ and $B^* = 0.2 \times 22 - 5 = -0.6$ and

$$R = 147 + 7 \times 0.6 - 4.5 \times (-0.6) + 0.5 \times 0.6 \times (-0.6) = 153.72 \approx 154 \text{ mg}$$

We can extend this approach to any number of factors. For a system with three factors— A , B , and C —we can use a 2^3 factorial design to determine the parameters in the following empirical model

$$R = \beta_0 + \beta_a A + \beta_b B + \beta_c C + \beta_{ab} AB + \beta_{ac} AC + \beta_{bc} BC + \beta_{abc} ABC$$

where A , B , and C are the factor levels. The terms β_0 , β_a , β_b , and β_{ab} are estimated using the following eight equations.

$$\beta_0 \approx b_0 = \frac{1}{n} \sum_{i=1}^n R_i$$

$$\beta_a \approx b_a = \frac{1}{n} \sum_{i=1}^n A_i^* R_i$$

$$\beta_b \approx b_b = \frac{1}{n} \sum_{i=1}^n B_i^* R_i$$

$$\beta_{ab} \approx b_{ab} = \frac{1}{n} \sum_{i=1}^n A_i^* B_i^* R_i$$

$$\beta_c \approx b_c = \frac{1}{n} \sum_{i=1}^n C_i^* R_i$$

$$\beta_{ac} \approx b_{ac} = \frac{1}{n} \sum_{i=1}^n A_i^* C_i^* R_i$$

$$\beta_{bc} \approx b_{bc} = \frac{1}{n} \sum_{i=1}^n B_i^* C_i^* R_i$$

$$\beta_{abc} \approx b_{abc} = \frac{1}{n} \sum_{i=1}^n A_i^* B_i^* C_i^* R_i$$

Example 9.5.4

The following table lists the uncoded factor levels, the coded factor levels, and the responses for a 2^3 factorial design.

run	A	B	C	A*	B*	C*	A*B*	A*C*	B*C*	A*B*C*	R
1	15	30	45	+1	+1	+1	+1	+1	+1	+1	137.5
2	15	30	15	+1	+1	-1	+1	-1	-1	-1	54.75
3	15	10	45	+1	-1	+1	-1	+1	-1	-1	73.75
4	15	10	15	+1	-1	-1	-1	-1	+1	+1	30.25
5	5	30	45	-1	+1	+1	-1	-1	+1	-1	61.75
6	5	30	15	-1	+1	-1	-1	+1	-1	+1	30.25
7	5	10	45	-1	-1	+1	+1	-1	-1	+1	41.25
8	5	10	15	-1	-1	-1	+1	+1	+1	-1	18.75

Determine the coded empirical model for the response surface based on the following equation.

$$R = \beta_0 + \beta_a A + \beta_b B + \beta_c C + \beta_{ab} AB + \beta_{ac} AC + \beta_{bc} BC + \beta_{abc} ABC$$

What is the expected response when A is 10, B is 15, and C is 50?

Solution

The equation for the empirical model has eight unknowns—the eight beta terms—and the table above describes eight experiments. We have just enough information to calculate values for β_0 , β_a , β_b , β_{ab} , β_{ac} , β_{bc} , and β_{abc} ; these values are

$$b_0 = \frac{1}{8} \times (137.25 + 54.75 + 73.75 + 30.25 + 61.75 + 30.25 + 41.25 + 18.75) = 56.0$$

$$b_a = \frac{1}{8} \times (137.25 + 54.75 + 73.75 + 30.25 - 61.75 - 30.25 - 41.25 - 18.75) = 18.0$$

$$b_b = \frac{1}{8} \times (137.25 + 54.75 - 73.75 - 30.25 + 61.75 + 30.25 - 41.25 - 18.75) = 15.0$$

$$b_c = \frac{1}{8} \times (137.25 - 54.75 + 73.75 - 30.25 + 61.75 - 30.25 + 41.25 - 18.75) = 22.5$$

$$b_{ab} = \frac{1}{8} \times (137.25 + 54.75 - 73.75 - 30.25 - 61.75 - 30.25 + 41.25 + 18.75) = 7.0$$

$$b_{ac} = \frac{1}{8} \times (137.25 - 54.75 + 73.75 - 30.25 - 61.75 + 30.25 - 41.25 + 18.75) = 9.0$$

$$b_{bc} = \frac{1}{8} \times (137.25 - 54.75 - 73.75 + 30.25 + 61.75 - 30.25 - 41.25 + 18.75) = 6.0$$

$$b_{abc} = \frac{1}{8} \times (137.25 - 54.75 - 73.75 + 30.25 - 61.75 + 30.25 + 41.25 - 18.75) = 3.75$$

The coded empirical model, therefore, is

$$R = 56.0 + 18.0A^* + 15.0B^* + 22.5C^* + 7.0A^*B^* + 9.0A^*C^* + 6.0B^*C^* + 3.75A^*B^*C^*$$

To find the response when A is 10, B is 15, and C is 50, we first convert these values into their coded form. Figure 9.5.3 helps us make the appropriate conversions; thus, A^* is 0, B^* is -0.5, and C^* is +1.33. Substituting back into the empirical model gives a response of

$$R = 56.0 + 18.0(0) + 15.0(-0.5) + 22.5(+1.33) + 7.0(0)(-0.5) + 9.0(0)(+1.33) + 6.0(-0.5)(+1.33) + 3.75(0)(-0.5)(+1.33) = 74.43$$

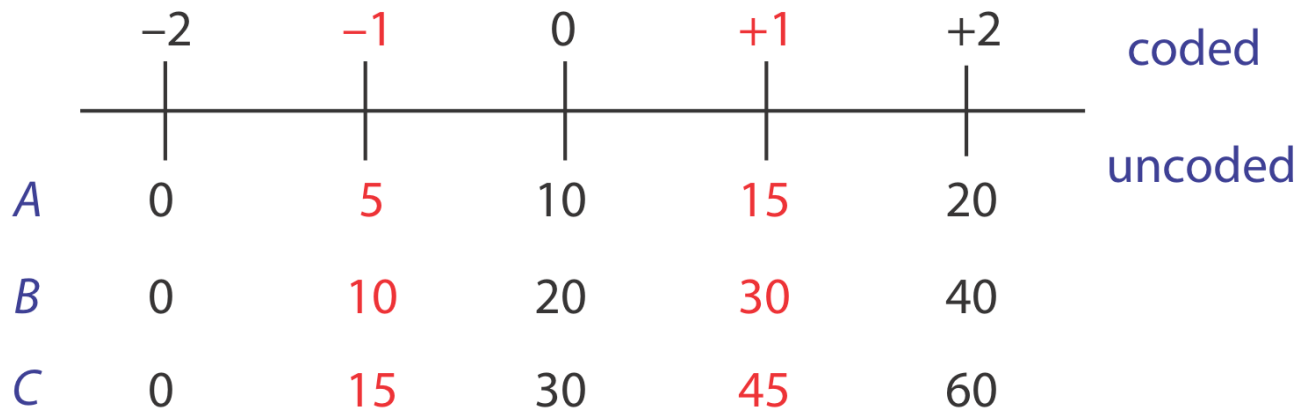


Figure 9.5.2. The relationship between the coded factor levels and the uncoded factor levels for Example 9.5.2. The numbers in red are the values defined in the 2^3 factorial design.

Evaluating an Empirical Model

A 2^k factorial design can model only a factor's first-order effect, including first-order interactions, on the response. A 2^2 factorial design, for example, includes each factor's first-order effect (β_a and β_b) and a first-order interaction between the factors (β_{ab}). A 2^k factorial design cannot model higher-order effects because there is insufficient information. Here is a simple example that illustrates the problem. Suppose we need to model a system in which the response is a function of a single factor, A. Figure 9.5.4a shows the result of an experiment using a 2^1 factorial design. The only empirical model we can fit to the data is a straight line.

$$R = \beta_0 + \beta_a A$$

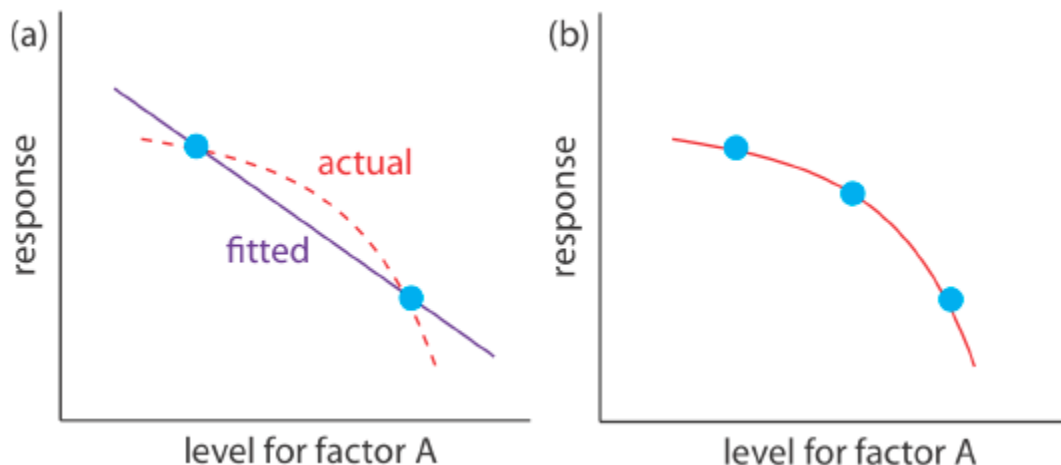


Figure 9.5.4. A curved one-factor response surface, in red, showing (a) the limitation of using a 2^1 factorial design, which can fit only a straight-line to the data, and (b) the application of a 3^1 factorial design that takes into account second-order effects.

If the actual response is a curve instead of a straight-line, then the empirical model is in error. To see evidence of curvature we must measure the response for at least three levels for each factor. We can fit the 3^1 factorial design in Figure 9.5.4b to an empirical model that includes second-order factor effects.

$$R = \beta_0 + \beta_a A + \beta_{aa} A^2$$

In general, an n -level factorial design can model single-factor and interaction terms up to the $(n - 1)^{\text{th}}$ order.

We can judge the effectiveness of a first-order empirical model by measuring the response at the center of the factorial design. If there are no higher-order effects, then the average response of the trials in a 2^k factorial design should equal the measured response at the center of the factorial design. To account for influence of random errors we make several determinations of the response at the center of the factorial design and establish a suitable confidence interval. If the difference between the two responses is significant, then a first-order empirical model probably is inappropriate.

Note

One of the advantages of working with a coded empirical model is that b_0 is the average response of the $2 \times k$ trials in a 2^k factorial design.

Example 9.5.5

One method for the quantitative analysis of vanadium is to acidify the solution by adding H_2SO_4 and oxidizing the vanadium with H_2O_2 to form a red-brown soluble compound with the general formula $(\text{VO})_2(\text{SO}_4)_3$. Palasota and Deming studied the effect of the relative amounts of H_2SO_4 and H_2O_2 on the solution's absorbance, reporting the following results for a 2^2 factorial design [Palasota, J. A.; Deming, S. N. *J. Chem. Educ.* **1992**, 62, 560–563].

H_2SO_4	H_2O_2	absorbance
+1	+1	0.330
+1	−1	0.359
−1	+1	0.293
−1	−1	0.420

Four replicate measurements at the center of the factorial design give absorbances of 0.334, 0.336, 0.346, and 0.323. Determine if a first-order empirical model is appropriate for this system. Use a 90% confidence interval when accounting for the effect of random error.

Solution

We begin by determining the confidence interval for the response at the center of the factorial design. The mean response is 0.335 with a standard deviation of 0.0094, which gives a 90% confidence interval of

$$\mu = \bar{X} \pm \frac{ts}{\sqrt{n}} = 0.335 \pm \frac{(2.35)(0.0094)}{\sqrt{4}} = 0.335 \pm 0.011$$

The average response, $\bar{R}$, from the factorial design is

$$\bar{R} = \frac{0.330 + 0.359 + 0.293 + 0.420}{4} = 0.350$$

Because $\bar{R}$ exceeds the confidence interval's upper limit of 0.346, we can reasonably assume that a 2^2 factorial design and a first-order empirical model are inappropriate for this system at the 95% confidence level.

Central Composite Designs

One limitation to a 3^k factorial design, which would allow us to use an empirical model with second-order effects, is the number of trials we need to run. As shown in Figure 9.5.5, a 3^2 factorial design requires 9 trials. This number increases to 27 for three factors and to 81 for 4 factors.

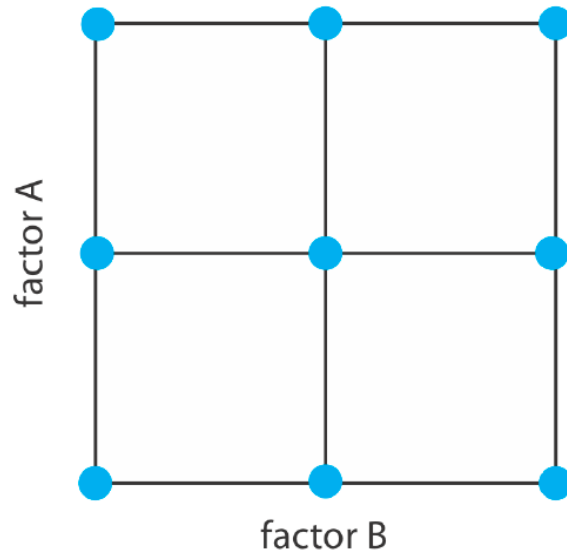


Figure 9.5.5. A 3^k factorial design for $k = 2$.

A more efficient experimental design for a system that contains more than two factors is a central composite design, two examples of which are shown in Figure 9.5.6. The central composite design consists of a 2^k factorial design, which provides data to estimate each factor's first-order effect and interactions between the factors, and a star design that has $2^k + 1$ points, which provides data to estimate second-order effects. Although a central composite design for two factors requires the same number of trials, nine, as a 3^2 factorial design, it requires only 15 trials and 25 trials when using three factors or four factors. See this chapter's additional resources for details about the central composite designs.

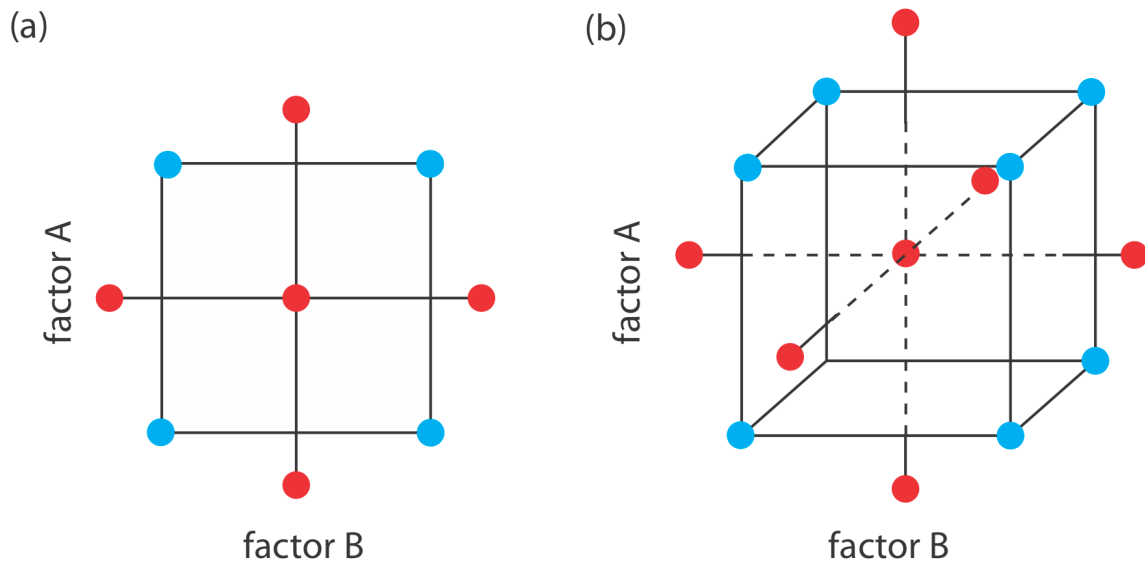


Figure 9.5.5. Two examples of a central composite design for (a) $k = 2$ and (b) $k = 3$. The points in blue are a 2^k factorial design, and the points in red are a star design.

9.6: Using R to Model a Response Surface (Multiple Regression)

The calculations for determining an empirical model of a response surface using a 2^k factorial design, as outlined in Section 9.5, are relatively easy to complete for a small number of factors and for experimental designs without replication where the number of experiments is equal to the number of parameters in the model. If we wish to work with more factors, if we wish to explore other experimental designs, and if we wish to build replication into the experimental design so that we can better evaluate our empirical model, then we need to do so by building a regression model, as we did earlier in Chapter 8.

Creating Empirical Models Using R

To illustrate how we can use R to create an empirical model, let's use data from an experiment exploring how to optimize a Grignard reaction leading to the synthesis of benzyl-1-cyclopentan-1-ol [Bouzidi, N.; Gozzi, C. *J. Chem. Educ.* **2008**, 85, 1544–1547]. In this study, students begin by studying the effect of six possible factors on the reaction's yield: the volume of diethyl ether used to prepare a solution of benzyl chloride, x_1 , the time over which benzyl chloride is added to the reaction mixture, x_2 , the stirring time used to prepare the benzyl magnesium chloride, x_3 , the relative excess of benzyl chloride to cyclopentanone, x_4 , the relative excess of magnesium turnings to benzyl chloride, x_5 , and the reaction time, x_6 .

With six factors to consider, a full 2^k factorial design requires 32 experiments, which is labor intensive. Instead, the students begin with a screening study that uses eight experiments to model only the first-order effects of the six factors, as outlined in the following two tables.

Table 9.6.1: Factor Levels for Screening Study

factor	low level	high level
x_1 : volume of diethyl ether in mL	18	50
x_2 : addition time in min	60	90
x_3 : stirring time in min	20	40
x_4 : relative excess of benzyl chloride as %	20	30
x_5 : relative excess of magnesium as %	12.5	25
x_6 : reaction time in min	30	60

Table 9.6.2: Experimental Design Showing Coded Factor Levels and Responses

run	x_1	x_2	x_3	x_4	x_5	x_6	percent yield
1	+1	+1	+1	−1	+1	−1	72
2	−1	+1	+1	+1	−1	+1	33
3	−1	−1	+1	+1	+1	−1	29
4	+1	−1	−1	+1	+1	+1	74
5	−1	+1	−1	−1	+1	+1	31
6	+1	+1	+1	−1	−1	+1	52
7	+1	−1	−1	+1	−1	−1	47
8	−1	−1	−1	−1	−1	−1	27

To carry out the calculations in R we first create vectors for the coded factor levels and the responses.

```
x1 = c(1, -1, -1, 1, -1, 1, 1, -1)
```

```
x2 = c(1,1,-1,-1,1,-1,1,-1)
x3 = c(1,1,1,-1,-1,1,-1,-1)
x4 = c(-1,1,1,1,-1,-1,1,-1)
x5 = c(1,-1,1,1,1,-1,-1,-1)
x6 = c(-1,1,-1,1,1,1,-1,-1)
yield = c(72,33,29,74,31,52,47,27)
```

Next, we use the `lm()` function to build a linear regression model that includes just the first-order effects of the factors (see Chapter 8.5 to review the syntax for this function), and the `summary()` function to review the resulting model.

```
screening = lm(yield ~ x1 + x2 + x3 + x4 + x5 + x6)
summary(screening)
```

Call:

```
lm(formula = yield ~ x1 + x2 + x3 + x4 + x5 + x6)
```

Residuals:

```
1 2 3 4 5 6 7 8
5.875 5.875 -5.875 5.875 -5.875 -5.875 -5.875 5.875
```

Coefficients:

```
Estimate Std.Error t value Pr(>|t|)
```

```
(Intercept) 45.625 5.875 7.766 0.0815 .
```

```
x1 15.625 5.875 2.660 0.2290
```

```
x2 0.125 5.875 0.021 0.9865
```

```
x3 0.875 5.875 0.149 0.9059
```

```
x4 0.125 5.875 0.021 0.9865
```

```
x5 5.875 5.875 1.000 0.5000
```

```
x6 1.875 5.875 0.319 0.8033
```

```
---
```

```
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 16.62 on 1 degrees of freedom
```

```
Multiple R-squared: 0.8913, Adjusted R-squared: 0.239
```

```
F-statistic: 1.366 on 6 and 1 DF, p-value: 0.5749
```

Because we have one more experiment than there are variables in our empirical model, the summary provides some information on the significance of the model's parameters; however, with just one degree of freedom this information is not really reliable. In addition to the intercept, the three factors with the largest coefficients are the volume of diethyl ether, x_1 , the relative excess of magnesium, x_5 , and the reaction time, x_6 .

Having identified three factors for further investigation, the students use a 2^3 factorial design to explore interactions between these three factors using the experimental design in the following table (see Table 9.6.1 for the actual factor levels).

Table 9.6.3: Coded Factor Levels and Response for a 2^3 Factorial Design

run	x_1	x_5	x_6	percent yield
1	-1	-1	-1	28.5
2	+1	-1	-1	55.5
3	-1	+1	-1	38
4	+1	+1	-1	68
5	-1	-1	+1	49
6	+1	-1	+1	66
7	-1	+1	+1	31.5
8	+1	+1	+1	72

As before, we create vectors for our factors and the response and then use the `lm()` and the `summary()` functions to complete and evaluate the resulting empirical model.

```
x1 = c(-1,1,-1,1,-1,1,-1,1)
x5 = c(-1,-1,1,1,-1,-1,1,1)
x6 = c(-1,-1,-1,-1,1,1,1,1)
yield = c(28.5,55.5,38,68,49,66,31.5,72)
fact23 = lm(yield ~ x1 * x5 * x6)
summary(fact23)
```

Call:

```
lm(formula = yield ~ x1 * x5 * x6)
```

Residuals:

ALL 8 residuals are 0: no residual degrees of freedom!

Coefficients:

```
Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) 51.0625 NA NA NA
```

```
x1 14.3125 NA NA NA
```

```
x5 1.3125 NA NA NA
```

```
x6 3.5625 NA NA NA
```

```
x1:x5 3.3125 NA NA NA
```

```
x1:x6 0.0625 NA NA NA
```

```
x5:x6 -4.1875 NA NA NA
```

```
x1:x5:x6 2.5625 NA NA NA
```

Residual standard error: NaN on 0 degrees of freedom

Multiple R-squared: 1, Adjusted R-squared: NaN

F-statistic: NaN on 7 and 0 DF, p-value: NA

With eight experiments and eight variables in the empirical model, we do not have any ability to evaluate the model statistically. Of the three first-order effects, we see that the volume of diethyl ether, x_1 , and reaction time, x_6 , are more important than the relative excess of magnesium, x_5 . We also see that the interaction between x_1 and x_5 is positive (high values for both favor an increased yield) and that the interaction between x_5 and x_6 is negative (yields improve when one factor is high and the other is low).

Finally, the students use a central composite model—which allows for adding second-order effects and curvature in the response surface—to study the effect of the volume of diethyl ether, x_1 , and reaction time, x_6 , on the percent yield. The relative excess of magnesium, x_5 was set at its high level for this study because this provides for greater percent yields (compare the results for runs 4 and 6 to the results for runs 3 and 5 in Table 9.6.3). The following tables provides the experimental design.

Table 9.6.4: Coded Factor Levels and Responses for a Central Composite Experimental Design

run	x_1	x_6	percent yield
1	−1	−1	39
2	+1	−1	66.5
3	−1	+1	22
4	+1	+1	72.5
5	−1.414	0	10.5
6	+1.414	0	72.5
7	0	−1.414	38
8	0	+1.414	70
9	0	0	59
10	0	0	57
11	0	0	54.5
12	0	0	63

As before, we create vectors for our factors and the response, and then use the `lm()` and the `summary()` functions to complete and evaluate the resulting empirical model.

```
x1 = c(-1,1,-1,1,-1.414,1.414,0,0,0,0,0,0)
x6 = c(-1,-1,1,1,0,0,-1.414,1.414,0,0,0,0)
yield = c(39,66.5,22,72.5,10.5,72.5,38,70,59,57,54.5,63)
centcomp = lm(yield ~ x1 * x6 + I(x1^2) + I(x6^2))
summary(centcomp)
```

Call:

```
lm(formula = yield ~ x1 * x6 + I(x1^2) + I(x6^2))
```

Residuals:

Min 1Q Median 3Q Max

```
-11.0724 -4.0794 -0.3938 5.2056 9.3695
```

Coefficients:

```
Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) 58.375 4.360 13.389 1.07e-05 ***
```

```
x1 20.712 3.083 6.718 0.000529 ***
```

```
x6 4.282 3.083 1.389 0.214267
I(x1^2) -7.876 3.447 -2.285 0.062398 .
I(x6^2) -1.625 3.447 -0.471 0.654130
x1:x6 5.750 4.360 1.319 0.235317
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: 8.72 on 6 degrees of freedom
Multiple R-squared: 0.9, Adjusted R-squared: 0.8167
F-statistic: 10.8 on 5 and 6 DF, p-value: 0.005835
```

With 12 experiments and just six variables, our model has sufficient degrees of freedom to suggest that it provides a reasonable picture of how the reaction time and the volume of diethyl ether affect the reaction's yield even if the residual errors in the responses range from a minimum of -11.7 to a maximum +9.37. The middle 50% of residual errors range between -4.1 to +5.2 with a median residual error of -0.4. We can compare the actual experimental yields to the yields predicted by the model by combining them into a data frame.

```
centcomp_results = data.frame(yield, centcomp$fitted.values, yield - centcomp$fitted.values)
colnames(centcomp_results) = c("expt yield", "pred yield", "residual error")
centcomp_results
```

```
expt yield pred yield residual error
```

```
1 39.0 29.63046 9.3695385
2 66.5 59.55372 6.9462836
3 22.0 26.69375 -4.6937546
4 72.5 79.61701 -7.1170095
5 10.5 13.34036 -2.8403635
6 72.5 71.91285 0.5871540
7 38.0 49.07236 -11.0723566
8 70.0 61.18085 8.8191471
9 59.0 58.37466 0.6253402
10 57.0 58.37466 -1.3746598
11 54.5 58.37466 -3.8746598
12 63.0 58.37466 4.6253402
```

Using R to Visualize the Response Surface

The `plot3D` package provides several functions that we can use to visualize a response surface defined by two factors. Here we consider three functions, one for drawing a two-dimensional contour plot of the response surface, one for drawing a three-dimensional surface plot of the response, and one for plotting a three-dimensional scatter plot of the responses. To begin, we

use the `library()` function to make the package available to us (note: you may need to first install the `plot3D` package; see Chapter 1 for details on how to do this).

```
library(plot3D)
```

Let's begin by creating a two-dimensional contour plot of our response surface that places the volume of diethyl ether, x_1 , on the x-axis and the reaction time, x_6 on the y-axis, and using calculated responses from the model to draw the contour lines. First, we create vectors with values for the x-axis and the y-axis

```
x1_axis = seq(-1.5, 1.5, 0.1)
x6_axis = seq(-1.5, 1.5, 0.1)
```

Next, we create a function that uses our empirical model to calculate the response for every combination of `x1_axis` and `x6_axis`

```
response = function(x,y){coef(centcomp)[1] + coef(centcomp)[2]*x + coef(centcomp)[3]*y +
coef(centcomp)[4]*x^2 + coef(centcomp)[5]*y^2 + coef(centcomp)[6]*x*y}
```

where `coef(centcomp)[i]` is used to extract the i^{th} coefficient from our empirical model. Now we use R's `outer()` function to calculate the response for every combination of the variables `x1_axis` and `x6_axis`

```
z_axis = outer(X = x1_axis,Y = x6_axis, response)
```

Finally, we use the `contour2D()` function to create the contour plot in Figure 9.6.1.

```
contour2D(x = x1_axis,y = x6_axis, z = z_axis, xlab = "x1: volume", ylab = "x6: time", clab =
"yield")
```

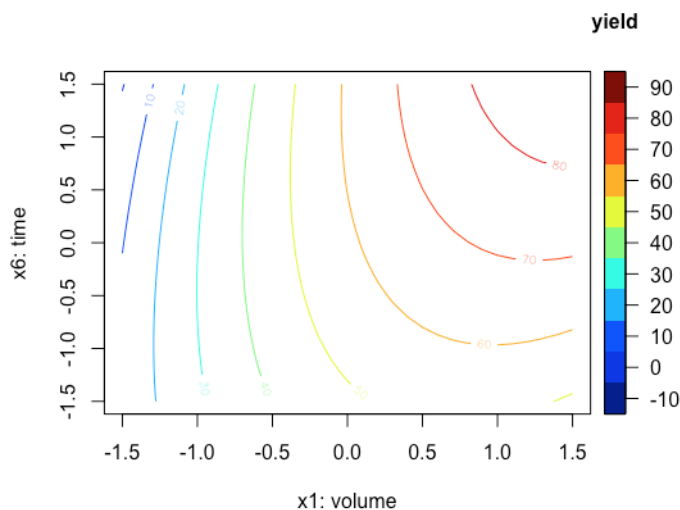


Figure 9.6.1: Contour plot for the response surface predicting the percent yield in a Grignard reaction as a function of the volume of diethyl ether and the reaction time. The x-axis and the y-axis values are coded factor levels.

Next, let's create a three-dimensional surface plot of our response surface that places the volume of diethyl ether, x_1 , on the x-axis, the reaction time, x_6 on the y-axis, and the calculated responses from the model on the z-axis. For this, we use the `persp3D()` function

```
persp3D(x = x1_axis, y = x6_axis, z = z_axis, ticktype = "detailed", phi = 15, theta = 25, xlab =
"x1: volume", ylab = "x6: time", zlab = "yield", clab = "yield", contour = TRUE, cex.axis =
0.75, cex.lab = 0.75)
```

where `phi` and `theta` adjust the angle at which we view the response surface—you will have to play with these values to create

a plot that is pleasing to look at—and `ticktype` controls how much information is displayed on the axes. The `cex.axis` and `cex.lab` commands adjust the size of the text displayed on the axes, and `countour = TRUE` places a contour plot on the figure's bottom side. Figure 9.6.2 shows the result.

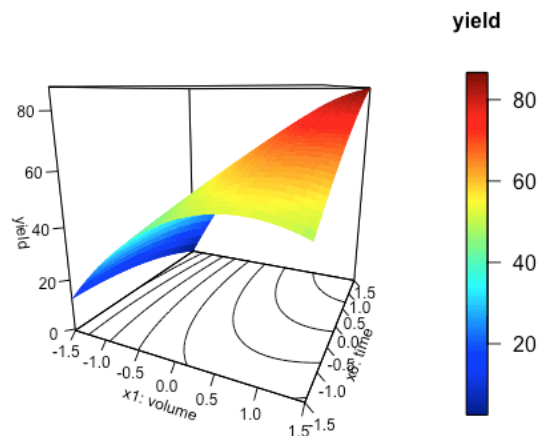


Figure 9.6.2: Three-dimensional surface (perspective) plot for the response surface predicting the percent yield in a Grignard reaction as a function of the volume of diethyl ether and the reaction time. The x-axis and the y-axis values are coded factor levels.

Finally, let's use the `type = "h"` option to overlay a scatterplot of the data used to build the empirical model on top of the three-dimensional surface plot.

```
scatter3D(x = x1, y = x6, z = yield, add = TRUE, type = "h", pch = 19, col = "black", lwd = 2, colkey = FALSE)
```

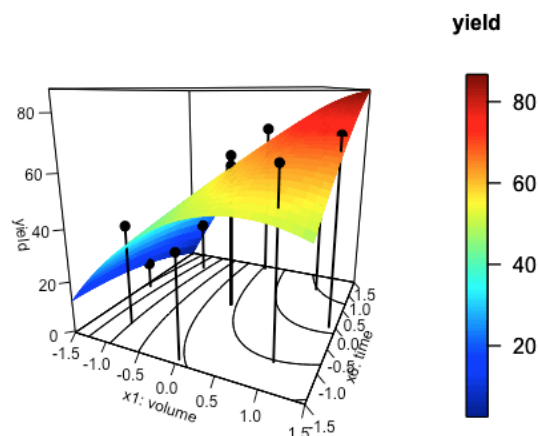


Figure 9.6.3: Three-dimensional surface (perspective) plot for the response surface predicting the percent yield in a Grignard reaction as a function of the volume of diethyl ether and the reaction time showing the original data used to build the empirical model. The x-axis and the y-axis values are coded factor levels.

Figure 9.6.3 shows the result using the data from Table 9.6.4. Although the general shape of the response surface is consistent with the underlying data, there is sufficient experimental uncertainty in the results of the four replicate experiments used to create this empirical model, as shown by the standard deviation for runs 9–12, to explain why some of the predicted yields have large errors.

```
sd(yield[9:12])
```

```
[1] 3.591077
```

9.7: Exercises

1. For each of the following equations determine the optimum response using a one-factor-at-a-time searching algorithm. Begin the search at (0,0) by first changing factor A, using a step-size of 1 for both factors. The boundary conditions for each response surface are $0 \leq A \leq 10$ and $0 \leq B \leq 10$. Continue the search through as many cycles as necessary until you find the optimum response. Compare your optimum response for each equation to the true optimum. Note: These equations are from Deming, S. N.; Morgan, S. L. *Experimental Design: A Chemometric Approach*, Elsevier: Amsterdam, 1987, and pseudo-three dimensional plots of the response surfaces can be found in their Figures 11.4, 11.5 and 11.14.

(a) $R = 1.68 + 0.24A + 0.56B - 0.04A^2 - 0.04B^2$ $\mu_{\text{opt}} = (3,7)$

(b) $R = 4.0 - 0.4A + 0.08AB$ $\mu_{\text{opt}} = (10,10)$

(c) $R = 3.264 + 1.537A + 0.5664B - 0.1505A^2 - 0.02734B^2 - 0.05785AB$ $\mu_{\text{opt}} = (3.91, 6.22)$

2. Use a fixed-sized simplex searching algorithm to find the optimum response for the equation in Problem 1c. For the first simplex, set one vertex at (0,0) with step sizes of one. Compare your optimum response to the true optimum.

3. A 2^k factorial design was used to determine the equation for the response surface in Problem 1b. The uncoded levels, coded levels, and the responses are shown in the following table. Determine the uncoded equation for the response surface.

A	B	A*	B*	response
8	8	+1	+1	5.92
8	2	+1	-1	2.08
2	8	-1	+1	4.48
2	2	-1	-1	3.52

4. Koscielniak and Parczewski investigated the influence of Al on the determination of Ca by atomic absorption spectrophotometry using the 2^k factorial design shown in the following table [data from Koscielniak, P.; Parczewski, A. *Anal. Chim. Acta* **1983**, 153, 111-119].

[Ca ²⁺] (ppm)	[Al ³⁺] (ppm)	Ca*	Al*	response
10	160	+1	+1	54.92
10	0	+1	-1	98.44
4	16	-1	+1	19.18
4	0	-1	-1	38.52

(a) Determine the uncoded equation for the response surface.

(b) If you wish to analyze a sample that is 6.0 ppm Ca²⁺, what is the maximum concentration of Al³⁺ that can be present if the error in the response must be less than 5.0%?

5. Strange [Strange, R. S. *J. Chem. Educ.* **1990**, 67, 113-115] studied a chemical reaction using the following 2^3 factorial design.

factor	high (+1) level	low (-1) level
X: temperature	140°C	120°C
Y: catalyst	type B	type A

factor	high (+1) level	low (-1) level
Z: [reactant]	0.50 M	0.25 M

run	X*	Y*	Z*	% yield
1	-1	-1	-1	28
2	+1	-1	-1	17
3	-1	+1	-1	41
4	+1	+1	-1	34
5	-1	-1	+1	56
6	+1	-1	+1	51
7	-1	+1	+1	42
8	+1	+1	+1	36

(a) Determine the coded equation for this data.

(b) If β terms of less than ± 1 are insignificant, what main effects and what interaction terms in the coded equation are important? Write down this simpler form for the coded equation.

(c) Explain why the coded equation for this data can not be transformed into an uncoded form.

(d) Which is the better catalyst, A or B?

(e) What is the yield if the temperature is set to 125°C, the concentration of the reactant is 0.45 M, and we use the appropriate catalyst?

6. Pharmaceutical tablets coated with lactose often develop a brown discoloration. The primary factors that affect the discoloration are temperature, relative humidity, and the presence of a base acting as a catalyst. The following data have been reported for a 2^3 factorial design [Armstrong, N. A.; James, K. C. *Pharmaceutical Experimental Design and Interpretation*, Taylor and Francis: London, 1996 as cited in Gonzalez, A. G. *Anal. Chim. Acta* **1998**, 360, 227–241].

factor	high (+1) level	low (-1) level
X: benzocaine	present	absent
Y: temperature	40°C	25°C
Z: relative humidity	75%	50%

run	X*	Y*	Z*	color (arb. unit)
1	-1	-1	-1	1.55
2	+1	-1	-1	5.40
3	-1	+1	-1	3.50
4	+1	+1	-1	6.75
5	-1	-1	+1	2.45
6	+1	-1	+1	3.60
7	-1	+1	+1	3.05
8	+1	+1	+1	7.10

(a) Determine the coded equation for this data.

(b) If β terms of less than 0.5 are insignificant, what main effects and what interaction terms in the coded equation are important? Write down this simpler form for the coded equation.

7. The following data for a 2^3 factorial design were collected during a study of the effect of temperature, pressure, and residence time on the % yield of a reaction [Akhnazarova, S.; Kafarov, V. *Experimental Optimization in Chemistry and Chemical Engineering*, MIR Publishers: Moscow, 1982 as cited in Gonzalez, A. G. *Anal. Chim. Acta* **1998**, 360, 227–241].

factor	high (+1) level	low (–1) level
X: temperature	200°C	100°C
Y: pressure	0.6 MPa	0.2 MPa
Z: residence time	20 min	10 min

run	X*	Y*	Z*	% yield
1	–1	–1	–1	2
2	+1	–1	–1	6
3	–1	+1	–1	4
4	+1	+1	–1	8
5	–1	–1	+1	10
6	+1	–1	+1	18
7	–1	+1	+1	8
8	+1	+1	+1	12

(a) Determine the coded equation for this data.

(b) If β terms of less than 0.5 are insignificant, what main effects and what interaction terms in the coded equation are important? Write down this simpler form for the coded equation.

(c) Three runs at the center of the factorial design—a temperature of 150°C, a pressure of 0.4 MPa, and a residence time of 15 min—give percent yields of 8%, 9%, and 8.8%. Determine if a first-order empirical model is appropriate for this system at $\alpha = 0.05$.

8. Duarte and colleagues used a factorial design to optimize a flow-injection analysis method for determining penicillin [Duarte, M. M. M. B.; de O. Netro, G.; Kubota, L. T.; Filho, J. L. L.; Pimentel, M. F.; Lima, F.; Lins, V. *Anal. Chim. Acta* **1997**, 350, 353–357]. Three factors were studied: reactor length, carrier flow rate, and sample volume, with the high and low values summarized in the following table.

factor	high (+1) level	low (–1) level
X: reactor length	1.3 cm	2.0 cm
Y: carrier flow rate	1.6 mL/min	2.2 mL/min
Z: sample volume	100 μ L	150 μ L

The authors determined the optimum response using two criteria: the greatest sensitivity, as determined by the change in potential for the potentiometric detector, and the largest sampling rate. The following table summarizes their optimization results.

run	X^*	Y^*	Z^*	ΔE (mV)	sample/h
1	-1	-1	-1	37.45	21.5
2	+1	-1	-1	31.70	26.0
3	-1	+1	-1	32.10	30.0
4	+1	+1	-1	27.30	33.0
5	-1	-1	+1	39.85	21.0
6	+1	-1	+1	32.85	19.5
7	-1	+1	+1	35.00	30.0
8	+1	+1	+1	32.15	34.0

- (a) Determine the coded equation for the response surface where ΔE is the response.
- (b) Determine the coded equation for the response surface where sample/h is the response.
- (c) Based on the coded equations in (a) and in (b), do conditions that favor sensitivity also improve the sampling rate?
- (d) What conditions would you choose if your goal is to optimize both sensitivity and sampling rate?

9. Here is a challenge! McMinn, Eatherton, and Hill investigated the effect of five factors for optimizing an H_2 -atmosphere flame ionization detector using a 2^5 factorial design [McMinn, D. G.; Eatherton, R. L.; Hill, H. H. *Anal. Chem.* **1984**, 56, 1293-1298]. The factors and their levels were

factor	high (+1) level	low (-1) level
A: H_2 flow rate	1460 mL/min	1382 mL/min
B: SiH_4	20.0 ppm	12.2 ppm
C: $O_2 + N_2$ flow rate	255 mL/min	210 mL/min
D: O_2/N_2 ratio	1.36	1.19
E: electrode height	75 (arb. unit)	55 (arb. unit)

The coded ("+" = +1, "-" = -1) factor levels and responses, R , for the 32 experiments are shown in the following table

run	A^*	B^*	C^*	D^*	E^*	R	run	A^*	B^*	C^*	D^*	E^*	R
1	-	-	-	-	-	0.36	17	-	-	-	-	+	0.39
2	+	-	-	-	-	0.51	18	+	-	-	-	+	0.45
3	-	+	-	-	-	0.15	19	-	+	-	-	+	0.32
4	+	+	-	-	-	0.39	20	+	+	-	-	+	0.25
5	-	-	+	-	-	0.79	21	-	-	+	-	+	0.18
6	+	-	+	-	-	0.83	22	+	-	+	-	+	0.29
7	-	+	+	-	-	0.74	23	-	+	+	-	+	0.07
8	+	+	+	-	-	0.69	24	+	+	+	-	+	0.19
9	-	-	-	+	-	0.60	25	-	-	-	+	+	0.53
10	+	-	-	+	-	0.82	26	+	-	-	+	+	0.60
11	-	+	-	+	-	0.42	27	-	+	-	+	+	0.36

run	A*	B*	C*	D*	E*	R	run	A*	B*	C*	D*	E*	R
12	+	+	-	+	-	0.59	28	+	+	-	+	+	0.43
13	-	-	+	+	-	0.96	29	-	-	+	+	+	0.23
14	+	-	+	+	-	0.87	30	+	-	+	+	+	0.51
15	-	+	+	+	-	0.76	31	-	+	+	+	+	0.13
16	+	+	+	+	-	0.74	32	+	+	+	+	+	0.43

(a) Determine the coded equation for this response surface, ignoring β terms less than ± 0.03 .

(b) A simplex optimization of this system finds optimal values for the factors of $A = 2278$ mL/min, $B = 9.90$ ppm, $C = 260.6$ mL/min, and $D = 1.71$. The value of E was maintained at its high level. Are these values consistent with your analysis of the factorial design.

10. A good empirical model provides an accurate picture of the response surface over the range of factor levels within the experimental design. The same model, however, may yield an inaccurate prediction for the response at other factor levels. For this reason, an empirical model, is tested before it is extrapolated to conditions other than those used in determining the model. For example, Palasota and Deming studied the effect of the relative amounts of H_2SO_4 and H_2O_2 on the absorbance of solutions of vanadium using the following central composite design [data from Palasota, J. A.; Deming, S. N. *J. Chem. Educ.* **1992**, 62, 560-563].

run	drops of 1% H_2SO_4	drops of 20% H_2O_2
1	15	22
2	10	20
3	20	20
4	8	15
5	15	15
6	15	15
7	15	15
8	15	15
9	22	15
10	10	10
11	20	10
12	15	8

The reaction of H_2SO_4 and H_2O_2 generates a red-brown solution whose absorbance is measured at a wavelength of 450 nm. A regression analysis on their data yields the following uncoded equation for the response (absorbance $\times 1000$).

$$R = 835.90 - 36.82X_1 - 21.34X_2 + 0.52X_1^2 + 0.15X_2^2 + 0.98X_1X_2$$

where X_1 is the drops of H_2O_2 , and X_2 is the drops of H_2SO_4 . Calculate the predicted absorbances for 10 drops of H_2O_2 and 0 drops of H_2SO_4 , 0 drops of H_2O_2 and 10 drops of H_2SO_4 , and for 0 drops of each reagent. Are these results reasonable? Explain. What does your answer tell you about this empirical model?

10: CLEANING UP DATA

When we try to calibrate an analytical method (Chapter 8) or to optimize an analytical system (Chapter 9), our ability to do so successfully is limited by the uncertainty, or noise, in our measurements and by background signals that interfere with our ability to measure the signal of interest to us. In this chapter we will consider ways to clean up our data by decreasing the contribution of noise to our measurements and by correcting for the presence of background signals.

[10.1: Signals and Noise](#)

[10.2: Improving the Signal-to-Noise Ratio](#)

[10.3: Background Removal](#)

[10.4: Using R to Clean Up Data](#)

[10.5: Exercises](#)

10.1: Signals and Noise

When we make a measurement it is the sum of two parts, a determinate, or fixed contribution that arises from the analyte and an indeterminate, or random, contribution that arises from uncertainty in the measurement process. We call the first of these the signal and we call the latter the noise. There are two broad categories of noise: that associated with obtaining samples and that associated with making measurements. Our interest here is in the latter.

What is Noise?

Noise is a random event characterized by a mean and standard deviation. There are many types of noise, but we will limit ourselves for now to noise that is stationary, in that its mean and its standard deviation are independent of time, and that is heteroscedastic, in that its mean and its variance (and standard deviation) are independent of the signal's magnitude. Figure 10.1.1a shows an example of a noisy signal that meets these criteria. The x-axis here is shown as time—perhaps a chromatogram—but other units, such as wavelength or potential, are possible. Figure 10.1.1b shows the underlying noise and Figure 10.1.1c shows the underlying signal. Note that the noise in Figure 10.1.1b appears consistent in its central tendency (mean) and its spread (variance) along the x-axis and is independent of the signal's strength.

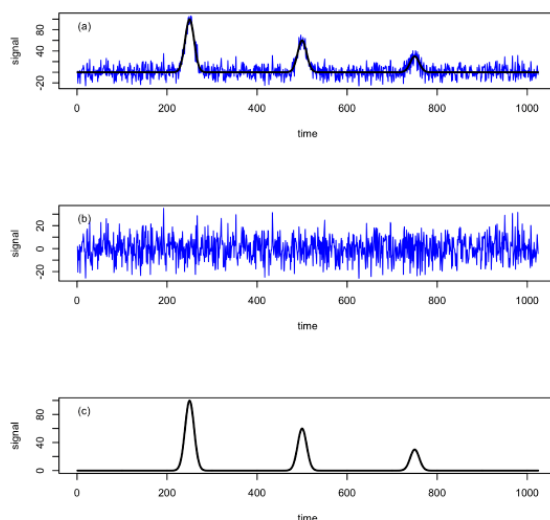


Figure 10.1.1: Plots showing (a) the signal and the noise in blue with the signal superimposed as a smooth line; (b) the noise only; and (c) the signal only. The signal consists of three peaks at times of 250, 500, and 750, and with maximum values of 100, 60, and 30, respectively. The noise is drawn at random from a normal distribution with a mean of 0 and a standard deviation of 10.

How Do We Characterize the Signal and the Noise?

Although we characterize noise by its mean and its standard deviation, the most important benchmark is the **signal-to-noise ratio**, S/N , which we define as

$$S/N = \frac{S_{\text{analyte}}}{s_{\text{noise}}}$$

where S_{analyte} is the signal's value at particular location on the x-axis and s_{noise} is the standard deviation of the noise using a signal-free portion of the data. As general rules-of-thumb, we can measure the signal with some confidence when $S/N \geq 3$ and we can detect the signal with some confidence when $3 \geq S/N \geq 2$. For the data in Figure 10.1.1, and using the information in the figure caption, the signal-to-noise ratios are, from left-to-right, 10, 6, and 3.

Note

To measure the signal with confidence implies we can use the signal's value in a calculation, such as constructing a calibration curve. To detect the signal with confidence means we are certain that a signal is present (and that an analyte responsible for the signal is present) even if we cannot measure the signal with sufficient confidence to allow for a meaningful calculation.

How Can We Improve the S/N Ratio?

There are two broad approaches that we can use to improve the signal-to-noise ratio: hardware and software. Hardware approaches are built into the instrument and include decisions on how the instrument is set-up for making measurements (for example, the choice of a scan rate or a slit width), and how the signal is processed by the instrument (for example, using electronic filters); such solutions are not of interest to us here in a textbook with a focus on chemometrics. Software solutions are computational approaches in which we manipulate the data either while we are collecting it or after data acquisition is complete.

10.2: Improving the Signal-to-Noise Ratio

In this section we will consider three common computational tools for improving the signal-to-noise ratio: signal averaging, digital smoothing, and Fourier filtering.

Signal Averaging

The most important difference between the signal and the noise is that a signal is determinate (fixed in value) and the noise is indeterminate (random in value). If we measure a pure signal several times, we expect its value to be the same each time; thus, if we add together n scans, we expect that the net signal, S_n , is defined as

$$S_n = nS$$

where S is the signal for a single scan. Because noise is random, its value varies from one run to the next, sometimes with a value that is larger and sometimes with a value that is smaller, and sometimes with a value that is positive and sometimes with a value that is negative. On average, the standard deviation of the noise increases as we make more scans, but it does so at a slower rate than for the signal

$$s_n = \sqrt{n}s$$

where s is the standard deviation for a single scan and s_n is the standard deviation after n scans. Combining these two equations, shows us that the signal-to-noise ratio, S/N , after n scans increases as

$$(S/N)_n = \frac{S_n}{s_n} = \frac{nS}{\sqrt{n}s} = \sqrt{n}(S/N)_{n=1}$$

where $(S/N)_{n=1}$ is the signal-to-noise ratio for the initial scan. Thus, when $n = 4$ the signal-to-noise ratio improves by a factor of 2, and when $n = 16$ the signal-to-noise ratio increases by a factor of 4. Figure 10.2.1 shows the improvement in the signal-to-noise ratio for 1, 2, 4, and 8 scans.

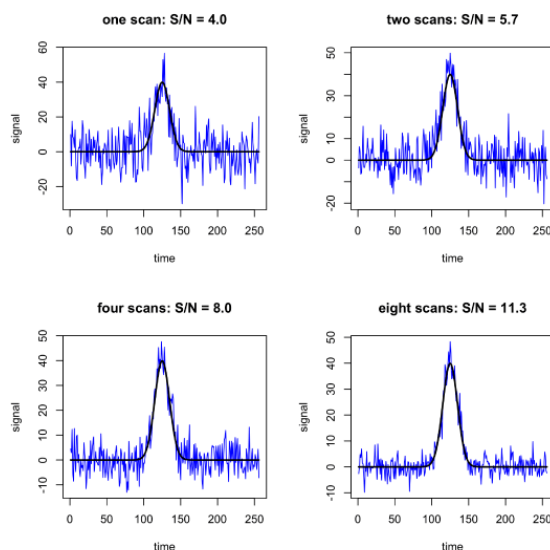


Figure 10.2.1: Improvement in the signal-to-noise ratio through signal averaging using 1, 2, 4, and 8 scans. Each plot shows the noisy signal in blue with the pure signal superimposed in black. The total signal is divided by the number of scans so that each y-axis has the same scale.

Signal averaging works well when the time it takes to collect a single scan is short and when the analyte's signal is stable with respect to time both because the sample is stable and the instrument is stable; when this is not the case, then we risk a time-dependent change in S_{analyte} and/or s_{noise} . Because the equation for $(S/N)_n$ is proportional to the $\sqrt{n}$, the relative improvement in the signal-to-noise ratio decreases as n increases; for example, 16 scans gives a $4 \times$ improvement in the signal-to-noise ratio,

but it takes an additional 48 scans (for a total of 64 scans) to achieve a $8 \times$ improvement in the signal-to-noise ratio.

Digital Smoothing Filters

One characteristic of noise is that its magnitude fluctuates rapidly in contrast to the underlying signal. We see this, for example, in Figure 10.2.1 where the underlying signal either remains constant or steadily increases or decreases while the noise fluctuates chaotically. Digital smoothing filters take advantage of this by using a mathematical function to average the data for a small range of consecutive data points, replacing the range's middle value with the average signal over that range.

Moving Average Filters

For a moving average filter, we replace each point by the average signal for that point and an equal number of points on either side; thus, a moving average filter has a width, w , of 3, 5, 7, ... points. For example, suppose the first five points in a sequence are

0.80	0.30	0.80	0.20	1.00
------	------	------	------	------

then a three-point moving average ($w = 3$) returns values of

NA	0.63	0.43	0.67	NA
----	------	------	------	----

where, for example, 0.63 is the average of 0.80, 0.30, and 0.80. Note that we lose $(w - 1)/2 = (3 - 1)/2 = 1$ points at each end of the data set because we do not have a sufficient number of data points to complete a calculation for the first and the last point. Figure 10.2.2 shows the improvement in the S/N ratio when using moving average filters with widths of 5, 9, and 13.

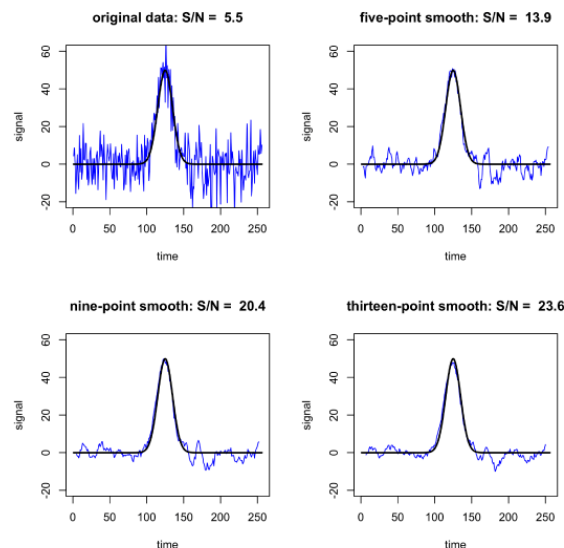


Figure 10.2.2: Improvement in the signal-to-noise ratio using moving average filters with ranges of 5, 9, and 13 on the original data shown in the upper left quadrant. Each plot shows the noisy signal in blue with the pure signal superimposed in black.

One limitation to a moving average filter is that it distorts the original data by removing points from both ends, although this is not a serious concern if the points in question are just noise. Of greater concern is the distortion in a signal's height if we use a range that is too wide; for example, Figure 10.2.3, shows how a 23-point moving average filter (shown in blue) applied to the noisy signal in the upper left quadrant of Figure 10.2.2, reduces the height of the original signal (shown in black). Because the filter's width—shown by the red bar—is similar to the peak's width, as the filter passes through the peak it systematically reduces the signal by averaging together values that are mostly smaller than the maximum signal.

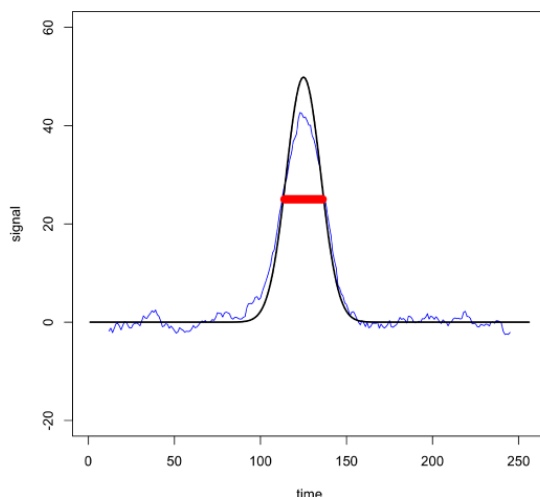


Figure 10.2.3: Example that shows how a moving average filter can distort the signal when applied to the noisy signal in the upper left quadrant of Figure 10.2.2. The original pure signal is shown in black and the signal after applying a 23-point moving average filter is shown in blue. The width of the moving average filter is shown by the red bar.

Savitzky-Golay Filters

A moving average filter weights all points equally; that is, points near the edges of the filter contribute to the average as a level equal to points near the filter's center. A Savitzky-Golay filter uses a polynomial model that weights each point differently, placing more weight on points near the center of the filter and less weight on points at the edge of the filter. Specific values depend on the size of the window and the polynomial model; for example, a five-point filter using a second-order polynomial has weights of

$$-3/35 \quad 12/35 \quad 17/35 \quad 12/35 \quad -3/35$$

For example, suppose the first five points in a sequence are

0.80	0.30	0.80	0.20	1.00
------	------	------	------	------

then this Savitzky-Golay filter returns values of

NA	NA	0.41	NA	NA
----	----	------	----	----

where, for example, the value for the middle point is

$$0.80 \times \frac{-3}{35} + 0.30 \times \frac{12}{35} + 0.80 \times \frac{17}{35} + 0.20 \times \frac{12}{35} + 1.00 \times \frac{-3}{35} = 0.406 \approx 0.41$$

Note that we lose $(w-1)/2 = (5-1)/2 = 2$ points at each end of the data set, where w is the filter's range, because we do not have a sufficient number of data points to complete the calculations. For other Savitzky-Golay smoothing filters, see [Savitzky, A.; Golay, M. J. E. *Anal Chem*, 1964, 36, 1627-1639](#). Figure 10.2.4 shows the improvement in the S/N ratio when using Savitzky-Golay filters using a second-order polynomial with 5, 9, and 13 points.

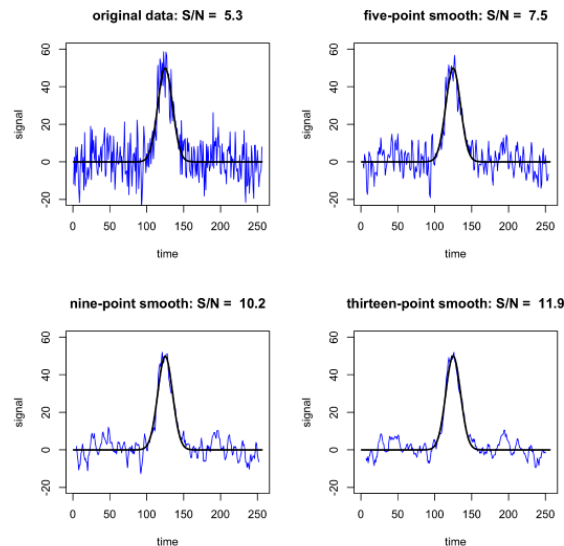


Figure 10.2.4: Improvement in the signal-to-noise ratio using Savitzky-Golay filters of ranges of 5, 9, and 13 on the original data shown in the upper left quadrant. Each plot shows the noisy signal in blue with the pure signal superimposed in black.

Because a Savitzky-Golay filter weights points differently than does a moving average smoothing filter, a Savitzky-Golay filter introduces less distortion to the signal, as we see in the following figure.

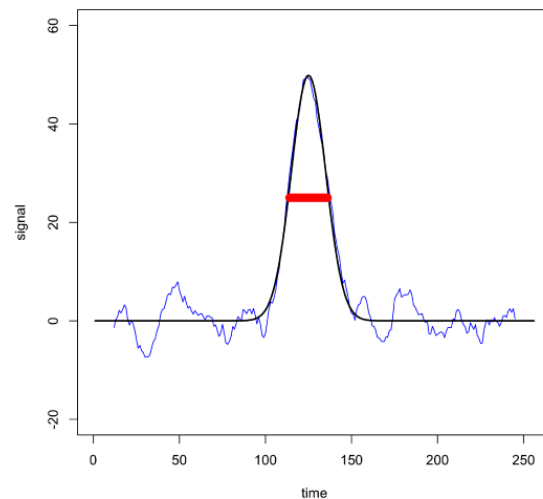


Figure 10.2.5: A Savitzky-Golay filter is less aggressive than a moving average filter. Applying a 23-point Savitzky-Golay filter to the noisy signal in the upper left quadrant of Figure 10.2.4 results in little distortion of the signal. Contrast this with Figure 10.2.3 where a 23-point moving average filter results in substantial distortion of the signal. The original pure signal is shown in black and the signal after applying a 23-point Savitzky-Golay filter is shown in blue. The width of the Savitzky-Golay filter is shown by the red bar.

Fourier Filtering

This approach to improving the signal-to-noise ratio takes advantage of a mathematical technique called a Fourier transform (FT). The basis of a Fourier transform is that we can express a signal in two separate domains. In the first domain the signal is characterized by one or more peaks, each defined by its position, its width, and its area; this is called the frequency domain. In the second domain, which is called the time domain, the signal consists of a set of oscillations, each defined by its frequency, its amplitude, and its decay rate. The Fourier transform—and the inverse Fourier transform—allow us to move between these two domains.

Note

The mathematical details behind the Fourier transform are beyond the level of this textbook; for a more in-depth treatment, consult this chapter's resources.

Figure 10.2.6a shows a single peak in the frequency domain and Figure 10.2.6b shows its equivalent time domain signal. There are correlations between the two domains:

- the further a peak in the frequency domain is from the origin, the greater its corresponding oscillation frequency in the time domain
- the broader a peak's width in the frequency domain, the faster its decay rate in the time domain
- the greater the area under a peak in the frequency domain, the higher its initial intensity in the time domain

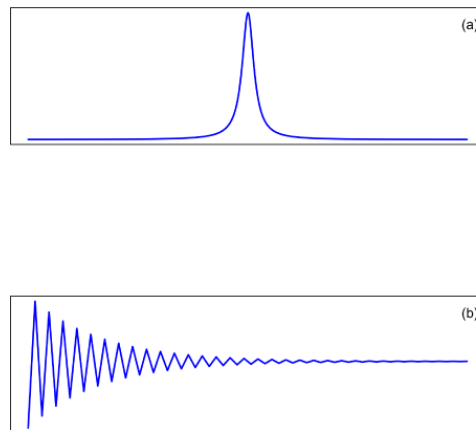


Figure 10.2.6: The plot in (a) shows a frequency domain consisting of a single peak defined by its position along the x -axis, its width, and its area. The plot in (b) shows the corresponding time domain that consists of a single oscillating signal defined by its oscillation frequency, its initial intensity, and its decay rate.

We can use a Fourier transform to improve the signal-to-noise ratio because the signal is a single broad peak and the noise appears as a multitude of very narrow peaks. As noted above, a broad peak in the frequency domain has a fast decaying signal in the time domain, which means that while the beginning of the time domain signal includes contributions from the signal and the noise, the latter part of the time domain signal includes contributions from noise only. The figure below shows how we can take advantage of this to reduce the noise and improve the signal-to-noise ratio for the noisy signal in Figure 10.2.7a, which has 256 points along the x -axis and has a signal-to-noise ratio of 5.1. First, we use the Fourier transform to convert its original domain into the new domain, the first 128 points of which are shown in Figure 10.2.7b (note: the first half of the data contains the same information as the second half of the data, so we only need to look at the first half of the data). The points at the beginning are dominated by the signal, which is why there is a systematic decrease in the intensity of the oscillations; the remaining points are dominated by noise, which is why the variation in intensity is random. To filter out the noise we retain the first 24 points as they are and set the intensities of the remaining points to zero (the choice of how many points to retain may require some adjustment). As shown in Figure 10.2.7c, we repeat this for the remaining 128 points, retaining the last 24 points as they are. Finally, we use an inverse Fourier transform to return to our original domain, with the result in Figure 10.2.7d, with the signal-to-noise ratio improving from 5.1 for the original noisy signal to 11.2 for the filtered signal.

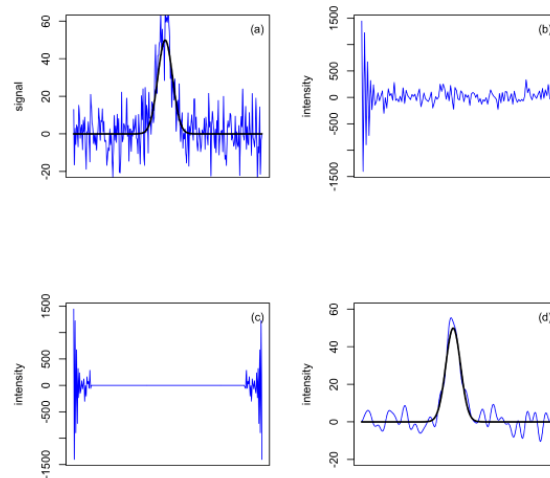


Figure 10.2.7: Example of removing noise using a Fourier filter. The original noisy signal with $S/N = 5.1$ is shown in (a) and is similar to the noisy signal in Figure 10.2.2a and Figure 10.2.4a. The first half of the Fourier transformed data is shown in (b). The Fourier transformed data is shown in (c) after zeroing out all but the first and the last 24 points. Returning to the original domain gives the final filtered signal with $S/N = 11.1$.

10.3: Background Removal

Another form of noise is a systematic background signal on which the analytical signal of interest is overlaid. For example, the following figure shows a Gaussian signal with a maximum value of 50 centered at $x = 125$ superimposed on an exponential background. The dotted line is the Gaussian signal, which has a maximum value of 50 at $x = 125$, and the solid line is the signal as measured, which has a maximum value of 57 at $x = 125$.

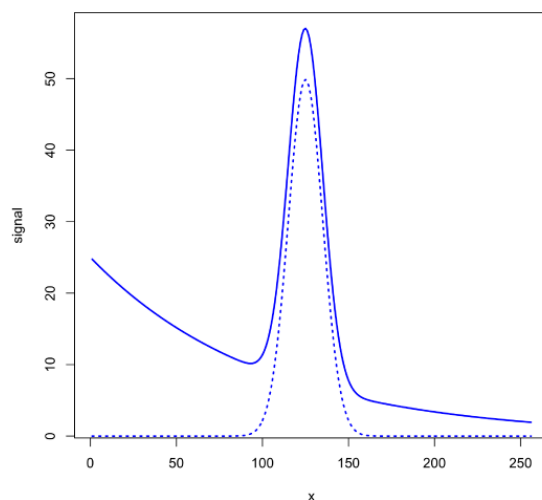


Figure 10.3.1: A Gaussian signal (dotted line) superimposed on an exponential background, which gives rise to the measured signal (solid line).

If the background signal is consistent across all samples, then we can analyze the data without first removing its contribution. For example, the following figure shows a set of calibration standards and their resulting calibration curve, for which the y -intercept of 7 gives the offset introduced by the background.

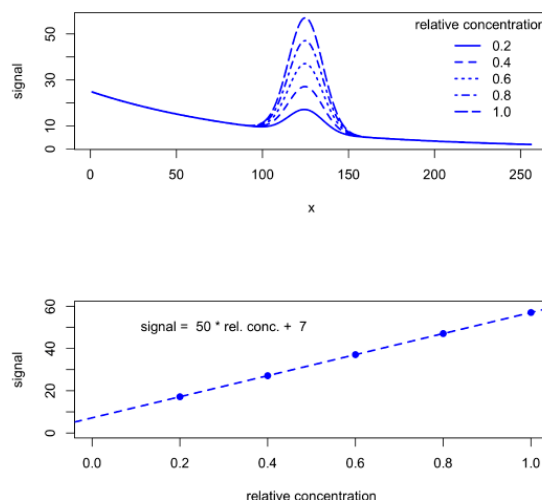


Figure 10.3.2: When the background is the same for all calibration standards and samples, then we can construct a calibration curve without taking into account the presence of the background.

But background signals often are not consistent across samples, particularly when the source of the background is a property of the samples we collect (natural water samples, for example, may have variations in color due to differences in the concentration

of dissolved organic matter) or a property of the instrument we are using (such as a variation in source intensity over time). When true, our data may look more like what we see in the following figure, which leads to a calibration curve with a greater uncertainty.

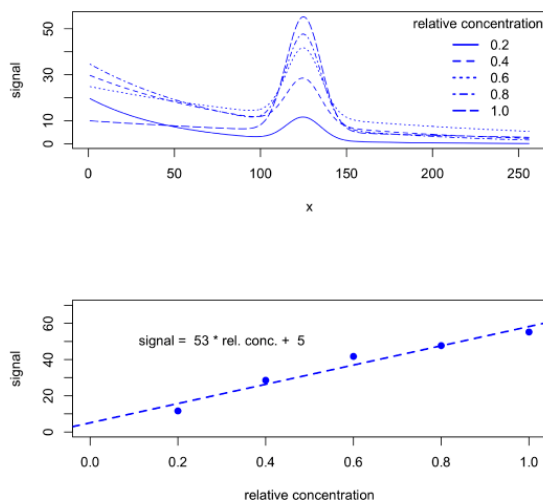


Figure 10.3.3: When the background is not the same for all calibration standards, the quality of the calibration curves suffers, making it less useful for the analysis of samples.

Because the background changes gradually with the values for x while the analyte's signal changes quickly, we can use a derivative to distinguish between the two. One approach is to use a Savitzky-Golay derivative filter using the same approach described in the last section. For example, applying a 7-point first-derivative Savitzky-Golay filter with weights of

$$-3/28 \quad -2/28 \quad -1/28 \quad 0/28 \quad 1/28 \quad 2/28 \quad 3/28$$

to the data in Figure 10.3.3 gives the results shown below. The calibration signal in this case is the difference between the maximum signal and the minimum signal, which are shown by the dotted red lines in the top part of the figure. The fit of the calibration curve to the data and the calibration curve's y -intercept of zero shows that we have successfully compensated for the background signals.

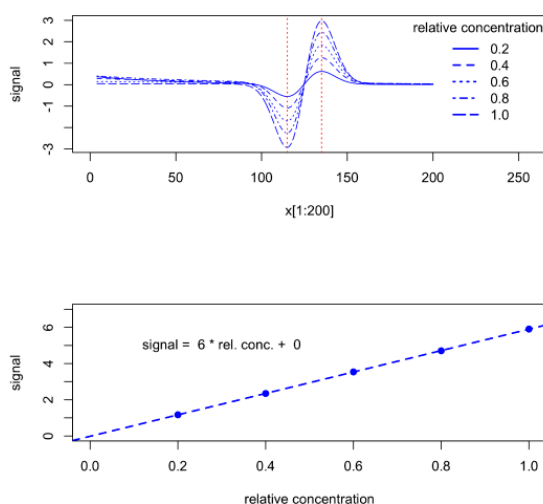


Figure 10.3.4: Applying a Savitzky-Golay derivative filter to the calibration curve data in Figure 10.3.3 corrects for the differences in the background signals, yielding an improved calibration curve.

For other Savitzky-Golay derivative filters, including second-derivative filters, see Savitzky, A.; Golay, M. J. E. *Anal Chem*, **1964**, 36,

1627-1639.

This page titled [10.3: Background Removal](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [David Harvey](#).

10.4: Using R to Clean Up Data

R has two useful functions, `filter()` and `fft()`, that we can use to smooth or filter noise and to remove background signals. To explore their use, let's first create two sets of data that we can use as examples: a noisy signal and a pure signal superimposed on an exponential background. To create the noisy signal, we first create a vector of 256 values that defines the x-axis; although we will not specify a unit here, these could be times or frequencies. Next we use R's `dnorm()` function to generate a pure Gaussian signal with a mean of 125 and a standard deviation of 10, and R's `rnorm()` function to generate 256 points of random noise with a mean of zero and a standard deviation of 10. Finally, we add the pure signal and the noise to arrive at our noisy signal and then plot the noisy signal and overlay the pure signal.

```
x = seq(1,256,1)
gaus_signal = 1250 * dnorm(x, mean = 125, sd = 10)
noise = rnorm(256, mean = 0, sd = 10)
noisy_signal = gaus_signal + noise
plot(x = x, y = noisy_signal, type = "l", lwd = 2, col = "blue", xlab = "x", ylab = "signal")
lines(x = x, y = gaus_signal, lwd = 2)
```

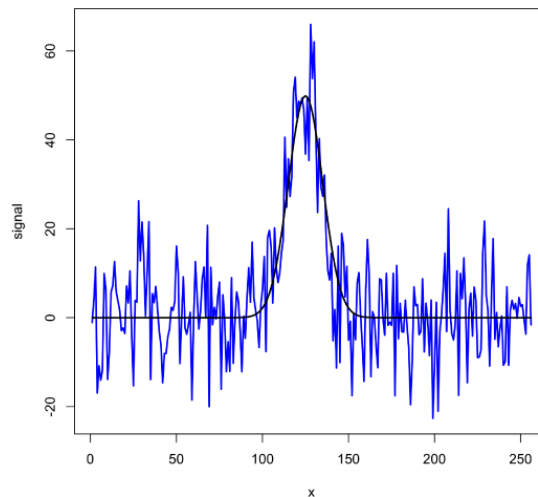


Figure 10.4.1: Example of a noisy signal with a signal-to-noise ratio of 5.1. Figure 10.4.3, Figure 10.4.4, and Figures 10.4.8 show this same figure after applying a seven-point moving average filter, a seven-point Savitzky-Golay smoothing filter, and a Fourier filter.

To estimate the signal-to-noise ratio, we use the maximum of the pure signal and the standard deviation of the noisy signal as determined using 100 points divided evenly between the two ends.

```
s_to_n = max(gaus_signal)/sd(noisy_signal[c(1:50,201:250)])
s_to_n

[1] 5.14663
```

To create a signal superimposed on an exponential background, we use R's `exp()` function to generate 256 points for the background's signal, add that to our pure Gaussian signal, and plot the result.

```
exp_bkgd = 30*exp(-0.01 * x)
plot(x,exp_bkgd,type = "l")
signal_bkgd = gaus_signal + exp_bkgd
plot(x = x, y = signal_bkgd, type = "l", lwd = 2, col = "blue", xlab = "x", ylab = "signal",
ylim = c(0,60))
lines(x = x, y = gaus_signal, lwd = 2, lty = 2)
```

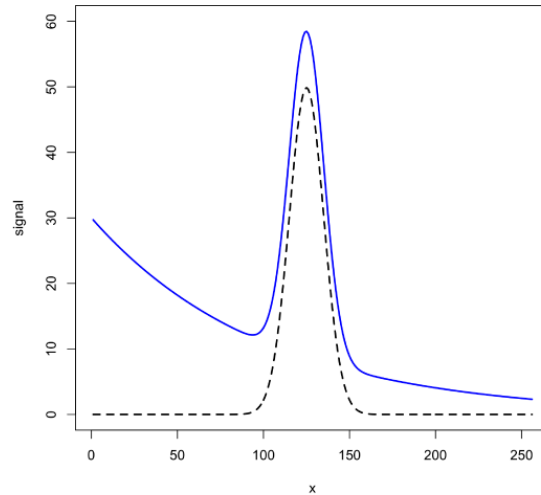


Figure 10.4.2: Example of a pure Gaussian signal superimposed on an exponential background. Figure 10.4.5 shows this same figure after using a seven-point first-derivative Savitzky-Golay filter to remove the background.

Using R's `filter()` Function to Smooth Noise and Remove Background Signals

R's `filter()` function takes the general form

```
filter(x, filter)
```

where `x` is the object being filtered and `filter` is an object that contains the filter's coefficients. To create a seven-point moving average filter, we use the `rep()` function to create a vector that has seven identical values, each equal to $1/7$.

```
mov_avg_7 = rep(1/7, 7)
```

Applying this filter to our noisy signal returns the following result

```
noisy_signal_movavg = filter(noisy_signal, mov_avg_7)
plot(x = x, y = noisy_signal_movavg, type = "l", lwd = 2, col = "blue", xlab = "x", ylab = "signal")
lines(x = x, y = gaus_signal, lwd = 2)
```

with the signal-to-noise ratio improved to

```
s_to_n_movavg = max(gaus_signal)/sd(noisy_signal_movavg[c(1:50,200:250)], na.rm = TRUE)
s_to_n_movavg
[1] 11.29943
```

Note that we must add `na.rm = TRUE` to the `sd()` function because applying a seven-point moving average filter replaces the first three and the last three points with values of `NA` which we must tell the `sd()` function to ignore.

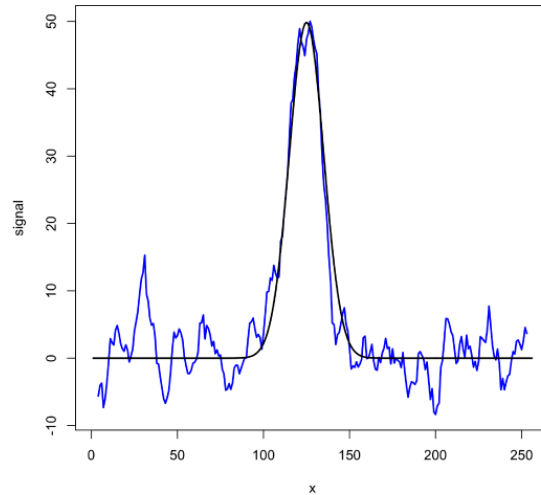


Figure 10.4.3: The result of using R's `filter()` function to apply a seven-point moving average filter to the noisy signal in Figure 10.4.1.

To create a seven-point Savitzky-Golay smoothing filter, we create a vector to store the coefficients, obtaining the values from the original paper (Savitzky, A.; Golay, M. J. E. *Anal Chem*, **1964**, 36, 1627-1639) and then apply it to our noisy signal, obtaining the results below.

```
sg_smooth_7 = c(-2,3,6,7,6,5,-2)/21
noisy_signal_sg = filter(noisy_signal, sg_smooth_7)
plot(x = x, y = noisy_signal_sg, type = "l", lwd = 2, col = "blue", xlab = "x", ylab = "signal")
lines(x = x, y = gaus_signal, lwd = 2)
s_to_n_movavg = max(gaus_signal)/sd(noisy_signal_sg[c(1:50,200:250)], na.rm = TRUE)
s_to_n_movavg

[1] 7.177931
```

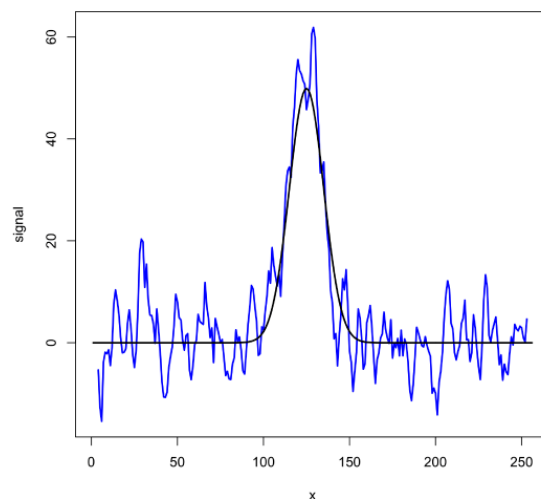


Figure 10.4.4: The result of using R's `filter()` function to apply a seven-point Savitzky-Golay smoothing filter to the noisy signal in Figure 10.4.1.

To remove a background from a signal, we use the same approach, substituting a first-derivative (or higher order) Savitzky-Golay filter.

```
sg_fd_7 = c(22, -67, -58, 0, 58, 67, -22)/252
signal_bkgd_sg = filter(signal_bkgd, sg_fd_7)
plot(x = x, y = signal_bkgd_sg, type = "l", lwd = 2, col = "blue", xlab = "x", ylab = "signal")
```

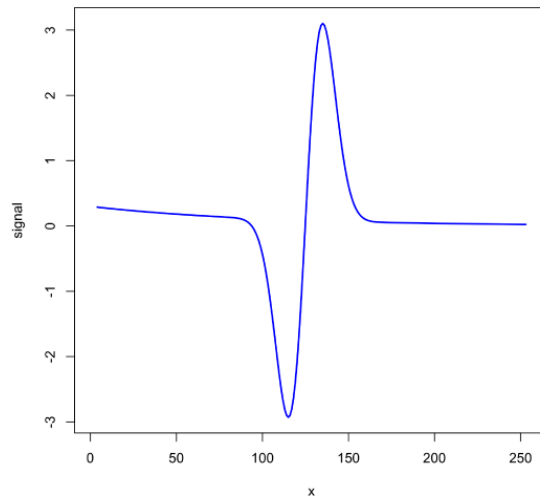


Figure 10.4.5: The result of using R's `filter()` function to apply a seven-point first-derivative Savitzky-Golay filter to the noisy signal in Figure 10.4.1.

Using R's `fft()` Function for Fourier Filtering

To complete a Fourier transform in R we use the `fft()` function, which takes the form `fft(z, inverse = FALSE)` where `z` is the object that contains the values to which we wish to apply the Fourier transform and where setting `inverse = TRUE` allows for an inverse Fourier transform. Before we apply Fourier filtering to our noisy signal, let's first apply the `fft()` function to a vector that contains the integers 1 through 8. First we create a vector to hold our values and then apply the `fft()` function to the vector, obtaining the following results

```
test_vector = seq(1, 8, 1)
test_vector_ft = fft(test_vector)
test_vector_ft

[1] 36+0.000000i -4+9.656854i -4+4.000000i -4+1.656854i -4+0.000000i -4-1.656854i
[7] -4-4.000000i -4-9.656854i
```

Each of the eight results is a complex number with a real and an imaginary component. Note that the real component of the first value is 36, which is the sum of the elements in our test vector. Note, also, the symmetry in the remaining values where the second and eighth values, the third and seventh values, and the fourth and sixth values are identical except for a change in sign for the imaginary component.

Taking the inverse Fourier transform returns the original eight values (note that the imaginary terms are now zero), but each is eight times larger in value than in our original vector.

```
test_vector_iftft = fft(test_vector_ft, inverse = TRUE)
test_vector_iftft
```

```
[1] 8+0i 16-0i 24+0i 32+0i 40+0i 48+0i 56-0i 64+0i
```

To compensate for this, we divide by the length of our vector

```
test_vector_iftft = fft(test_vector_ft, inverse = TRUE)/length(test_vector)
```

```
test_vector_ift
```

```
[1] 1+0i 2-0i 3+0i 4+0i 5+0i 6+0i 7-0i 8+0i
```

which returns our original vector.

With this background in place, let's use R to complete a Fourier filtering of our noisy signal. First, we complete the Fourier transform of the noisy signal and examine the values for the real component, using R's `Re()` function to extract them. Because of the symmetry noted above, we need only look at the first half of the real components ($x = 1$ to $x = 128$).

```
noisy_signal_ft = fft(noisy_signal)
plot(x = x[1:128], y = Re(noisy_signal_ft)[1:128], type = "l", col = "blue", xlab = "", ylab =
"intensity", lwd = 2)
```

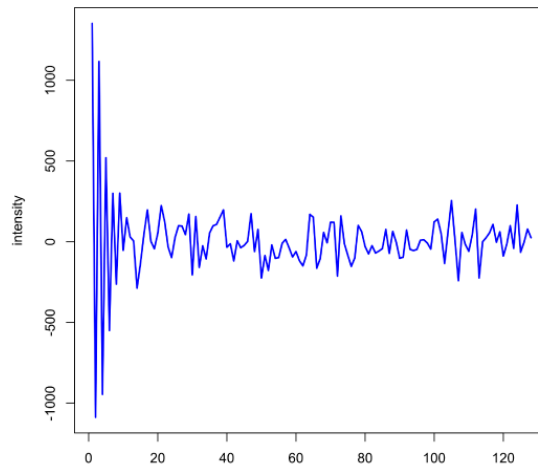


Figure 10.4.6: Plot showing the first 128 real components of the data from Figure 10.4.1 after completing a Fourier transform.

Next, we look for where the signal's magnitude has decayed to what appears to be random noise and set these values to zero. In this example, we retain the first 24 points (and the last 24 points; remember the symmetry noted above) and set both the real and the imaginary components to $0 + 0i$.

```
noisy_signal_ft[25:232] = 0 + 0i
plot(x = x, y = Re(noisy_signal_ft), type = "l", col = "blue", xlab = "", ylab = "intensity",
lwd = 2)
```

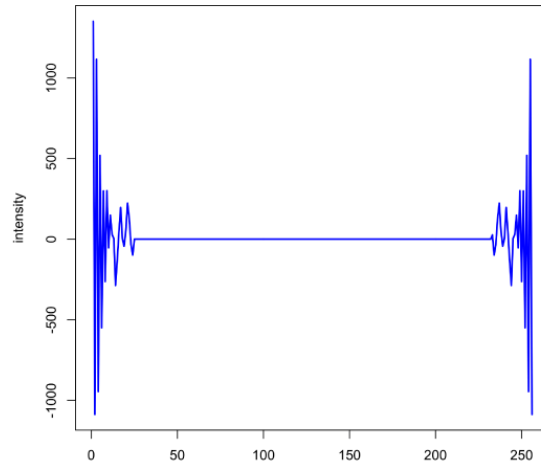


Figure 10.4.7: Plot showing the real components of the data from Figure 10.4.1 after completing a Fourier transform and zeroing out all values other than the first 24 and the last 24 points. Here we are assuming that these points are dominated by the original signal, while the remaining points are mostly from the noise.

Finally, we take the inverse Fourier transform and display the resulting filtered signal and report the signal-to-noise ratio.

```
noisy_signal_ift = fft(noisy_signal_ft, inverse = TRUE)/length(noisy_signal_ft)
plot(x = x, y = Re(noisy_signal_ift), type = "l", col = "blue", xlab = "", ylab = "intensity",
     ylim = c(-20,60), lwd = 3)
lines(x = x,y = gaus_signal,lwd =2, col = "black")

s_to_n = 50/sd(Re(noisy_signal_ift)[c(1:50,200:250)], na.rm = TRUE)
s_to_n

[1] 9.695329
```

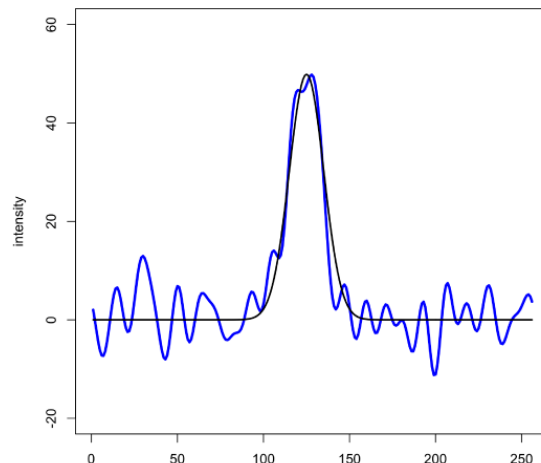


Figure 10.4.8: The final result of Fourier filtering the data from Figure 10.4.1.

This page titled [10.4: Using R to Clean Up Data](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [David Harvey](#).

10.5: Exercises

1. The goal when smoothing data is to improve the signal-to-noise ratio without distorting the underlying signal. The data in the file `problem10_1.csv` consists of four columns of data: the vector x , which contains 200 values for plotting on the x -axis; the vector y , which contains 200 values for a step-function that satisfies the following criteria

$$y = 0 \text{ for } x \leq 75 \text{ and for } x \geq 126$$

$$y = 1 \text{ for } 75 < x < 126$$

the vector n , which contains 200 values drawn from random normal distribution with a mean of 0 and standard deviation of 0.1, and the vector s , which is the sum of y and n . In essence, y is the pure signal, n is the noise, and s is a noisy signal. Using this data, complete the following tasks:

(a) Determine the mean signal, the standard deviation of the noise, and the signal-to-noise ratio for the noisy signal using just the data in the object s .

(b) Explore the effect of applying to the noisy signal, one pass each of moving average filters of widths 5, 7, 9, 11, 13, 15, and 17. For each moving average filter, determine the mean signal, the standard deviation of the noise, and the signal-to-noise ratio. Organize these measurements using a table and comment on your results. Prepare a single plot that displays the original noisy signal and the smoothed signals using widths of 5, 9, 13, and 17, off-setting each so that all five signals are displayed. Comment on your results.

(c) Repeat the calculations in (b) using Savitzky-Golay quadratic/cubic smoothing filters of widths 5, 7, 9, 11, 13, 15, and 17; see the [original paper](#) for each filter's coefficients.

(d) Considering your results for (b) and for (c), what filter and what width provides the greatest improvement in the signal-to-noise ratio with the least distortion of the original signal's step-function? Be sure to justify your choice.

2. The file `problem10_2.csv` consists of two columns, each with 1024 points: x is an index for the x -axis and y is noisy data with a hint of a signal. Show that there is a signal in this file by using any one moving average or Savitzky-Golay smoothing filter of your choice and using a Fourier filter. Present your results in a single figure that shows the original signal, the signal after smoothing, and the signal after Fourier filtering. Comment on your results.

3. The file `problem 10_3.csv` consists of six columns: x is an index for the x -axis and y_1, y_2, y_3, y_4 , and y_5 are signals superimposed on a variable background. Use a Savitzky-Golay nine-point cubic second-derivative filter to remove the background from the data and then build a calibration model using these results, and report the calibration equation and a plot of the calibration curve. See the [original paper](#) for the filter's coefficients.

11: FINDING STRUCTURE IN DATA

One of the more intriguing aspects of chemometrics is the ability to discover and extract information from a large data set that appears, at first glance, to lack any defined order. And yet, it is likely that there are determinate factors that explain the data. Consider a data set that consists of the daily concentration of NOX—the combined amounts of NO₂ and of NO in the air expressed as $\mu\text{g}/\text{m}^3$ —in samples of urban air. Although a plot of the concentration of NOX as a function of time likely appears noisy, we can easily identify variables that might affect the daily measurements:

- temperature: we need more energy on colder days, which increases the use of fuels that generate NOX emissions
- day of the week: perhaps more traffic on work days than on weekends
- atmospheric conditions: strong winds may disperse NOX emissions and stagnation may concentrate NOX emissions
- location of air samplers: samplers at busy intersections may give different results from samplers located in city parks

The chemometric methods introduced in this chapter—cluster analysis, principal component analysis, and multivariate linear regression—provide ways to probe the underlying factors that provide structure to our data.

11.1: What Do We Mean By Structure?

11.2: Cluster Analysis

11.3: Principal Component Analysis

11.4: Multivariate Linear Regression

11.5: Using R for a Cluster Analysis

11.6: Using R for a Principal Component Analysis

11.7: Using R for a Multivariate Linear Regression

11.8: Exercises

11.1: What Do We Mean By Structure?

The signals we measure include contributions from determinate and indeterminate sources, with the determinate components resulting from the analytes in our sample and with the indeterminate sources resulting from noise. When we describe our data as having structure, or that we are looking for structure in our data, our interest is in the determinate contributions to the signal. Consider, for example, the data in the following figure, which shows the visible spectra for 24 samples at 635 wavelengths.

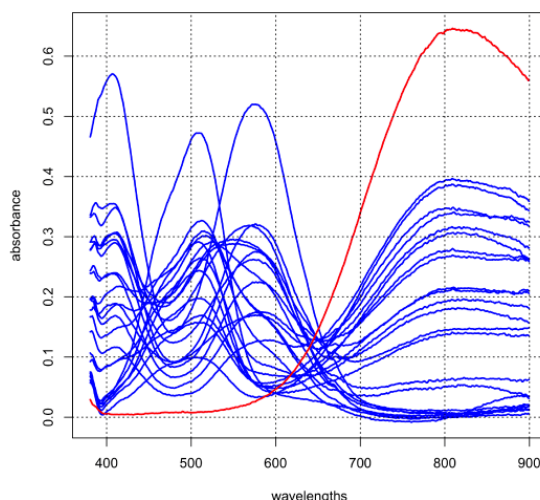


Figure 11.1.1: Visible spectra for 24 samples recorded at 635 wavelengths between 380.5 nm and 889.5 nm. The spectrum in red highlights one of the 24 spectra included in this data set.

Each curve in this figure, such as the one shown in red, is one of the 24 samples that make up this data set and shows the extent to which each of the 635 discrete wavelengths of light are absorbed by that sample: this is the determinate contribution to the data. Looking closely at the spectrum shown in red, we see small variations in the absorbance superimposed on the determinate signal: this is the indeterminate contribution to the data.

Although when first examined, the 24 spectra in Figure 11.1.1 may create a sense of disorder, there is a clear underlying structure to the data. For example, there are four apparent peaks centered at wavelengths around 400 nm, 500 nm, 580 nm, and 800 nm. Each of the individual spectra include one or more of these peaks. Further, at a wavelength of 800 nm, we see that some samples show no absorbance, and presumably lack whatever analyte is responsible for this peak; other samples, however, clearly include contributions from this analyte. This is what we mean by finding structure in data. In this chapter we explore three tools for finding structure in data—cluster analysis, principal component analysis, and multivariate linear regression—that allow us to make sense of that structure.

11.2: Cluster Analysis

In the previous section we examined the spectra of 24 samples at 635 wavelengths, displaying the data by plotting the absorbance as a function of wavelength. Another way to examine the data is to plot the absorbance of each sample at one wavelength against the absorbance of the same sample at a second wavelength, as we see in the following figure using wavelengths of 400.3 nm and 508.7 nm. Note that this plot suggests an underlying structure to our data as the 24 points occupy a triangular-shaped space, defined by the samples identified as 1, 2, and 3.

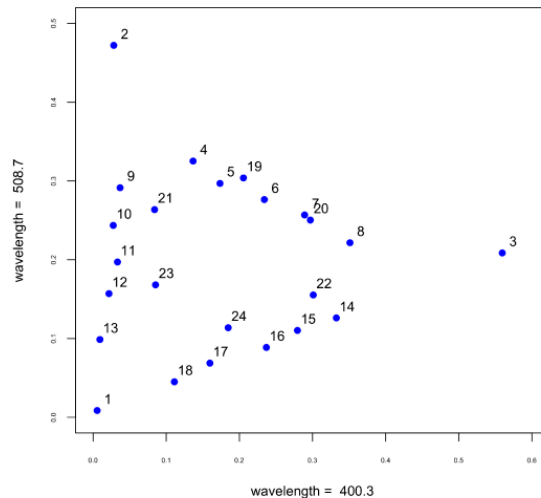


Figure 11.2.1: Plot showing the absorbance values for the 24 samples from Figure 11.1 at wavelengths of 400.3 nm and 508.7 nm. The numbers next to the points are index values for the samples.

We can extend this analysis to three wavelengths, as we see in the following figure, and, to as many as all 635 wavelengths (Of course we cannot examine a plot of this as it exists in 635-dimensional space!).

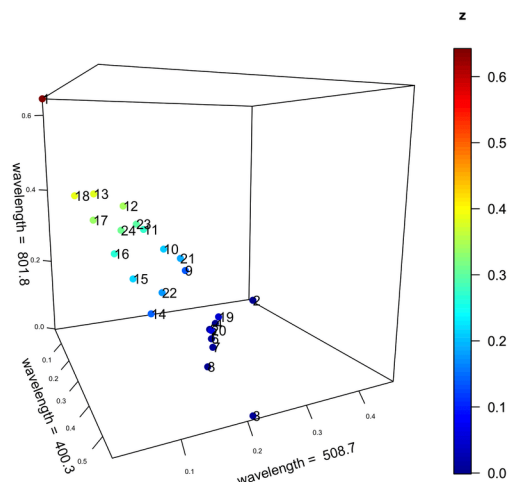


Figure 11.2.2: Plot showing the absorbance values for the 24 samples from Figure 11.1 at wavelengths of 400.3 nm, 508.7 nm, and 801.8 nm. The color of the points shows the absorbance at 801.8 nm, which is the z-axis. The numbers next to the points are index values for the samples. Note that the 24 points here also reside in a triangular-shaped space.

In both Figure 11.2.1 and Figure 11.2.2 (and the higher dimensional plots that we cannot display), some samples are closer to each other in space than are other points. For example, in Figure 11.2.1, samples 7 and 20 are closer to each other than any other pair of samples; samples 2 and 3, however, are further from each other than any other pair of samples.

How Does a Cluster Analysis Work?

A cluster analysis is a way to examine our data in terms of the similarity of the samples to each other. Figure 11.2.3 outlines the steps using a small set of six points defined by two variables, a and b . Panel (a) shows the six data points. The two points closest in distance are 3 and 4, which make the first cluster and which we replace with the red point midway between them, as seen in panel (b). The next two points closest in distance are 2 and 6, which make the second cluster and which we replace with the red point between them, as seen in panel (c). Continuing in this way yields the results in panel (d) where the third cluster brings together points 2, 3, 4, and 6, the fourth cluster brings together points 1, 2, 3, 4, and 6, and the final cluster brings together all six points.

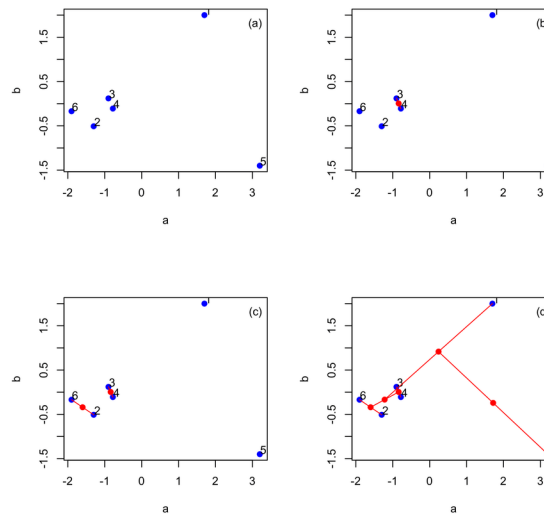


Figure 11.2.3: Example of how cluster analysis works. See the text for details.

To visualize the clusters, in terms of the identify of the points in the clusters, the order in which the clusters form, and the relative similarity of difference between points and clusters, we display the information in Figure 11.2.3d as the dendrogram shown in Figure 11.2.4, which shows, for example, that the clusters of points 3 and 4, and of 2 and 6 are more similar to each other than they are to point 1 and to point 6. The vertical scale, which is identified as Height, provides a measure of the distance of the individual points or clusters of points from each other.

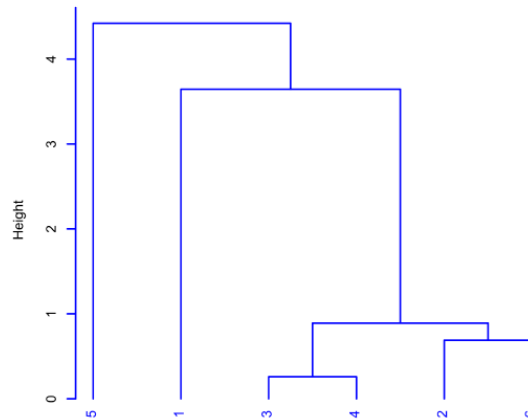


Figure 11.2.4: The results of the cluster analysis in Figure 11.2.3d displayed as a dendrogram.

How Do We Interpret the Results of a Cluster Analysis?

A cluster analysis of the 24 samples from Figure 11.1.1 is shown in Figure 11.2.5 using 40 equally-spaced wavelengths. There is much we can learn from this diagram about the structure of these samples, which we can divide into three distinct clusters of samples, as shown by the boxes. The samples within each cluster are more similar to each other than they are to samples in other clusters. One possible explanation for this structure is that the 24 samples are comprised of three analytes, where, for each cluster, one of the analytes is present at a higher concentration than the other two analytes.

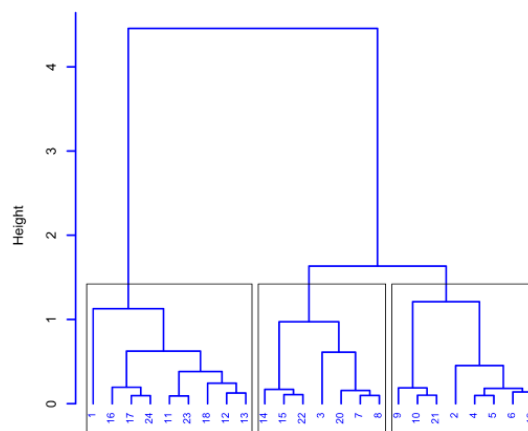


Figure 11.2.5: Cluster analysis of the 24 samples from Figure 11.1.1. The boxes divide the 24 samples into three distinct clusters.

11.3: Principal Component Analysis

The figure below—which is similar in structure to Figure 11.2.2 but with more samples—shows the absorbance values for 80 samples at wavelengths of 400.3 nm, 508.7 nm, and 801.8 nm. Although the axes define the space in which the points appear, the individual points themselves are, with a few exceptions, not aligned with the axes. The cloud of 80 points has a global mean position within this space and a global variance around the global mean (see Chapter 7.3 where we used these terms in the context of an analysis of variance).

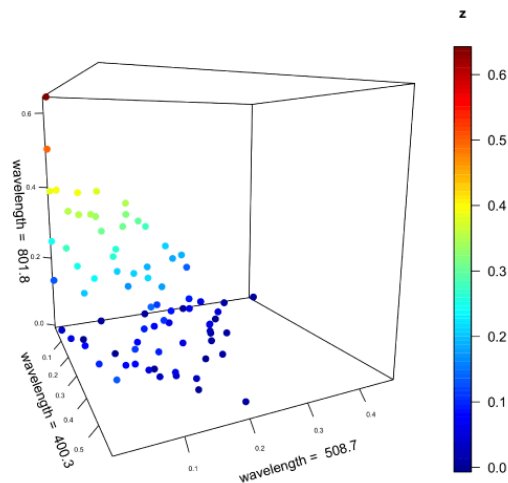


Figure 11.3.1: Scatterplot showing the absorbance values for 80 samples at three wavelengths: 400.3 nm, 508.7 nm, and 801.8 nm. Figure 11.2.2 shows a subset of this data using 24 samples of the 80 samples.

Suppose we leave the points in space as they are and rotate the three axes. We might rotate the three axes until one passes through the cloud in a way that maximizes the variation of the data along that axis, which means this new axis accounts for the greatest contribution to the global variance. Having aligned this primary axis with the data, we then hold it in place and rotate the remaining two axes around the primary axis until one of them passes through the cloud in a way that maximizes the data's remaining variance along that axis; this becomes the secondary axis. Finally, the third, or tertiary axis, is left, which explains whatever variance remains. In essence, this is what comprises a principal component analysis (PCA).

How Does a Principal Component Analysis Work?

One of the challenges with understanding how PCA works is that we cannot visualize our data in more than three dimensions. The data in Figure 11.3.1, for example, consists of spectra for 24 samples recorded at 635 wavelengths. To visualize all of this data requires that we plot it along 635 axes in 635-dimensional space! Let's consider a much simpler system that consists of 21 samples for each of which we measure just two properties that we will call the first variable and the second variable. Figure 11.3.2 shows our data, which we can express as a matrix with 21 rows, one for each of the 21 samples, and 2 columns, one for each of the two variables.

$$[D]_{21 \times 2}$$

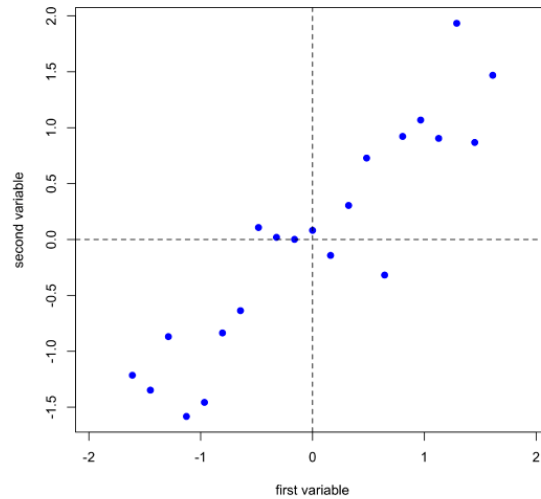


Figure 11.3.2: The scatterplot of our 21 samples as a function of their values for first variable and the second variable.

Next, we complete a linear regression analysis on the data and add the regression line to the plot; we call this the first principal component.

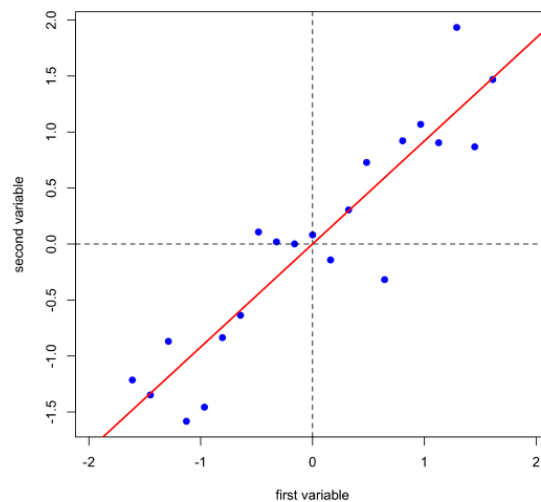


Figure 11.3.3: The data from Figure 11.3.2 showing the regression line that is the first principal component axis.

Projecting our data (the blue points) onto the regression line (the red points) gives the location of each point on the first principal component's axis; these values are called the scores, S . The cosines of the angles between the first principal component's axis and the original axes are called the loadings, L . We can express the relationship between the data, the scores, and the loadings using matrix notation. Note that from the dimensions of the matrices for D , S , and L , each of the 21 samples has a score and each of the two variables has a loading.

$$[D]_{21 \times 2} = [S]_{21 \times 1} \times [L]_{1 \times 2}$$

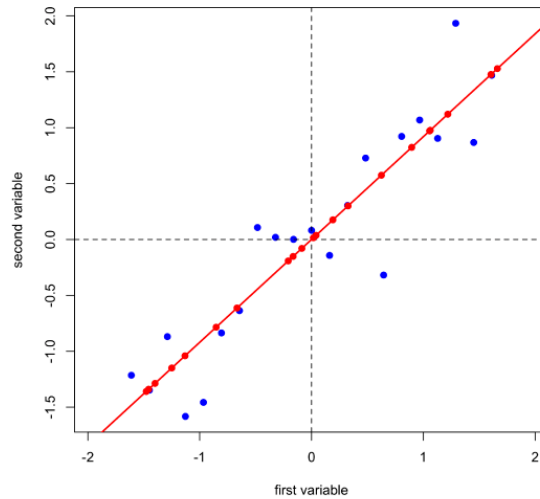


Figure 11.3.4: The projections (red dots) onto the first principal component axis of the original data (blue dots) provide the scores, which are a measure of the distance of the projections from the origin.

Next, we draw a line perpendicular to the first principal component axis, which becomes the second (and last) principal component axis, project the original data onto this axis (points in green) and record the scores and loadings for the second principal component.

$$[D]_{21 \times 2} = [S]_{21 \times 2} \times [L]_{2 \times 2}$$

Note

In matrix multiplication the number of columns in the first matrix must equal the number of rows in the second matrix. The result of matrix multiplication is a new matrix that has a number of rows equal to that of the first matrix and that has a number of columns equal to that of the second matrix; thus multiplying together a matrix that is 5×4 with one that is 4×8 gives a matrix that is 5×8 .

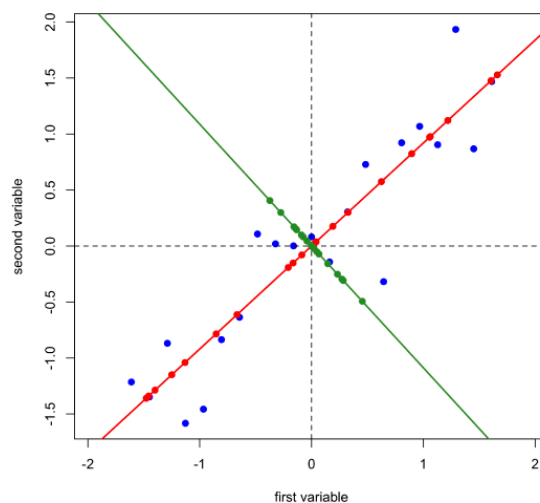


Figure 11.3.5: The projections (green dots) of the original data (blue dots) onto the second, and final, principal component's axis.

If we were working with 21 samples and 10 variables, then we would do this:

1. plot the data for the 21 samples in 10-dimensional space where each variable is an axis
2. find the first principal component's axis and make note of the scores and loadings
3. project the data points for the 21 samples onto the 9-dimensional surface that is perpendicular to the first principal component's axis
4. find the second principal component's axis and make note of the scores and loading
5. project the data points for the 21 samples onto the 8-dimensional surface that is perpendicular to the second (and the first) principal component's axis
6. repeat until all 10 principal components are identified and all scores and loadings reported

How Do We Interpret the Results of a Principal Component Analysis?

The results of a principal component analysis are given by the scores and the loadings. Let's return to the data from Figure 11.3.1, but to make things more manageable, we will work with just 24 of the 80 samples and expand the number of wavelengths from three to 16 (a number that is still a small subset of the 635 wavelengths available to us). The figure below shows the full spectra for these 24 samples and the specific wavelengths we will use as dotted lines; thus, our data is a matrix with 24 rows and 16 columns, $[D]_{24 \times 16}$. A principal component analysis of this data will yield 16 principal component axes.

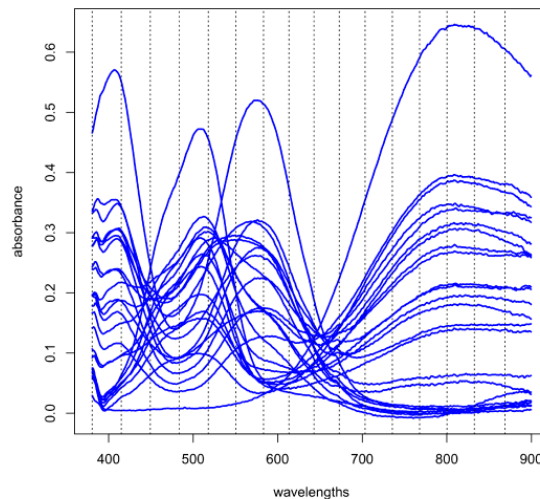


Figure 11.3.6: The spectra for 24 samples. The dotted lines shown the 16 individual wavelengths, which are 380.5 nm, 414.9 nm, 449.3 nm, 483.7 nm, 517.9 nm, 550.6 nm, 538.2 nm, 613.3 nm, 642.9 nm, 672.7 nm, 703.3 nm, 735.5 nm, 767.8 nm, 800.2 nm, 832.6 nm, and 868.6 nm. This is the same data used to illustrate cluster analysis.

Each principal component accounts for a portion of the data's overall variances and each successive principal component accounts for a smaller proportion of the overall variance than did the preceding principal component. Those principal components that account for insignificant proportions of the overall variance presumably represent noise in the data; the remaining principal components presumably are determinate and sufficient to explain the data. The following table provides a summary of the proportion of the overall variance explained by each of the 16 principal components.

Table 11.3.1: The Proportion of Overall Variance Explained by the Principal Components for the Data in Figure 11.3.6.

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
standard deviation	3.3134	2.1901	0.42561	0.17585	0.09384	0.04607	0.04026	0.01253
proportion of variance	0.6862	0.2998	0.01132	0.00193	0.00055	0.00013	0.00010	0.00001
cumulative proportion	0.6862	0.9859	0.99725	0.99919	0.99974	0.99987	0.99997	0.99998
	PC9	PC10	PC11	PC12	PC13	PC14	PC15	PC16
standard deviation	0.01049	0.009211	0.007084	0.004478	0.00416	0.003039	0.002377	0.001504
proportion of variance	0.00001	0.000010	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8
cumulative proportion	0.99999	0.999990	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

The first principal component accounts for 68.62% of the overall variance and the second principal component accounts for 29.98% of the overall variance. Collectively, these two principal components account for 98.59% of the overall variance; adding a third component accounts for more than 99% of the overall variance. Clearly we need to consider at least two components (maybe three) to explain the data in Figure 11.3.1. The remaining 14 (or 13) principal components simply account for noise in the original data. This leaves us with the following equation relating the original data to the scores and loadings

$$[D]_{24 \times 16} = [S]_{24 \times n} \times [L]_{n \times 16}$$

where n is the number of components needed to explain the data, in this case two or three.

To examine the principal components more closely, we plot the scores for PC1 against the scores for PC2 to give the scores plot seen below, which shows the scores occupying a triangular-shaped space.

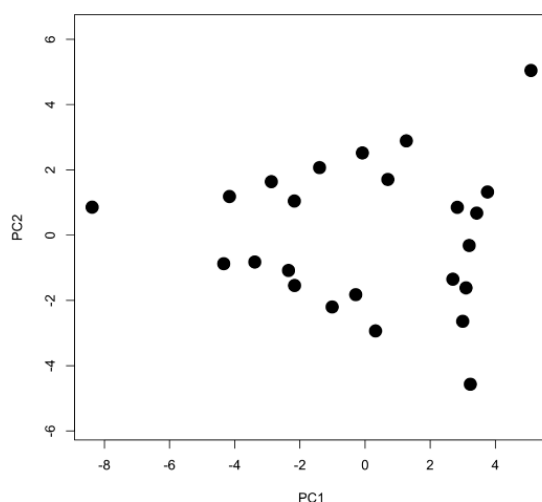


Figure 11.3.7: The scores plot for 24 samples showing their scores along the first principal component's axis and the second principal component's axis.

Because our data are visible spectra, it is useful to compare the equation

$$[D]_{24 \times 16} = [S]_{24 \times n} \times [L]_{n \times 16}$$

to Beer's Law, which in matrix form is

$$[A]_{24 \times 16} = [C]_{24 \times n} \times [\epsilon b]_{n \times 16}$$

where $[A]$ gives the absorbance values for the 24 samples at 16 wavelengths, $[C]$ gives the concentrations of the two or three components that make up the samples, and $[\epsilon b]$ gives the products of the molar absorptivity and the pathlength for each of the two or three components at each of the 16 wavelengths. Comparing these two equations suggests that the scores are related to the concentrations of the n components and that the loadings are related to the molar absorptivities of the n components. Furthermore, we can explain the pattern of the scores in Figure 11.3.7 if each of the 24 samples consists of a 1-3 analytes with the three vertices being samples that contain a single component each, the samples falling more or less on a line between two vertices being binary mixtures of the three analytes, and the remaining points being ternary mixtures of the three analytes.

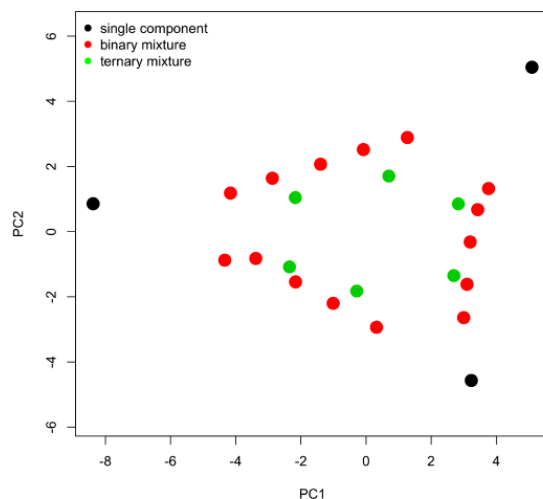


Figure 11.3.8: The scores plot from Figure 11.3.7 color coded to show samples that contain one component, samples that contain two components, and samples that contain three components. Note that the binary mixtures fall along a line (or gently curving arc) that connects two single component samples, and that the ternary mixtures occupy the innermost interior space defined by the single component samples and binary mixtures.

Note

If there are three components in our 24 samples, why are two components sufficient to account for almost 99% of the over variance? Suppose we prepared each sample by using a volumetric digital pipet to combine together aliquots drawn from solutions of the pure components, diluting each to a fixed volume in a 10.00 mL volumetric flask. For example, to make a ternary mixture we might pipet in 5.00 mL of component one and 4.00 mL of component two. If we are diluting to a final volume of 10 mL, then the volume of the third component must be less than 1.00 mL to allow for diluting to the mark. Because the volume of the third component is limited by the volumes of the first two components, two components are sufficient to explain most of the data.

The loadings, as noted above, are related to the molar absorptivities of our sample's components, providing information on the wavelengths of visible light that are most strongly absorbed by each sample. We can overlay a plot of the loadings on our scores plot (this is called a biplot), as shown here.

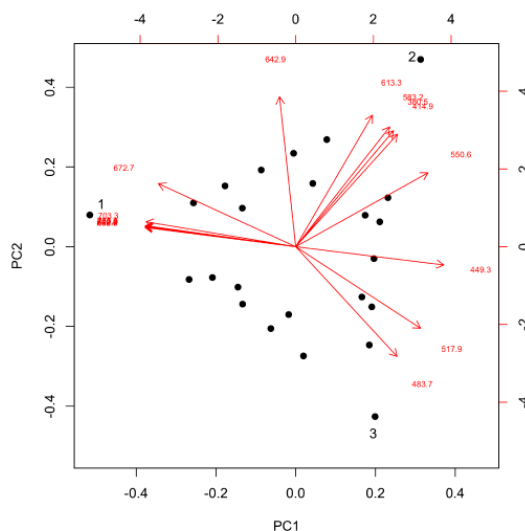


Figure 11.3.9: Biplot showing the scores (dots) and loadings (arrows) for our 24 samples and 16 wavelengths. The position of the loadings relative to the scores provides information about the relationship between the two. For example, the loading for a wavelength of 703.3 nm aligns almost perfectly with the scores for sample 1, suggesting that this wavelength is absorbed by essentially only the single component that makes up this sample. Light with a wavelength of 642.9 nm, however, has a loading that falls in between the scores for samples 1 and 2, suggesting it is absorbed by the single component that makes up sample 1 and the single component that makes up sample 2.

Each arrow is identified with one of our 16 wavelengths and points toward the combination of PC1 and PC2 to which it is most strongly associated. For example, although difficult to read here, all wavelengths from 672.7 nm to 868.7 nm (see the caption for Figure 11.3.6 for a complete list of wavelengths) are strongly associated with the analyte that makes up the single component sample identified by the number one, and the wavelengths of 380.5 nm, 414.9 nm, 583.2 nm, and 613.3 nm are strongly associated with the analyte that makes up the single component sample identified by the number two.

If we have some knowledge about the possible source of the analytes, then we may be able to match the experimental loadings to the analytes. The samples in Figure 11.3.1 were made using solutions of several first row transition metal ions. Figure 11.3.10 shows the visible spectra for four such metal ions. Comparing these spectra with the loadings in Figure 11.3.9 shows that Cu^{2+} absorbs at those wavelengths most associated with sample 1, that Cr^{3+} absorbs at those wavelengths most associated with sample 2, and that Co^{2+} absorbs at wavelengths most associated with sample 3; the last of the metal ions, Ni^{2+} , is not present in the samples

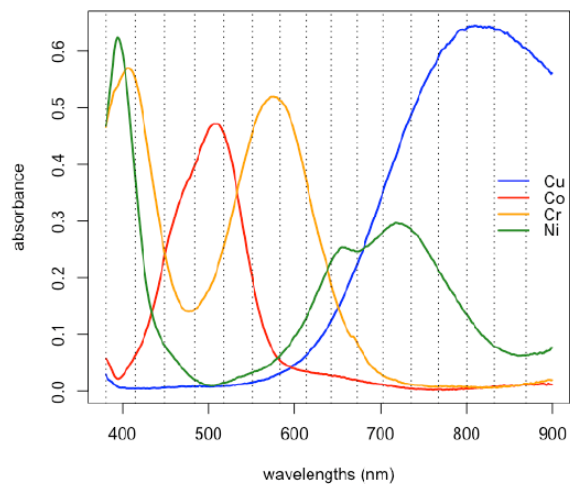


Figure 11.3.10: The visible spectra for the four metal ions that might be in the 24 samples. Of these metal ions, $\text{Ni}^{2+}(\text{aq})$ is not present in the samples.

11.4: Multivariate Linear Regression

In Chapter 11.2 we used a cluster analysis of the spectra for 24 samples measured at 16 wavelengths to show that we could divide the samples into three distinct groups, speculating that the samples contained three analytes and that in each group one of the analytes was present at a concentration greater than that of the other two analytes. In Chapter 11.3 we used a principal component analysis of the same set of samples to suggest that the three analytes are Cu^{2+} , Cr^{3+} , and Co^{2+} . In this section we will use a multivariate linear regression analysis to determine the concentration of these analytes in each of the 24 samples.

How Does a Calibration Using Multivariate Regression Work?

In a simple linear regression analysis, as outlined in Chapter 8, we model the relationship between a single dependent variable, y , and a single independent variable, x , using the equation

$$y = \beta_0 + \beta_1 x$$

where y is a vector of measured responses for the dependent variable, where x is a vector of values for the independent variable, where β_0 is the expected y -intercept, and where β_1 is the expected slope. For example, to complete a Beer's law calibration curve for a single analyte, where A is the absorbance and C is the analyte's concentration

$$A = \epsilon b C$$

we prepare a set of n standard solutions, each with a known concentration of the analyte and measure the absorbance for each of the standard solutions at a single wavelength. A linear regression analysis returns values for ϵb , allowing us to determine the concentration of analyte in a sample by measuring its absorbance. See Chapter 8 for a review of how to complete a linear regression analysis using R.

In a multivariate linear regression we have j dependent variables, Y , and k independent variables, X , and we measure the dependent variable for each of the n values for the independent variables; we can represent this using matrix notation as

$$[Y]_{n \times j} = [X]_{n \times k} \times [\beta]_{k \times j}$$

In this case, to complete a Beer's law calibration curve we prepare a set of n standard solutions, each of which contains known concentrations of the k analytes, and measure the absorbance of each standard at each of the j wavelengths

$$[A]_{n \times j} = [C]_{n \times k} \times [\epsilon b]_{k \times j}$$

where $[A]$ is a matrix of absorbance values, $[C]$ is a matrix of concentrations, and $[\epsilon b]$ is a matrix of ϵb values for each analyte at each wavelength.

Because matrix algebra does not allow for division, we solve for $[\epsilon b]$ by first pre-multiplying both sides of the equation by the transpose of the matrix of concentrations

$$[C]_{k \times n}^T \times [A]_{n \times j} = [C]_{k \times n}^T \times [C]_{n \times k} \times [\epsilon b]_{k \times j}$$

and then pre-multiplying both sides of the equation by $([C]_{k \times n}^T \times [C]_{n \times k})^{-1}$ to give

$$([C]_{k \times n}^T \times [C]_{n \times k})^{-1} \times [C]_{k \times n}^T \times [A]_{n \times j} = ([C]_{k \times n}^T \times [C]_{n \times k})^{-1} \times [C]_{k \times n}^T \times [C]_{n \times k} \times [\epsilon b]_{k \times j}$$

Multiplying $([C]_{k \times n}^T \times [C]_{n \times k})^{-1}$ by $([C]_{k \times n}^T \times [C]_{n \times k})$ is equivalent to multiplying a value by its inverse, which is equal to 1; thus, we have

$$([C]_{k \times n}^T \times [C]_{n \times k})^{-1} \times [C]_{k \times n}^T \times [A]_{n \times j} = [\epsilon b]_{k \times j}$$

With the ϵb matrix in hand, we can determine the concentration of the analytes in a set of samples using the same general approach, as shown here

$$[A]_{n \times j} = [C]_{n \times k} \times [\epsilon b]_{k \times j}$$

$$\begin{aligned}
 [A]_{n \times j} \times [\epsilon b]_{j \times k}^T &= [C]_{n \times k} \times [\epsilon b]_{k \times j} \times [\epsilon b]_{j \times k}^T \\
 [A]_{n \times j} \times [\epsilon b]_{j \times k}^T \times ([\epsilon b]_{k \times j} \times [\epsilon b]_{j \times k}^T)^{-1} &= [C]_{n \times k} \times [\epsilon b]_{k \times j} \times [\epsilon b]_{j \times k}^T \times ([\epsilon b]_{k \times j} \times [\epsilon b]_{j \times k}^T)^{-1} \\
 [A]_{n \times j} \times [\epsilon b]_{j \times k}^T \times ([\epsilon b]_{k \times j} \times [\epsilon b]_{j \times k}^T)^{-1} &= [C]_{n \times k}
 \end{aligned}$$

Note

Completing these calculations by hand is a chore; see Chapter 11.7 to see how you can complete a multivariate linear regression using R.

How Do We Evaluate the Results of a Calibration Using a Multivariate Linear Regression?

One way to evaluate the results of a calibration based on a multivariate linear regression is to use it to examine the values for each analyte's ϵb values from the calibration and compare them to the spectra of the individual analytes; the shape of the two plots should be similar. Another way to evaluate a calibration based on a multivariate regression calibration is to use it to analyze a set of samples with known concentrations of the analytes.

11.5: Using R for a Cluster Analysis

To illustrate how we can use R to complete a cluster analysis: use this link and save the file `allSpec.csv` to your working directory. The data in this file consists of 80 rows and 642 columns. Each row is an independent sample that contains one or more of the following transition metal cations: Cu^{2+} , Co^{2+} , Cr^{3+} , and Ni^{2+} . The first seven columns provide information about the samples:

- a sample id (in the form `custd_1` for a single standard of Cu^{2+} or `nicu_mix1` for a mixture of Ni^{2+} and Cu^{2+})
- a list of the analytes in the sample (in the form `cuco` for a sample that contains Cu^{2+} and Co^{2+})
- the number of analytes in the sample (a number from 1 to 4 and labeled as dimensions)
- the molar concentration of Cu^{2+} in the sample
- the molar concentration of Co^{2+} in the sample
- the molar concentration of Cr^{3+} in the sample
- the molar concentration of Ni^{2+} in the sample

The remaining columns contain absorbance values at 635 wavelengths between 380.5 nm and 899.5 nm.

First, we need to read the data into R, which we do using the `read.csv()` function

```
spec_data <- read.csv("allSpec.csv", check.names = FALSE)
```

where the option `check.names = FALSE` overrides the function's default to not allow a column's name to begin with a number. Next, we will create a subset of this large data set to work with

```
wavelength_ids = seq(8, 642, 40)
sample_ids = c(1, 6, 11, 21:25, 38:53)
cluster_data = spec_data[sample_ids, wavelength_ids ]
```

where `wavelength_ids` is a vector that identifies the 16 equally spaced wavelengths, `sample_ids` is a vector that identifies the 24 samples that contain one or more of the cations Cu^{2+} , Co^{2+} , and Cr^{3+} , and `cluster_data` is a data frame that contains the absorbance values for these 24 samples at these 16 wavelengths.

Before we can complete the cluster analysis, we first must calculate the distance between the $24 \times 16 = 384$ points that make up our data. To do this, we use the `dist()` function, which takes the general form

```
dist(object, method)
```

where `object` is a data frame or matrix with our data. There are a number of options for method, but we will use the default, which is euclidean.

```
cluster_dist = dist(cluster_data, method = "euclidean")
cluster_dist
```

```
1 6 11 21 22 23 24 25
```

```
6 1.53328104
```

```
11 1.73128979 0.96493008
```

```
21 1.48359716 0.24997370 0.77766228
```

```
22 1.49208058 0.32863786 0.68852029 0.09664215
```

```
23 1.49457333 0.42903074 0.57495499 0.21089686 0.11755129
```

```
24 1.51211374 0.52218072 0.47457024 0.31016429 0.21830998 0.10205547
25 1.55862311 0.61154277 0.39798649 0.39406580 0.30194838 0.19121251 0.09771283
38 1.17069314 0.38098750 0.96982420 0.34254297 0.38830178 0.45418483 0.53114050 0.61729900
```

Only a small portion of the values in `cluster_dist` are shown here; each entry shows the distance between two of the 24 samples.

With distances calculated, we can use R's `hclust()` function to complete the cluster analysis. The general form of the function is

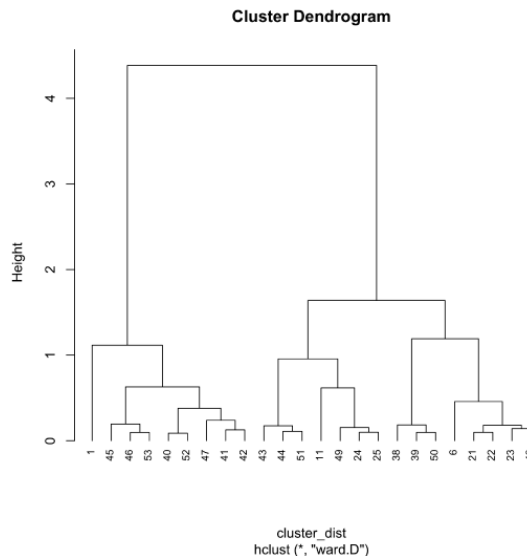
```
hclust(object, method)
```

where `object` is the output created using `dist()` that contains the distances between points. There are a number of options for `method`—here we use the `ward.D` method—saving the output to the object `cluster_results` so that we have access to the results.

```
cluster_results = hclust(cluster_dist, method = "ward.D")
```

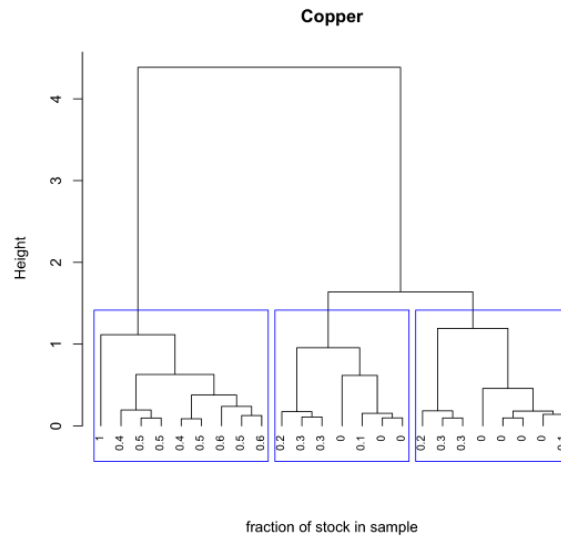
To view the cluster diagram, we pass the object `cluster_results` to the `plot()` function where `hang = -1` extends each vertical line to a height of zero. By default, the labels at the bottom of the dendrogram are the sample ids; `cex` adjusts the size of these labels.

```
plot(cluster_results, hang = -1, cex = 0.75)
```



With a few lines of code we can add useful details to our plot. Here, for example, we determine the the fraction of the stock Cu^{2+} solution in each sample and use these values as labels, and divide the 24 samples into three large clusters using the `rect.clust()` function where `k` is the number of clusters to highlight and `which` indicates which of these clusters to display using a rectangular box.

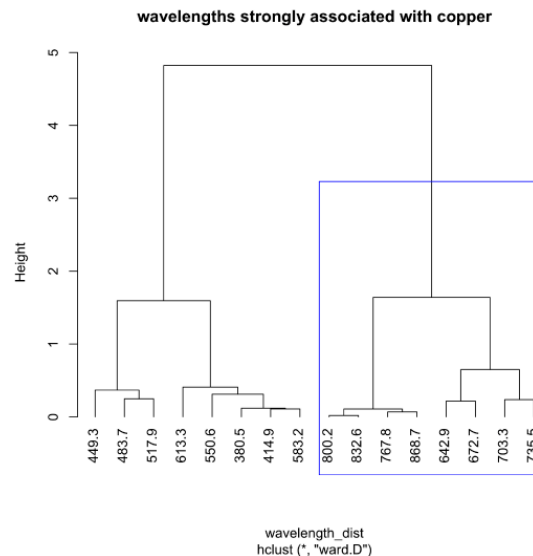
```
cluster_copper = spec_data$concCu/spec_data$concCu[1]
plot(cluster_results, hang = -1, labels = cluster_copper[sample_ids], main = "Copper", xlab =
"fraction of stock in sample", sub = "", cex = 0.75)
rect.hclust(cluster_results, k = 3, which = c(1,2,3), border = "blue")
```



The following code shows how we can use the same data set of 24 samples and 16 wavelength to complete a cluster diagram for the wavelengths. The use of the `t()` function within the `dist()` function takes the transpose of our data so that the rows are the 16 wavelengths and the columns are the 24 samples. We do this because the `dist()` function calculates distances using the rows.

```
wavelength_dist = dist(t(cluster_data))
wavelength_clust = hclust(wavelength_dist, method = "ward.D")
plot(wavelength_clust, hang = -1, main = "wavelengths strongly associated with copper")
rect.hclust(wavelength_clust, k = 2, which = 2, border = "blue")
```

The figure below highlights the cluster of wavelengths most strongly associated with the absorption by Cu^{2+} .



11.6: Using R for a Principal Component Analysis

To illustrate how we can use R to complete a cluster analysis: use this link and save the file `allSpec.csv` to your working directory. The data in this file consists of 80 rows and 642 columns. Each row is an independent sample that contains one or more of the following transition metal cations: Cu^{2+} , Co^{2+} , Cr^{3+} , and Ni^{2+} . The first seven columns provide information about the samples:

- a sample id (in the form `custd_1` for a single standard of Cu^{2+} or `nicu_mix1` for a mixture of Ni^{2+} and Cu^{2+})
- a list of the analytes in the sample (in the form `cuco` for a sample that contains Cu^{2+} and Co^{2+})
- the number of analytes in the sample (a number from 1 to 4 and labeled as dimensions)
- the molar concentration of Cu^{2+} in the sample
- the molar concentration of Co^{2+} in the sample
- the molar concentration of Cr^{3+} in the sample
- the molar concentration of Ni^{2+} in the sample

The remaining columns contain absorbance values at 635 wavelengths between 380.5 nm and 899.5 nm.

First, we need to read the data into R, which we do using the `read.csv()` function

```
spec_data <- read.csv("allSpec.csv", check.names = FALSE)
```

where the option `check.names = FALSE` overrides the function's default to not allow a column's name to begin with a number. Next, we will create a subset of this large data set to work with

```
wavelength_ids = seq(8, 642, 40)
sample_ids = c(1, 6, 11, 21:25, 38:53)
pca_data = spec_data[sample_ids, wavelength_ids ]
```

where `wavelength_ids` is a vector that identifies the 16 equally spaced wavelengths, `sample_ids` is a vector that identifies the 24 samples that contain one or more of the cations Cu^{2+} , Co^{2+} , and Cr^{3+} , and `cluster_data` is a data frame that contains the absorbance values for these 24 samples at these 16 wavelengths.

To complete the principal component analysis we will use R's `prcomp()` function, which takes the general form

```
prcomp(object, center, scale)
```

where `object` is a data frame or matrix that contains our data, and `center` and `scale` are logical values that indicate if we should first center and scale the data before we complete the analysis. When we center and scale our data each variable (in this case, the absorbance at each wavelength) is adjusted so that its mean is zero and its variance is one. This has the effect of placing all variables on a common scale, which ensures that any difference in the relative magnitude of the variables does not affect the principal component analysis.

```
pca_results = prcomp(pca_data, center = TRUE, scale = TRUE)
```

The `prcomp()` function returns a variety of information that we can use to examine the results, including the standard deviation for each principal component, `sdev`, a matrix with the loadings, `rotation`, a matrix with the scores, `x`, and the values used to `center` and `scale` the original data. The `summary()` function, for example, returns the standard deviations for and the proportion of the overall variance explained by each principal component, and the cumulative proportion of variance explained by the principal components.

```
summary(pca_results)
```

Importance of components:

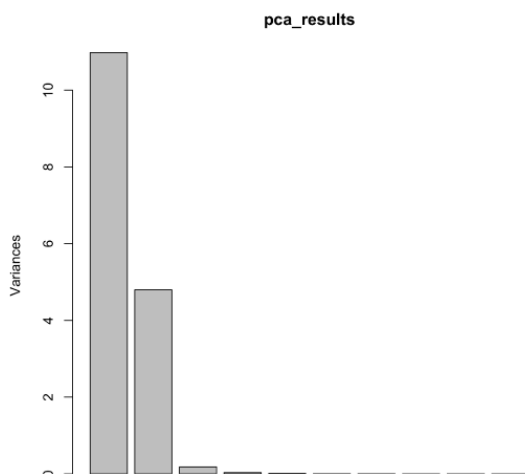
```
PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8 PC9
```

```
Standard deviation 3.3134 2.1901 0.42561 0.17585 0.09384 0.04607 0.04026 0.01253 0.01049
Proportion of Variance 0.6862 0.2998 0.01132 0.00193 0.00055 0.00013 0.00010 0.00001 0.00001
Cumulative Proportion 0.6862 0.9859 0.99725 0.99919 0.99974 0.99987 0.99997 0.99998 0.99999
PC10 PC11 PC12 PC13 PC14 PC15 PC16
```

```
Standard deviation 0.009211 0.007084 0.004478 0.00416 0.003039 0.002377 0.001504
Proportion of Variance 0.000010 0.000000 0.000000 0.00000 0.000000 0.000000 0.000000
Cumulative Proportion 0.999990 1.000000 1.000000 1.00000 1.000000 1.000000 1.000000
```

We can also examine each principal component's variance (the square of its standard deviation) in the form of a bar plot by passing the results of the principal component analysis to the `plot()` function.

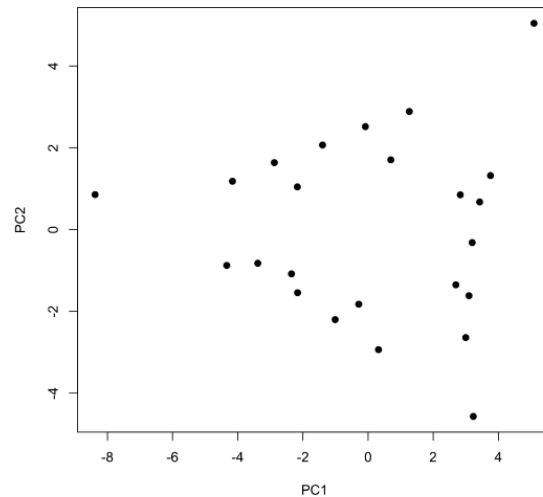
```
plot(pca_results)
```



As noted above, the 24 samples include one, two, or three of the cations Cu^{2+} , Co^{2+} , and Cr^{3+} , which is consistent with our results if individual solutions are made by combining together aliquots of stock solutions of Cu^{2+} , Co^{2+} , and Cr^{3+} and diluting to a common volume. In this case, the volume of stock solution for one cation places limits on the volumes of the other cations such that a three-component mixture essentially has two independent variables.

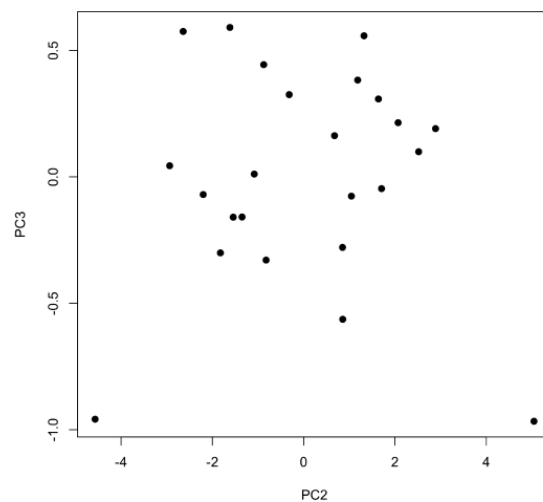
To examine the scores for the principal component analysis, we pass the scores to the `plot()` function, here using `pch = 19` to display them as filled points.

```
plot(pca_results$x, pch = 19)
```



By default, the `plot()` function displays the values for the first two principal components, with the first (PC1) placed on the x-axis and the second (PC2) placed on the y-axis. If we wish to examine other principal components, then we must specify them when calling the `plot()` function; the following command, for example, uses the scores for the second and the third principal components.

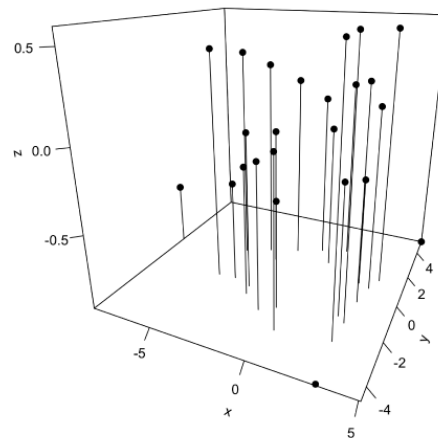
```
plot(x = pca_results$x[,2], y = pca_results$x[,3], pch = 19, xlab = "PC2", ylab = "PC3")
```



If we wish to display the first three principal components using the same plot, then we can use the `scatter3D()` function from the `plot3D` package, which takes the general form

```
library(plot3D)
scatter3D(x = pca_results$x[,1], y = pca_results$x[,2], z = pca_results$x[,3], pch = 19, type = "h", theta = 25, phi = 20, ticktype = "detailed", colvar = NULL)
```

where we use the `library()` function to load the package into our R session (note: this assumes you have installed the `plot3D` package). The option `type = "h"` drops a horizontal line from each point down to the plane for PC1 and PC2, which helps us orient the points in space. By default, the plot uses color to show each points value of the third principal component (displayed on the z-axis); here we set `colvar = NULL` to display all points using the same color.



Although the plots are not shown here, we can use the same commands, replacing `x` with `rotation`, to display the loadings.

```
plot(pca_results$rotation, pch = 19)
```

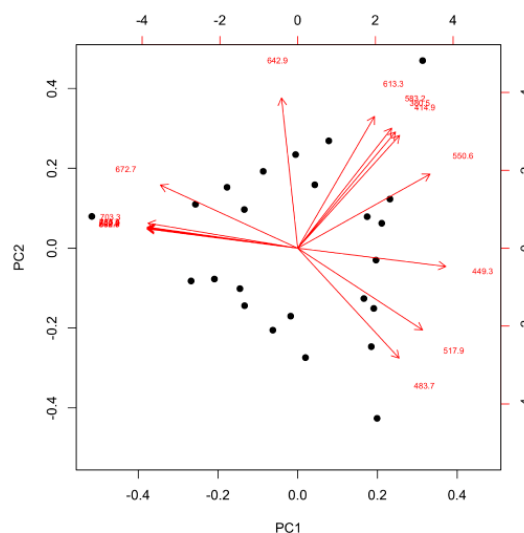
```
plot(x = pca_results$rotation[,2], y = pca_results$rotation[,3], pch = 19, xlab = "PC2", ylab = "PC3")
```

```
scatter3D(x = pca_results$rotation[,1], y = pca_results$rotation[,2], z = pca_results$rotation[,3], pch = 19, type = "h", theta = 25, phi = 20, ticktype = "detailed", colvar = NULL)
```

Another way to view the results of a principal component analysis is to display the scores and the loadings on the same plot, which we can do using the `biplot()` function.

```
biplot(pca_results, cex = c(2, 0.6), xlabs = rep(".", 24))
```

where the option `xlabs = rep(".", 24)` overrides the function's default to display the scores as numbers, replacing them with dots, and `cex = c(2, 0.6)` is used to increase the size of the dots and decrease the size of the labels for the loadings.



In this biplot, the scores are displayed as dots and the loadings are displayed as arrows that begin at the origin and point toward the individual loadings, which are indicated by the wavelengths associated with the loadings. For this set of data, scores and loadings that are co-located with each other represent samples and wavelengths that are strongly correlated with each other. For example, the sample whose score is in the upper right corner is strongly associated with absorbance of light with wavelengths of 613.3 nm, 583.2 nm, 380.5 nm, and 414.9 nm.

Finally, we can use color to highlight features from our data set. For example, the following lines of code creates a scores plot that uses a color pallet to indicate the relative concentration of Cu^{2+} in the sample.

```
cu_palette = colorRampPalette(c("white", "blue"))
cu_color = cu_palette(50)[as.numeric(cut(spec_data$concCu[sample_ids], breaks = 50))]
```

The `colorRampPalette()` function takes a vector of colors—in this case white and blue—and returns a function that we can use to create a palette of colors that runs from pure white to pure blue. We then use this function to create 50 shades of white and blue

```
cu_palette(50)

[1] "#FFFFFF" "#F9F9FF" "#F4F4FF" "#EFEFFF" "#EAEAFF" "#E4E4FF" "#DFDFFF" "#DADAFF"
[9] "#D5D5FF" "#D0D0FF" "#CACAFF" "#C5C5FF" "#C0C0FF" "#BBBBFF" "#B6B6FF" "#B0B0FF"
[17] "#ABABFF" "#A6A6FF" "#A1A1FF" "#9C9CFF" "#9696FF" "#9191FF" "#8C8CFF" "#8787FF"
[25] "#8282FF" "#7C7CFF" "#7777FF" "#7272FF" "#6D6DFF" "#6868FF" "#6262FF" "#5D5DFF"
[33] "#5858FF" "#5353FF" "#4E4EFF" "#4848FF" "#4343FF" "#3E3EFF" "#3939FF" "#3434FF"
[41] "#2E2EFF" "#2929FF" "#2424FF" "#1F1FFF" "#1A1AFF" "#1414FF" "#0F0FFF" "#0A0AFF"
[49] "#0505FF" "#0000FF"
```

where `#FFFFFF` is the hexadecimal code for pure white and `#0000FF` is the hexadecimal code for pure blue. The latter part of this line of code

```
cu_color = cu_palette(50)[as.numeric(cut(spec_data$concCu[sample_ids], breaks = 50))]
```

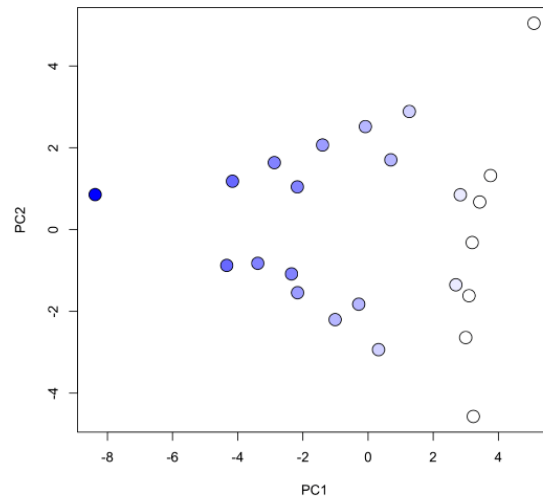
retrieves the concentrations of copper in each of our 24 samples and assigns a hexadecimal code for a shade of blue that indicates the relative concentration of copper in the sample. Here we see that the first sample has a hexadecimal code of `#0000FF` for pure blue, which means this sample has the largest concentration of copper and samples 2–8 have hexadecimal codes of `#FFFFFF` for pure white, which means these samples do not contain any copper.

```
cu_color

[1] "#0000FF" "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FFFFFF" "#FFFFFF"
[9] "#D0D0FF" "#B6B6FF" "#9C9CFF" "#8282FF" "#6868FF" "#D0D0FF" "#B6B6FF" "#9C9CFF"
[17] "#8282FF" "#6868FF" "#EAEAFF" "#EAEAFF" "#B6B6FF" "#B6B6FF" "#8282FF" "#8282FF"
```

Finally, we create the scores plot, using `pch = 21` for an open circle whose background color we designate using `bg = cu_color` and where we use `cex = 2` to increase the size of the points.

```
plot(pca_results$x, pch = 21, bg = cu_color, cex = 2)
```



11.7: Using R for a Multivariate Linear Regression

To illustrate how we can use R to complete a multivariate linear regression, use this link and save the file `allSpec.csv` to your working directory. The data in this file consists of 80 rows and 642 columns. Each row is an independent sample that contains one or more of the following transition metal cations: Cu^{2+} , Co^{2+} , Cr^{3+} , and Ni^{2+} . The first seven columns provide information about the samples:

- a sample id (in the form `custd_1` for a single standard of Cu^{2+} or `nicu_mix1` for a mixture of Ni^{2+} and Cu^{2+})
- a list of the analytes in the sample (in the form `cuco` for a sample that contains Cu^{2+} and Co^{2+})
- the number of analytes in the sample (a number from 1 to 4 and labeled as dimensions)
- the molar concentration of Cu^{2+} in the sample
- the molar concentration of Co^{2+} in the sample
- the molar concentration of Cr^{3+} in the sample
- the molar concentration of Ni^{2+} in the sample

The remaining columns contain absorbance values at 635 wavelengths between 380.5 nm and 899.5 nm. We will use a subset of this data that is identical to that used to illustrate a cluster analysis and a principal component analysis.

First, we need to read the data into R, which we do using the `read.csv()` function

```
spec_data <- read.csv("allSpec.csv", check.names = FALSE)
```

where the option `check.names = FALSE` overrides the function's default to not allow a column's name to begin with a number. Next, we will create objects to hold the concentrations and absorbances for standard solutions of Cu^{2+} , Cr^{3+} , and Co^{2+} , which are the three analytes

```
wavelength_ids = seq(8, 642, 40)
abs_stds = spec_data[1:15, wavelength_ids]
conc_stds = data.frame(spec_data[1:15, 4], spec_data[1:15, 5], spec_data[1:15, 6])
abs_samples = spec_data[c(1, 6, 11, 21:25, 38:53), wavelength_ids]
```

where `wavelength_ids` is a vector that identifies the 16 equally spaced wavelengths, `abs_stds` is a data frame that gives the absorbance values for 15 standard solutions of the three analytes Cu^{2+} , Cr^{3+} , and Co^{2+} at the 16 wavelengths, `conc_stds` is a data frame that contains the concentrations of the three analytes in the 15 standard solutions, and `abs_samples` is a data frame that contains the absorbances of the 24 sample at the 16 wavelengths. This is the same data used to illustrate cluster analysis and principal component analysis.

To solve for the ϵb matrix we will write and source the following function that takes two objects—a data frame of absorbance values and a data frame of concentrations—and returns a matrix of ϵb values.

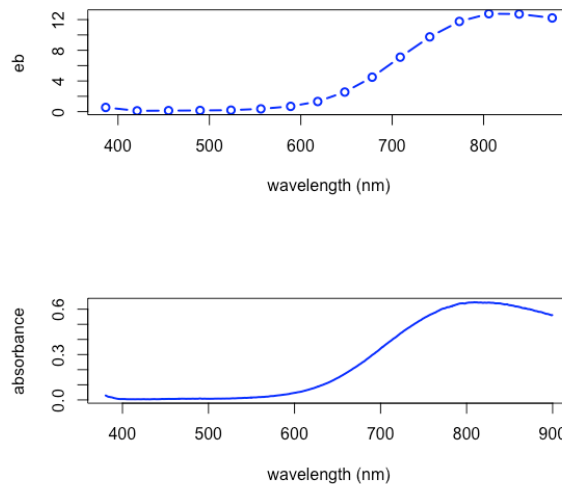
```
findeb = function(abs, conc){
  abs.m = as.matrix(abs)
  conc.m = as.matrix(conc)
  ct = t(conc.m)
  ctc = ct %*% conc.m
  invctc = solve(ctc)
  eb = invctc %*% ct %*% abs.m
  output = eb
  invisible(output)
}
```

Passing `abs_stds` and `conc_stds` to the function

```
eb_pred = findeb(abs_stds, conc_stds)
```

returns the predicted values for ϵb that make up our calibration. As we see below, a plot of the ϵb values for Cu^{2+} has the same shape as a plot of the absorbance values for one of the Cu^{2+} standards.

```
wavelengths = as.numeric(colnames(spec_data[8:642]))
old.par = par(mfrow = c(2,1))
plot(x = wavelengths[wavelength_ids], y = eb_pred[1,], type = "b",
     xlab = "wavelength (nm)", ylab = "eb", lwd = 2, col = "blue")
plot(x = wavelengths, y = spec_data[1,8:642], type = "l",
     xlab = "wavelength (nm)", ylab = "absorbance", lwd = 2, col = "blue")
par(old.par)
```



Having completed the calibration, we can determine the concentrations of the three analytes in the 24 samples using the following function, which takes as inputs the data frame of absorbance values and the ϵb matrix returned by the function `findeb`

```
findconc = function(abs, eb){
  abs.m = as.matrix(abs)
  eb.m = as.matrix(eb)
  ebt = t(eb.m)
  ebebt = eb %*% ebt
  invebebt = solve(ebebt)
  pred_conc = round(abs.m %*% ebt %*% invebebt, digits = 5)
  output = pred_conc
  invisible(output)
}
pred_conc = findconc(abs_samples, eb_pred)
```

To determine the error in the predicted concentrations, we first extract the actual concentrations from the original data set as a data frame, adjusting the column names for clarity.

```
real_conc = data.frame(spec_data[c(1, 6, 11, 21:25, 38:53), 4],
  spec_data[c(1, 6, 11, 21:25, 38:53), 5],
  spec_data[c(1, 6, 11, 21:25, 38:53), 6])
colnames(real_conc) = c("copper", "cobalt", "chromium")
```

and determine the difference between the actual concentrations and the predicted concentrations

```
conc_error = real_conc - pred_conc
```

Finally, we can report the mean error, the standard deviation, and the 95% confidence interval for each analyte.

```
means = apply(conc_error, 2, mean)
round(means, digits = 6)

copper cobalt chromium
-0.000280 -0.000153 -0.000210

sds = apply(conc_error, 2, sd)
round(sds, digits = 6)

copper cobalt chromium
0.001037 0.000811 0.000688

conf.it = abs(qt(0.05/2, 20)) * sds
round(conf.it, digits = 6)

copper cobalt chromium
0.002163 0.001693 0.001434
```

Compared to the ranges of concentrations for the three analytes in the 24 samples

```
range(real_conc$copper)
[1] 0.00 0.05

range(real_conc$cobalt)
[1] 0.0 0.1

range(real_conc$chromium)
[1] 0.0000 0.0375
```

the mean errors and confidence intervals are sufficiently small that we have confidence in the results.

11.8: Exercises

The file [rare_earth.csv](#) contains data for the 17 rare earth elements, which consists of the lanthanides (La → Lu) plus Sc and Y. The data is from Horovitz, O.; Sărbu, C. "Characterization and Classification of Lanthanides by Multivariate-Analysis Methods," *J. Chem. Educ.* **2005**, 82, 473-483. Each row in the file contains data for one element; the columns in the file provide values for the following 16 properties:

- mass: atomic mass (g/mol)
- density: (g/cm^3)
- radius: atomic radius (pm)
- en: electronegativity (Pauling scale)
- ionenergy_1: first ionization energy (kJ/mol)
- ionenergy_2: second ionization energy (kJ/mol)
- ionenergy_3: third ionization energy (kJ/mol)
- mp: melting point (K)
- bp: boiling point (K)
- h_fusion: enthalpy of fusion (kJ/mol)
- h_atom: enthalpy of atomization (kJ/mol)
- entropy: absolute entropy ($\text{J}/\text{mol}\cdot\text{K}$)
- sp_heat: specific heat ($\text{J}/\text{g}\cdot\text{K}$)
- resist: electrical resistivity ($\mu\Omega\text{ cm}$)
- head_cond: heat conductivity ($\text{W cm}^{-1}\text{K}^{-1}$)
- gibbs: Gibbs free energy of formation (kJ/mol)

Two variables included in the original paper—the enthalpy of vaporization and the surface tension at the melting point—are omitted from this data set as they include missing values. Problems 1-3 draw upon the data in this file.

1. Perform a cluster analysis for the 17 elements in the file [rare_earth.csv](#) and comment on the results paying particular attention to the positions of Sc and Y, and the 15 lanthanides. You may wish to compare your results with those reported in the paper cited above.

2. Perform a cluster analysis for the 16 properties in the file [rare_earth.csv](#) and comment on the results. You may wish to compare your results with those reported in the paper cited above.

3. Complete a principal component analysis for the 17 elements in the file [rare_earth.csv](#). Create two-dimensional scores plots that compare PC1 to PC2, PC1 to PC3, and PC2 to PC3, and a three-dimensional scores plot for the first three principal components. Comment on your results paying particular attention to the positions of Sc and Y, and the 15 lanthanides. You may wish to compare your results to those from Exercise 11.1 and the results reported in the paper cited above. Create two-dimensional loadings plots that compare PC1 to PC2, PC1 to PC3, and PC2 to PC3, and a three-dimensional loadings plot for the first three principal components. Comment on your results. You may wish to compare your results to those from Exercise 11.2 and the results reported in the paper cited above.

4. The files [mvr_abs](#) and [mvr_conc](#) contain absorbance values for 10 samples that contain one or more the analytes Co^{2+} , Cu^{2+} , and Ni^{2+} at five wavelengths, and the mM concentrations of the same analytes in the 10 samples. The data are from Dado, G.; Rosenthal, J. "Simultaneous Determination of Cobalt, Copper, and Nickel by Multivariate Linear Regression," *J. Chem. Educ.* **1990**, 67, 797-800. Use the first seven samples as calibration standards and use a multivariate linear regression to determine the concentrations of the analytes in the last three samples. You may wish to compare your results with those reported in the paper cited above.

This page titled [11.8: Exercises](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [David Harvey](#).

12: APPENDICES

12.1: Single-Sided Normal Distribution

12.2: Critical Values for t-Test

12.3: Critical Values for F-Test

12.4: Critical Values for Dixon's Q-Test

12.5: Critical Values for Grubb's Test

12.6: Critical Values for the Wilcoxon Signed Rank Test

12.7: Critical Values for the Wilcoxon Ranked Sum Test

12.1: Single-Sided Normal Distribution

Table 12.1.1, at the bottom of this appendix, gives the proportion, P , of the area under a normal distribution curve that lies to the right of a deviation, z

$$z = \frac{X - \mu}{\sigma}$$

where X is the value for which the deviation is defined, μ is the distribution's mean value and σ is the distribution's standard deviation. For example, the proportion of the area under a normal distribution to the right of a deviation of 0.04 is 0.4840 (see entry in red in the table), or 48.40% of the total area (see the area shaded blue in Figure 12.1.1). The proportion of the area to the left of the deviation is $1 - P$. For a deviation of 0.04 , this is $1 - 0.4840$, or 51.60% .

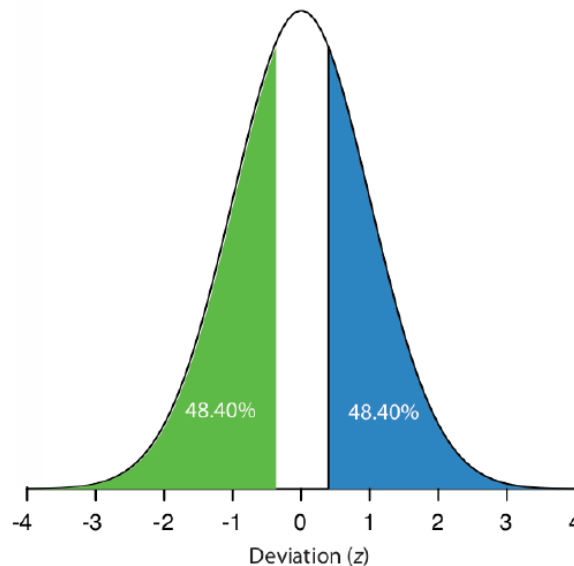


Figure 12.1.1. Normal distribution curve showing the area under a curve greater than a deviation of $+0.04$ (blue) and with a deviation less than -0.04 (green).

When the deviation is negative—that is, when X is smaller than μ —the value of z is negative. In this case, the values in the table give the area to the left of z . For example, if z is -0.04 , then 48.40% of the area lies to the left of the deviation (see area shaded green in Figure 12.1.1).

To use the single-sided normal distribution table, sketch the normal distribution curve for your problem and shade the area that corresponds to your answer (for example, see Figure 12.1.2, which is for Example 4.4.2).

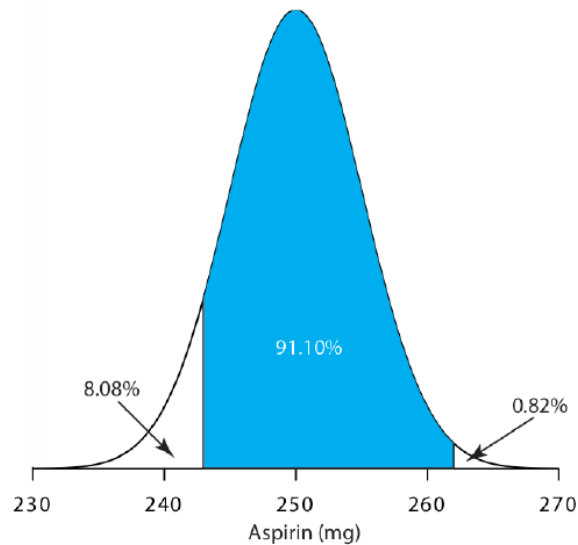


Figure 12.1.2. Normal distribution for the population of aspirin tablets in Example 4.4.2. The population's mean and standard deviation are 250 mg and 5 mg, respectively. The shaded area shows the percentage of tablets containing between 243 mg and 262 mg of aspirin.

This divides the normal distribution curve into three regions: the area that corresponds to our answer (shown in blue), the area to the right of this, and the area to the left of this. Calculate the values of z for the limits of the area that corresponds to your answer. Use the table to find the areas to the right and to the left of these deviations. Subtract these values from 100% and, voilà, you have your answer.

Table 12.1.1: Values for a Single-Sided Normal Distribution

z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.4960	0.4920	0.4880	0.4840	0.4801	0.4761	0.4721	0.4681	0.4641
0.1	0.4602	0.4562	0.4522	0.4483	0.4443	0.4404	0.4365	0.4325	0.4286	0.4247
0.2	0.4207	0.4168	0.4129	0.4090	0.4050	0.4013	0.3974	0.3936	0.3897	0.3859
0.3	0.3821	0.3783	0.3745	0.3707	0.3669	0.3632	0.3594	0.3557	0.3520	0.3483
0.4	0.3446	0.3409	0.3372	0.3336	0.3300	0.3264	0.3228	0.3192	0.3156	0.3121
0.5	0.3085	0.3050	0.3015	0.2981	0.2946	0.2912	0.2877	0.2843	0.2810	0.2776
0.6	0.2743	0.2709	0.2676	0.2643	0.2611	0.2578	0.2546	0.2514	0.2483	0.2451
0.7	0.2420	0.2389	0.2358	0.2327	0.2296	0.2266	0.2236	0.2206	0.2177	0.2148
0.8	0.2119	0.2090	0.2061	0.2033	0.2005	0.1977	0.1949	0.1922	0.1894	0.1867
0.9	0.1841	0.1814	0.1788	0.1762	0.1736	0.1711	0.1685	0.1660	0.1635	0.1611
1.0	0.1587	0.1562	0.1539	0.1515	0.1492	0.1469	0.1446	0.1423	0.1401	0.1379
1.1	0.1357	0.1335	0.1314	0.1292	0.1271	0.1251	0.1230	0.1210	0.1190	0.1170
1.2	0.1151	0.1131	0.1112	0.1093	0.1075	0.1056	0.1038	0.1020	0.1003	0.0985
1.3	0.0968	0.0951	0.0934	0.0918	0.0901	0.0885	0.0869	0.0853	0.0838	0.0823
1.4	0.0808	0.0793	0.0778	0.0764	0.0749	0.0735	0.0721	0.0708	0.0694	0.0681
1.5	0.0668	0.0655	0.0643	0.0630	0.0618	0.0606	0.0594	0.0582	0.0571	0.0559
1.6	0.0548	0.0537	0.0526	0.0516	0.0505	0.0495	0.0485	0.0475	0.0465	0.0455
1.7	0.0466	0.0436	0.0427	0.0418	0.0409	0.0401	0.0392	0.0384	0.0375	0.0367
1.8	0.0359	0.0351	0.0344	0.0336	0.0329	0.0322	0.0314	0.0307	0.0301	0.0294

1.9	0.0287	0.0281	0.0274	0.0268	0.0262	0.0256	0.0250	0.0244	0.0239	0.0233
2.0	0.0228	0.0222	0.0217	0.0212	0.0207	0.0202	0.0197	0.0192	0.0188	0.0183
2.1	0.0179	0.0174	0.0170	0.0166	0.0162	0.0158	0.0154	0.0150	0.0146	0.0143
2.2	0.0139	0.0136	0.0132	0.0129	0.0125	0.0122	0.0119	0.0116	0.0113	0.0110
2.3	0.0107	0.0104	0.0102		0.00964		0.00914		0.00866	
2.4	0.00820		0.00776		0.00734		0.00695		0.00657	
2.5	0.00621		0.00587		0.00554		0.00523		0.00494	
2.6	0.00466		0.00440		0.00415		0.00391		0.00368	
2.7	0.00347		0.00326		0.00307		0.00289		0.00272	
2.8	0.00256		0.00240		0.00226		0.00212		0.00199	
2.9	0.00187		0.00175		0.00164		0.00154		0.00144	
3.0	0.00135									
3.1	0.000968									
3.2	0.000687									
3.3	0.000483									
3.4	0.000337									
3.5	0.000233									
3.6	0.000159									
3.7	0.000108									
3.8	0.0000723									
3.9	0.0000481									
4.0	0.0000317									

12.2: Critical Values for t-Test

Assuming we have calculated t_{exp} , there are two approaches to interpreting a t -test. In the first approach we choose a value of α for rejecting the null hypothesis and read the value of $t(\alpha, \nu)$ from the table below. If $t_{\text{exp}} > t(\alpha, \nu)$, we reject the null hypothesis and accept the alternative hypothesis. In the second approach, we find the row in the table below that corresponds to the available degrees of freedom and move across the row to find (or estimate) the α that corresponds to $t_{\text{exp}} = t(\alpha, \nu)$; this establishes largest value of α for which we can retain the null hypothesis. Finding, for example, that α is 0.10 means that we retain the null hypothesis at the 90% confidence level, but reject it at the 89% confidence level. The examples in this textbook use the first approach.

Table 12.2.1: Critical Values of t for the t -Test

Values of t for...				
...a confidence interval of:	90%	95%	98%	99%
...an α value of:	0.10	0.05	0.02	0.01
Degrees of Freedom				
1	6.314	12.706	31.821	63.657
2	2.920	4.303	6.965	9.925
3	2.353	3.182	4.541	5.841
4	2.132	2.776	3.747	4.604
5	2.015	2.571	3.365	4.032
6	1.943	2.447	3.143	3.707
7	1.895	2.365	2.998	3.499
8	1.860	2.306	2.896	3.255
9	1.833	2.262	2.821	3.250
10	1.812	2.228	2.764	3.169
12	1.782	2.179	2.681	3.055
14	1.761	2.145	2.624	2.977
16	1.746	2.120	2.583	2.921
18	1.734	2.101	2.552	2.878
20	1.725	2.086	2.528	2.845
30	1.697	2.042	2.457	2.750
50	1.676	2.009	2.311	2.678
∞	1.645	1.960	2.326	2.576

The values in this table are for a two-tailed t -test. For a one-tailed test, divide the α values by 2. For example, the last column has an α value of 0.005 and a confidence interval of 99.5% when conducting a one-tailed t -test.

12.3: Critical Values for F-Test

The following tables provide values for $F(0.05, \nu_{\text{num}}, \nu_{\text{denom}})$ for one-tailed and for two-tailed F -tests. To use these tables, we first decide whether the situation calls for a one-tailed or a two-tailed analysis and calculate F_{exp}

$$F_{\text{exp}} = \frac{s_A^2}{s_B^2}$$

where S_A^2 is greater than s_B^2 . Next, we compare F_{exp} to $F(0.05, \nu_{\text{num}}, \nu_{\text{denom}})$ and reject the null hypothesis if $F_{\text{exp}} > F(0.05, \nu_{\text{num}}, \nu_{\text{denom}})$. You may replace s with σ if you know the population's standard deviation.

Table 12.3.1: Critical Values of F for a One-Tailed F -Test

$\frac{\nu_{\text{num}} \rightarrow}{\nu_{\text{denom}} \downarrow}$	1	2	3	4	5	6	7	8	9	10	15	20	∞
1	161.4	199.5	215.7	224.6	230.2	234.0	236.8	238.9	240.5	241.9	245.9	248.0	254.3
2	18.51	19.00	19.16	19.25	19.30	19.33	19.35	19.37	19.38	19.40	19.43	19.45	19.50
3	10.13	9.552	9.277	9.117	9.013	8.941	8.887	8.845	8.812	8.786	8.703	8.660	8.526
4	7.709	6.994	6.591	6.388	6.256	6.163	6.094	6.041	5.999	5.964	5.858	5.803	5.628
5	6.608	5.786	5.409	5.192	5.050	4.950	4.876	4.818	4.722	4.753	4.619	4.558	4.365
6	5.987	5.143	4.757	4.534	4.387	4.284	4.207	4.147	4.099	4.060	3.938	3.874	3.669
7	5.591	4.737	4.347	4.120	3.972	3.866	3.787	3.726	3.677	3.637	3.511	3.445	3.230
8	5.318	4.459	4.066	3.838	3.687	3.581	3.500	3.438	3.388	3.347	3.218	3.150	2.928
9	5.117	4.256	3.863	3.633	3.482	3.374	3.293	3.230	3.179	3.137	3.006	2.936	2.707
10	4.965	4.103	3.708	3.478	3.326	3.217	3.135	3.072	3.020	2.978	2.845	2.774	2.538
11	4.844	3.982	3.587	3.257	3.204	3.095	3.012	2.948	2.896	2.854	2.719	2.646	2.404
12	4.747	3.885	3.490	3.259	3.106	2.996	2.913	2.849	2.796	2.753	2.617	2.544	2.296
13	4.667	3.806	3.411	3.179	3.025	2.915	2.832	2.767	2.714	2.671	2.533	2.459	2.206
14	4.600	3.739	3.344	3.112	2.958	2.848	2.764	2.699	2.646	2.602	2.463	2.388	2.131
15	4.534	3.682	3.287	3.056	2.901	2.790	2.707	2.641	2.588	2.544	2.403	2.328	2.066
16	4.494	3.634	3.239	3.007	2.852	2.741	2.657	2.591	2.538	2.494	2.352	2.276	2.010
17	4.451	3.592	3.197	2.965	2.810	2.699	2.614	2.548	2.494	2.450	2.308	2.230	1.960
18	4.414	3.555	3.160	2.928	2.773	2.661	2.577	2.510	2.456	2.412	2.269	2.191	1.917
19	4.381	3.522	3.127	2.895	2.740	2.628	2.544	2.477	2.423	2.378	2.234	2.155	1.878
20	4.351	3.493	3.098	2.866	2.711	2.599	2.514	2.447	2.393	2.348	2.203	2.124	1.843
∞	3.842	2.996	2.605	2.372	2.214	2.099	2.010	1.938	1.880	1.831	1.666	1.570	1.000

Table 12.3.2: Critical Values of F for a Two-Tailed F -Test

$\frac{\nu_{\text{num}} \rightarrow}{\nu_{\text{denom}} \downarrow}$	1	2	3	4	5	6	7	8	9	10	15	20	∞
1	647.8	799.5	864.2	899.6	921.8	937.1	948.2	956.7	963.3	968.6	984.9	993.1	1018
2	38.51	39.00	39.17	39.25	39.30	39.33	39.36	39.37	39.39	39.40	39.43	39.45	39.50
3	17.44	16.04	15.44	15.10	14.88	14.73	14.62	14.54	14.47	14.42	14.25	14.17	13.90
4	12.22	10.65	9.979	9.605	9.364	9.197	9.074	8.980	8.905	8.844	8.657	8.560	8.257

5	10.01	8.434	7.764	7.388	7.146	6.978	6.853	6.757	6.681	6.619	6.428	6.329	6.015
6	8.813	7.260	6.599	6.227	5.988	5.820	5.695	5.600	5.523	5.461	5.269	5.168	4.894
7	8.073	6.542	5.890	5.523	5.285	5.119	4.995	4.899	4.823	4.761	4.568	4.467	4.142
8	7.571	6.059	5.416	5.053	4.817	4.652	4.529	4.433	4.357	4.259	4.101	3.999	3.670
9	7.209	5.715	5.078	4.718	4.484	4.320	4.197	4.102	4.026	3.964	3.769	3.667	3.333
10	6.937	5.456	4.826	4.468	4.236	4.072	3.950	3.855	3.779	3.717	3.522	3.419	3.080
11	6.724	5.256	4.630	4.275	4.044	3.881	3.759	3.644	3.588	3.526	3.330	3.226	2.883
12	6.544	5.096	4.474	4.121	3.891	3.728	3.607	3.512	3.436	3.374	3.177	3.073	2.725
13	6.414	4.965	4.347	3.996	3.767	3.604	3.483	3.388	3.312	3.250	3.053	2.948	2.596
14	6.298	4.857	4.242	3.892	3.663	3.501	3.380	3.285	3.209	3.147	2.949	2.844	2.487
15	6.200	4.765	4.153	3.804	3.576	3.415	3.293	3.199	3.123	3.060	2.862	2.756	2.395
16	6.115	4.687	4.077	3.729	3.502	3.341	3.219	3.125	3.049	2.986	2.788	2.681	2.316
17	6.042	4.619	4.011	3.665	3.438	3.277	3.156	3.061	2.985	2.922	2.723	2.616	2.247
18	5.978	4.560	3.954	3.608	3.382	3.221	3.100	3.005	2.929	2.866	2.667	2.559	2.187
19	5.922	4.508	3.903	3.559	3.333	3.172	3.051	2.956	2.880	2.817	2.617	2.509	2.133
20	5.871	4.461	3.859	3.515	3.289	3.128	3.007	2.913	2.837	2.774	2.573	2.464	2.085
∞	5.024	3.689	3.116	2.786	2.567	2.408	2.288	2.192	2.114	2.048	1.833	1.708	1.000

12.4: Critical Values for Dixon's Q-Test

The following table provides critical values for $Q(\alpha, n)$, where α is the probability of incorrectly rejecting the suspected outlier and n is the number of samples in the data set. There are several versions of Dixon's Q-Test, each of which calculates a value for Q_{ij} where i is the number of suspected outliers on one end of the data set and j is the number of suspected outliers on the opposite end of the data set. The critical values for Q here are for a single outlier, Q_{10} , where

$$Q_{\text{exp}} = Q_{10} = \frac{|\text{outlier's value} - \text{nearest value}|}{\text{largest value} - \text{smallest value}}$$

The suspected outlier is rejected if Q_{exp} is greater than $Q(\alpha, n)$. For additional information consult Rorabacher, D. B. "Statistical Treatment for Rejection of Deviant Values: Critical Values of Dixon's 'Q' Parameter and Related Subrange Ratios at the 95% confidence Level," *Anal. Chem.* **1991**, 63, 139–146.

Table 12.4.1: Critical Values for Dixon's Q-Test

$\frac{\alpha \rightarrow}{n \downarrow}$	0.1	0.05	0.04	0.02	0.01
3	0.941	0.970	0.976	0.988	0.994
4	0.765	0.829	0.846	0.889	0.926
5	0.642	0.710	0.729	0.780	0.821
6	0.560	0.625	0.644	0.698	0.740
7	0.507	0.568	0.586	0.637	0.680
8	0.468	0.526	0.543	0.590	0.634
9	0.437	0.493	0.510	0.555	0.598
10	0.412	0.466	0.483	0.527	0.568

12.5: Critical Values for Grubb's Test

The following table provides critical values for $G(\alpha, n)$, where α is the probability of incorrectly rejecting the suspected outlier and n is the number of samples in the data set. There are several versions of Grubb's Test, each of which calculates a value for G_{ij} where i is the number of suspected outliers on one end of the data set and j is the number of suspected outliers on the opposite end of the data set. The critical values for G given here are for a single outlier, G_{10} , where

$$G_{\text{exp}} = G_{10} = \frac{|X_{\text{out}} - \bar{X}|}{s}$$

The suspected outlier is rejected if G_{exp} is greater than $G(\alpha, n)$.

Table 12.5.1: Critical Values for the Grubb's Test

$\frac{\alpha \rightarrow}{n \downarrow}$	0.05	0.01
3	1.155	1.155
4	1.481	1.496
5	1.715	1.764
6	1.887	1.973
7	2.202	2.139
8	2.126	2.274
9	2.215	2.387
10	2.290	2.482
11	2.355	2.564
12	2.412	2.636
13	2.462	2.699
14	2.507	2.755
15	2.549	2.755

12.6: Critical Values for the Wilcoxon Signed Rank Test

The following table provides critical values at $\alpha = 0.05$ for the Wilcoxon signed rank test where n is the number of samples in the data set. An entry of NA means the test cannot be applied. The null hypothesis of no difference between the samples can be rejected when the test statistic is less than or equal to the critical values for the number of samples.

Table 12.6.1: Critical Values for Wilcoxon Signed Rank Test with $\alpha = 0.05$

n	one-tailed test	two-tailed test
5	0	NA
6	2	0
7	3	2
8	5	3
9	8	5
10	10	8
11	13	10
12	17	13
13	21	17
14	25	21
15	30	25
16	35	30
17	41	35
18	47	40
19	53	46
20	60	52

This page titled 12.6: Critical Values for the Wilcoxon Signed Rank Test is shared under a CC BY-NC-SA 4.0 license and was authored, remixed, and/or curated by LibreTexts.

12.7: Critical Values for the Wilcoxon Ranked Sum Test

The following table provides critical values at $\alpha = 0.05$ for the Wilcoxon ranked sum test where n_1 and n_2 are the number of samples in the two sets of data where $n_1 \leq n_2$. An entry of NA means the test cannot be applied. The null hypothesis of no difference between the samples can be rejected when the test statistic is less than or equal to the critical values for the number of samples.

12.7.1: Critical Values for Wilcoxon Ranked Sum Test with $\alpha = 0.05$

n_1	n_2	one-tailed test	two-tailed test
3	3	0	NA
3	4	0	NA
3	5	1	0
3	6	2	1
4	4	1	0
4	5	2	1
4	6	3	2
4	7	4	3
5	5	4	2
5	6	5	3
5	7	6	5
5	8	8	6
6	6	7	5
6	7	8	6
6	8	10	8
6	9	12	10
7	7	11	8
7	8	13	10
7	9	15	12
7	10	17	14

This page titled [12.7: Critical Values for the Wilcoxon Ranked Sum Test](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by LibreTexts.

13: RESOURCES

A collection of resources on chemometrics and R.

[13.1: Chemometric Resources](#)

[13.2: R Resources](#)

This page titled [13: Resources](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [David Harvey](#).

13.1: Chemometric Resources

Books

The following small collection of books provide a broad introduction to chemometric methods of analysis. The text by Miller and Miller is a good entry-level textbook suitable for the undergraduate curriculum. The text by Massart, et. al. is a particularly comprehensive resource.

- Anderson, R. L. **Practical Statistics for Analytical Chemists**, Van Nostrand Reinhold: New York; 1987.
- Beebe, K. R.; Pell, R. J.; Seasholtz, M. B. **Chemometrics: A Practical Guide**, Wiley, 1998.
- Brereton, Richard G. **Data Driven Extraction for Science**, 2nd Edition, Wiley, 2018.
- Graham, R. C. **Data Analysis for the Chemical Sciences**, VCH Publishers: New York; 1993.
- Larose, D. T.; Larose, C. D. **Discovering Knowledge in Data: An Introduction to Data Mining**, Wiley, 2014.
- Mark, H.; Workman, J. **Statistics in Spectroscopy**, Academic Press: Boston; 1991.
- Massart, D. L.; Vandeginste, B. G. M.; Lewi, P. J.; Smeyers-Verbeke, J. **Handbook of Chemometrics and Qualimetrics: Part A and Part B**, Elsevier, 1997.
- Miller, J. N.; Miller, J. C. **Statistics and Chemometrics for Analytical Chemistry**, 7th Edition, Pearson, 2018.
- Schutt, R.; O'Neil, C. **Doing Data Science: Straight Talk From the Frontline**, O'Reilly, 2014.
- Sharaf, M. H.; Illman, D. L.; Kowalski, B. R. **Chemometrics**, Wiley-Interscience: New York; 1986.

Although not resources on chemometrics, the following books provide a broad introduction to the statistical methods that underlie chemometrics.

- Boslaugh, S. **Statistics in a Nutshell: A Desktop Quick Reference**, O'Reilly, 2013.
- Larose, D. T.; Larose, C. D. **Discovering Knowledge in Data: An Introduction to Data Mining**, Wiley, 2014.
- Schutt, R.; O'Neil, C. **Doing Data Science: Straight Talk From the Frontline**, O'Reilly, 2014.
- van Belle, G. **Statistical Rules of Thumb**, Wiley, 2008.

The following books provide more specialized coverage of topics relevant to chemometrics.

- Mason, R. L.; Gunst, R. F.; Hess, J. L. **Statistical Design and Analysis of Experiments**, Wiley: New York, 1989.
- Myers, R. H.; Montgomery, D. C. **Response Surface Methodology**, Wiley, 2002.

The following books provide guidance on the visualization of data, both in figures and in tables.

- Bertin, J. **Semiology of Graphics**, esri press, 1983.
- Few, S. **Now You See It**, Analytics Press, 2009.
- Few, S. **Show Me the Numbers**, Analytics Press, 2012.
- Few, S. **Information Dashboard Design**, Analytics Press, 2013.
- Robins, N. B. **Creating More Effective Graphs**, Charchouse, 2013.
- Tufte, E. R. **Envisioning Information**, Graphics Press, 1990.
- Tufte, E. R. **Visual Explanations**, Graphics Press, 1997.
- Tufte, E. R. **The Visual Display of Quantitative Information**, Graphics Press, 2001.
- Tufte, E. R. **Beautiful Evidence**, Graphics Press, 2006.

The following textbook provides a broad introduction to analytical chemistry, including sections on chemometric topics.

- Harvey, D. T. **Analytical Chemistry 2.1** (available [here](#) and [here](#)).

Articles

The following paper provides a general theory of types of measurements.

- Stevens, S. S. "On the Theory of Scales of Measurements," *Science*, **1946**, *103*, 677-680.

The detection of outliers, particularly when working with a small number of samples, is discussed in the following papers.

- Analytical Methods Committee "Robust Statistics—How Not To Reject Outliers Part 1. Basic Concepts," *Analyst* **1989**, 114, 1693–1697.
- Analytical Methods Committee "Robust Statistics—How Not to Reject Outliers Part 2. Inter-laboratory Trials," *Analyst* **1989**, 114, 1699–1702.
- Analytical Methods Committee "Rogues and Suspects: How to Tackle Outliers," AMCTB 39, 2009.
- Analytical Methods Committee "Robust statistics: a method of coping with outliers," AMCTB 6, 2001.
- Analytical Methods Committee "Using the Grubbs and Cochran tests to identify outliers," *Anal. Methods*, **2015**, 7, 7948–7950.
- Efstathiou, C. "Stochastic Calculation of Critical Q-Test Values for the Detection of Outliers in Measurements," *J. Chem. Educ.* **1992**, 69, 773–736.
- Efstathiou, C. "Estimation of type 1 error probability from experimental Dixon's Q parameter on testing for outliers within small data sets," *Talanta* **2006**, 69, 1068–1071.
- Kelly, P. C. "Outlier Detection in Collaborative Studies," *Anal. Chem.* **1990**, 73, 58–64.
- Mitschele, J. "Small Sample Statistics," *J. Chem. Educ.* **1991**, 68, 470–473.

The following papers provide additional information on error and uncertainty.

- Analytical Methods Committee "Optimizing your uncertainty—a case study," AMCTB 32, 2008.
- Analytical Methods Committee "Dark Uncertainty," AMCTB 53, 2012.
- Analytical Methods Committee "What causes most errors in chemical analysis?" AMCTB 56, 2013.
- Andraos, J. "On the Propagation of Statistical Errors for a Function of Several Variables," *J. Chem. Educ.* **1996**, 73, 150–154.
- Donato, H.; Metz, C. "A Direct Method for the Propagation of Error Using a Personal Computer Spreadsheet Program," *J. Chem. Educ.* **1988**, 65, 867–868.
- Gordon, R.; Pickering, M.; Bisson, D. "Uncertainty Analysis by the 'Worst Case' Method," *J. Chem. Educ.* **1984**, 61, 780–781.
- Guare, C. J. "Error, Precision and Uncertainty," *J. Chem. Educ.* **1991**, 68, 649–652.
- Guedens, W. J.; Yperman, J.; Mullens, J.; Van Poucke, L. C.; Pauwels, E. J. "Statistical Analysis of Errors: A Practical Approach for an Undergraduate Chemistry Lab Part 1. The Concept," *J. Chem. Educ.* **1993**, 70, 776–779.
- Guedens, W. J.; Yperman, J.; Mullens, J.; Van Poucke, L. C.; Pauwels, E. J. "Statistical Analysis of Errors: A Practical Approach for an Undergraduate Chemistry Lab Part 2. Some Worked Examples," *J. Chem. Educ.* **1993**, 70, 838–841.
- Heydorn, K. "Detecting Errors in Micro and Trace Analysis by Using Statistics," *Anal. Chim. Acta* **1993**, 283, 494–499.
- Hund, E.; Massart, D. L.; Smeyers-Verbeke, J. "Operational definitions of uncertainty," *Trends Anal. Chem.* **2001**, 20, 394–406.
- Kragten, J. "Calculating Standard Deviations and Confidence Intervals with a Universally Applicable Spreadsheet Technique," *Analyst* **1994**, 119, 2161–2165.
- Taylor, B. N.; Kuyatt, C. E. "Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results," NIST Technical Note 1297, 1994.
- Van Bramer, S. E. "A Brief Introduction to the Gaussian Distribution, Sample Statistics, and the Student's t Statistic," *J. Chem. Educ.* **2007**, 84, 1231.
- Yates, P. C. "A Simple Method for Illustrating Uncertainty Analysis," *J. Chem. Educ.* **2001**, 78, 770–771.

The following articles provide thoughts on the limitations of statistical analysis based on significance testing.

- Analytical Methods Committee "Significance, importance, and power," AMCTB 38, 2009.
- Analytical Methods Committee "An introduction to non-parametric statistics," AMCTB 57, 2013.
- Berger, J. O.; Berry, D. A. "Statistical Analysis and the Illusion of Objectivity," *Am. Sci.* **1988**, 76, 159–165.
- Kryzinski, M. "Importance of being uncertain," *Nat. Methods* **2013**, 10, 809–810.
- Kryzinski, M. "Significance, P values, and t-tests," *Nat. Methods* **2013**, 10, 1041–1042.
- Kryzinski, M. "Power and sample size," *Nat. Methods* **2013**, 10, 1139–1140.
- Leek, J. T.; Peng, R. D. "What is the question?," *Science* **2015**, 347, 1314–1315.

The following papers provide insight into organizing data in spreadsheets and visualizing data.

- Analytical Methods Committee "Representing data distributions with kernel density estimates," AMC Technical Brief, March 2006.
- Broman, K. W.; Woo, K. H. "Data Organization in Spreadsheets," *The American Statistician*, **2018**, 72, 2–10.
- Frigge, M.; Hoaglin, D. C.; Iglewicz, B. "Some Implementations of the Boxplot," *The American Statistician* **1989**, 43, 50–54.
- Midway, S. R. "Principles of Effective Data Visualizations," *PATTER*, **2020**, 1(9).
- Schwabish, J. A. "Ten Guidelines for Better Tables," *J. Benefit Cost Anal.* **2020**, 11, 151–178.

Websites

- NIST Engineering Statistics HandbookST (<https://www.itl.nist.gov/div898/handbook/>)
- Rice Virtual Lab in Statistics (<https://onlinestatbook.com/rvls.html>)
- Statistics for Analytical Chemistry (<https://science.widener.edu/svb/stats/stats.html>)

This page titled [13.1: Chemometric Resources](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by [David Harvey](#).

13.2: R Resources

Books

The following books, which I have found useful, either provide a broad introduction to the R programming language, or a more targeted coverage of a particular application. The texts published by O'Reilly have on-line versions made available for free; there entries here provide links to the on-line versions.

- Chambers, J. M. **Software for Data Analysis: Programming with R**, Springer: New York, 2008.
- [Chang, W. **R Graphics Cookbook**, O'Reilly, 2013.](#)
- Gardner, M. **Beginning R: The Statistical Programming Language**, Wiley, 2012.
- Gillespie, C.; Lovelace, R. **Efficient R Programming**, O'Reilly, 2020.
- Golemund, G. **Hands-On Programming with R**, O'Reilly, 2014.
- Horton, N. J.; Kleinman, K. **Using R and RStudio for Data Management, Statistical Analysis, and Graphics**, 2nd Edition, CRC Press, 2015.
- James, G.; Witten, D.; Hastie, T.; Tibshirani, R. **An Introduction to Statistical Learning with Applications in R**, Springer, 2013.
- Lander, J. P. **R for Everyone: Advanced Analytics and Graphics**, Addison Wesley, 2014.
- Kabacoff, Robert I. **R in Action: Data Analysis and Graphics with R**, Manning, 2011.
- Maindonald, J.; Braun, J. **Data Analysis and Graphics Using R**, Cambridge University Press: Cambridge, UK, 2003.
- Matloff, N. **The Art of R Programming**, No Starch Press, 2011.
- Sarkar, D. **Lattice: Multivariate Data Visualization With R**, Springer: New York, 2008.
- Vaughn, S. **Scientific Inference**, Cambridge, 2013.
- Wickham, H. **ggplot2**, Springer, 2009.
- [Wickham, H.; Golemund, G. **R for Data Science**, O'Reilly, 2017.](#)

Articles

- Doi, J.; Potter, G.; Wong, J. "Web Application Teaching Tools for Statistics Using R and Shiny", *Technology Innovations in Statistics Education*, **2016**, 9.

Websites

- CRANberries (<http://dirk.eddelbuettel.com/cranberries/>)
- The R Project for Statistical Computing (<https://www.r-project.org/>)
- The R Graph Gallery (<http://www.r-graph-gallery.com/>)
- R-Bloggers (<https://www.r-bloggers.com/>)
- RStudio (<https://www.rstudio.com/products/rstudio/>)
- RStudio Packages (<https://www.rstudio.com/products/rpackages/>)
- RWeekly (<https://rweekly.org/>)
- Stackoverflow (<https://stackoverflow.com/questions/tagged/r>)

INDEX

- B**
- bar plots
 - 3.1: Types of Visualizations
- C**
- cluster analysis
 - 11.2: Cluster Analysis
- D**
- dot plots
 - 3.1: Types of Visualizations
- F**
- field:achem
 - 1: R and RStudio
 - 1.1: Installing and Accessing R and RStudio
 - 10: Cleaning Up Data
 - 10.1: Signals and Noise
 - 10.2: Improving the Signal-to-Noise Ratio
 - 10.3: Background Removal
 - 10.4: Using R to Clean Up Data
 - 10.5: Exercises
 - 11: Finding Structure in Data
 - 11.1: What Do We Mean By Structure?
 - 11.2: Cluster Analysis
 - 11.3: Principal Component Analysis
 - 11.5: Using R for a Cluster Analysis
 - 11.6: Using R for a Principal Component Analysis
 - 11.8: Exercises
 - 13: Resources
 - 13.1: Chemometric Resources
 - 13.2: R Resources
 - 2: Types of Data
 - 3: Visualizing Data
 - 3.1: Types of Visualizations
 - 3.2: Using R to Visualize Data
 - 3.3: Creating Plots From Scratch in R Using Base Graphics
 - 4: Summarizing Data
 - 4.1: Ways to Summarize Data
 - 4.2: Using R to Summarize Data
 - 4.3: Exercises
 - 5: The Distribution of Data
 - 5.1: Terminology
 - 5.2: Theoretical Models for the Distribution of Data
 - 5.3: The Central Limit Theorem
 - 5.4: Modeling Distributions Using R
 - 5.5: Exercises
 - 6: Uncertainty of Data
 - 6.1: Properties of a Normal Distribution
 - 6.2: Confidence Intervals
 - 6.3: Using R to Model Properties of a Normal Distribution
 - 6.4: Using R to Find Confidence Intervals
 - 6.5: Exercises
 - 7: Testing the Significance of Data
 - 7.1: Significance Testing
 - 7.2: Significance Tests for Normal Distributions
 - 7.3: Analysis of Variance
 - 7.4: Non-Parametric Significance Tests
 - 7.5: Using R for Significance Testing and Analysis of Variance
 - 7.6: Exercises
 - 8: Calibrating Data
 - 8.1: Unweighted Linear Regression With Errors in y
 - 8.2: Weighted Linear Regression with Errors in y
 - 8.3: Weighted Linear Regression With Errors in Both x and y
 - 8.4: Curvilinear, Multivariable, and Multivariate Regression
 - 8.5: Using R for a Linear Regression Analysis
 - 8.6: Exercises
 - 9: Optimizing Data
 - 9.1: Response Surfaces
 - 9.2: Searching Algorithms
 - 9.3: One-Factor-at-a-Time Optimizations
 - 9.4: Simplex Optimization
 - 9.5: Mathematical Models of Response Surfaces
 - 9.6: Using R to Model a Response Surface (Multiple Regression)
 - 9.7: Exercises
- H**
- histograms
 - 3.1: Types of Visualizations
- J**
- jupyter:r
 - 1: R and RStudio
 - 2: Types of Data
- P**
- principal component analysis
 - 11.3: Principal Component Analysis
- S**
- simplex optimization
 - 9.4: Simplex Optimization
 - stripcharts
 - 3.1: Types of Visualizations
- W**
- Wilcoxon rank sum test
 - 7.4: Non-Parametric Significance Tests
 - Wilcoxon signed rank test
 - 7.4: Non-Parametric Significance Tests

GLOSSARY

S

Sample Word 1 | Sample Definition 1

This page titled [Glossary](#) is shared under a [CC BY-NC-SA 4.0](#) license and was authored, remixed, and/or curated by LibreTexts.

DETAILED LICENSING

Overview

Title: Chemometrics Using R (Harvey)

Webpages: 88

Applicable Restrictions: Noncommercial

All licenses found:

- [CC BY-NC-SA 4.0](#): 89.8% (79 pages)
- [Undeclared](#): 10.2% (9 pages)

By Page

- Chemometrics Using R (Harvey) - [CC BY-NC-SA 4.0](#)
 - Front Matter - [Undeclared](#)
 - TitlePage - [Undeclared](#)
 - InfoPage - [Undeclared](#)
 - Table of Contents - [Undeclared](#)
 - Licensing - [Undeclared](#)
 - What is Chemometrics and Why Study it? - [Undeclared](#)
 - 1: R and RStudio - [CC BY-NC-SA 4.0](#)
 - 1.1: Installing and Accessing R and RStudio - [CC BY-NC-SA 4.0](#)
 - 1.2: The Basics of Working With R - [CC BY-NC-SA 4.0](#)
 - 1.3: Exercises - [CC BY-NC-SA 4.0](#)
 - 2: Types of Data - [CC BY-NC-SA 4.0](#)
 - 2.1: Ways to Describe Data - [CC BY-NC-SA 4.0](#)
 - 2.2: Using R to Organize and Manipulate Data - [CC BY-NC-SA 4.0](#)
 - 2.3: Exercises - [CC BY-NC-SA 4.0](#)
 - 3: Visualizing Data - [CC BY-NC-SA 4.0](#)
 - 3.1: Types of Visualizations - [CC BY-NC-SA 4.0](#)
 - 3.2: Using R to Visualize Data - [CC BY-NC-SA 4.0](#)
 - 3.3: Creating Plots From Scratch in R Using Base Graphics - [CC BY-NC-SA 4.0](#)
 - 3.4: Exercises - [CC BY-NC-SA 4.0](#)
 - 4: Summarizing Data - [CC BY-NC-SA 4.0](#)
 - 4.1: Ways to Summarize Data - [CC BY-NC-SA 4.0](#)
 - 4.2: Using R to Summarize Data - [CC BY-NC-SA 4.0](#)
 - 4.3: Exercises - [CC BY-NC-SA 4.0](#)
 - 5: The Distribution of Data - [CC BY-NC-SA 4.0](#)
 - 5.1: Terminology - [CC BY-NC-SA 4.0](#)
 - 5.2: Theoretical Models for the Distribution of Data - [CC BY-NC-SA 4.0](#)
 - 5.3: The Central Limit Theorem - [CC BY-NC-SA 4.0](#)
 - 5.4: Modeling Distributions Using R - [CC BY-NC-SA 4.0](#)
 - 5.5: Exercises - [CC BY-NC-SA 4.0](#)
 - 6: Uncertainty of Data - [CC BY-NC-SA 4.0](#)
 - 6.1: Properties of a Normal Distribution - [CC BY-NC-SA 4.0](#)
 - 6.2: Confidence Intervals - [CC BY-NC-SA 4.0](#)
 - 6.3: Using R to Model Properties of a Normal Distribution - [CC BY-NC-SA 4.0](#)
 - 6.4: Using R to Find Confidence Intervals - [CC BY-NC-SA 4.0](#)
 - 6.5: Exercises - [CC BY-NC-SA 4.0](#)
 - 7: Testing the Significance of Data - [CC BY-NC-SA 4.0](#)
 - 7.1: Significance Testing - [CC BY-NC-SA 4.0](#)
 - 7.2: Significance Tests for Normal Distributions - [CC BY-NC-SA 4.0](#)
 - 7.3: Analysis of Variance - [CC BY-NC-SA 4.0](#)
 - 7.4: Non-Parametric Significance Tests - [CC BY-NC-SA 4.0](#)
 - 7.5: Using R for Significance Testing and Analysis of Variance - [CC BY-NC-SA 4.0](#)
 - 7.6: Exercises - [CC BY-NC-SA 4.0](#)
 - 8: Calibrating Data - [CC BY-NC-SA 4.0](#)
 - 8.1: Unweighted Linear Regression With Errors in y - [CC BY-NC-SA 4.0](#)
 - 8.2: Weighted Linear Regression with Errors in y - [CC BY-NC-SA 4.0](#)
 - 8.3: Weighted Linear Regression With Errors in Both x and y - [CC BY-NC-SA 4.0](#)
 - 8.4: Curvilinear, Multivariable, and Multivariate Regression - [CC BY-NC-SA 4.0](#)
 - 8.5: Using R for a Linear Regression Analysis - [CC BY-NC-SA 4.0](#)
 - 8.6: Exercises - [CC BY-NC-SA 4.0](#)
 - 9: Optimizing Data - [CC BY-NC-SA 4.0](#)
 - 9.1: Response Surfaces - [CC BY-NC-SA 4.0](#)
 - 9.2: Searching Algorithms - [CC BY-NC-SA 4.0](#)

- 9.3: One-Factor-at-a-Time Optimizations - CC BY-NC-SA 4.0
- 9.4: Simplex Optimization - CC BY-NC-SA 4.0
- 9.5: Mathematical Models of Response Surfaces - CC BY-NC-SA 4.0
- 9.6: Using R to Model a Response Surface (Multiple Regression) - CC BY-NC-SA 4.0
- 9.7: Exercises - CC BY-NC-SA 4.0
- 10: Cleaning Up Data - CC BY-NC-SA 4.0
 - 10.1: Signals and Noise - CC BY-NC-SA 4.0
 - 10.2: Improving the Signal-to-Noise Ratio - CC BY-NC-SA 4.0
 - 10.3: Background Removal - CC BY-NC-SA 4.0
 - 10.4: Using R to Clean Up Data - CC BY-NC-SA 4.0
 - 10.5: Exercises - CC BY-NC-SA 4.0
- 11: Finding Structure in Data - CC BY-NC-SA 4.0
 - 11.1: What Do We Mean By Structure? - CC BY-NC-SA 4.0
 - 11.2: Cluster Analysis - CC BY-NC-SA 4.0
 - 11.3: Principal Component Analysis - CC BY-NC-SA 4.0
 - 11.4: Multivariate Linear Regression - CC BY-NC-SA 4.0
 - 11.5: Using R for a Cluster Analysis - CC BY-NC-SA 4.0
 - 11.6: Using R for a Principal Component Analysis - CC BY-NC-SA 4.0
 - 11.7: Using R for a Multivariate Linear Regression - CC BY-NC-SA 4.0
 - 11.8: Exercises - CC BY-NC-SA 4.0
- 12: Appendices - CC BY-NC-SA 4.0
 - 12.1: Single-Sided Normal Distribution - CC BY-NC-SA 4.0
 - 12.2: Critical Values for t-Test - CC BY-NC-SA 4.0
 - 12.3: Critical Values for F-Test - CC BY-NC-SA 4.0
 - 12.4: Critical Values for Dixon's Q-Test - CC BY-NC-SA 4.0
 - 12.5: Critical Values for Grubb's Test - CC BY-NC-SA 4.0
 - 12.6: Critical Values for the Wilcoxon Signed Rank Test - CC BY-NC-SA 4.0
 - 12.7: Critical Values for the Wilcoxon Ranked Sum Test - CC BY-NC-SA 4.0
- 13: Resources - CC BY-NC-SA 4.0
 - 13.1: Chemometric Resources - CC BY-NC-SA 4.0
 - 13.2: R Resources - CC BY-NC-SA 4.0
- Back Matter - *Undeclared*
 - Index - *Undeclared*
 - Glossary - CC BY-NC-SA 4.0
 - Detailed Licensing - *Undeclared*