GLOVER & MITCHELL

AN INTRODUCTION TO BIOSTATISTICS USING R

QUE POR EL HILO SE SACARÁ EL OVILLO.

(BY A SMALL SAMPLE, WE MAY JUDGE OF THE WHOLE PIECE.)

MIGUEL DE CERVANTES, *DON QUIXOTE*

Contents

0. Introduction to R

We assume that your are reading this supplement to *An Introduction to Biostatistics* because your instructor has decided to use R as the statistical software for your course or because you are a very motivated student and want to learn both elementary statistics and R at the same time. This supplement does not provide screen shots of the program or lots of help installing R. This is better done by your instructor or TA who can actually demonstrate the process.

What is R?

R is a language and environment for statistical computing and graphics. It is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form. It compiles and runs on a wide variety of operating systems such as Windows, MacOS, and most UNIX platforms and similar systems (including FreeBSD and Linux). While there are many statistical software programs available, we have chosen to provide this supplement to *An Introduction to Biostatistics* using R because it is both powerful and free.

To learn much more about R, go to `http://www.r-project.org`. Note: If you are reading this supplement either online or as a pdf on your own computer or tablet, just click on any link to be redirected to the appropriate web page.

Installation of R

If you don't already have R on your computer or have access to a copy of R through your institution, then you can download R for free by going to `http://www.r-project.org`. Follow the instructions. We assume your instructor or TA or a friend will help you with this. There are useful `youtube` videos available, as well.

R Resources and Getting Started

There are several great free online resources to help you learn more about R. Start with the R-project homepage at `http://www.r-project.org`. Here are a few others that are written for those just beginning to use R.

1. John Verzani's `http://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf` is a 113 page introduction to R that is well worth downloading.

2. The web pages "Software Resources for R" starting at `http://courses.statistics.com/software/R/Rhome.htm` are useful, especially the Getting Started link, `http://courses.statistics.com/software/R/R00.htm`.

3. A brief "R: A self-learn tutorial" may be found at `http://www.nceas.ucsb.edu/files/scicomp/Dloads/RProgramming/BestFirstRTutorial.pdf`.

4. A series of tutorials by Google Developers consisting of 21 short videos (total length, one hour and seven minutes) can be found at `https://www.youtube.com/playlist?list=PLOU2XLYxmsIK9qQfztXeybpHvru-TrqAP`. You may wish to watch the first few for tips on getting started. Return to the later videos as you gain more experience with R.

5. Kelly Black's extensive online "R Tutorial" can be found at `http://www.cyclismo.org/tutorial/R/index.html`.

6. Another 100-page introduction to R by Germán Rodíguez can be found at `http://data.princeton.edu/R/introducingR.pdf`.

7. For a very short introduction to R and the R-Studio graphical interface see `http://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf`. To keep things simple, we will not assume that you are using R-Studio.

There are many other introductions to R available online and more being written all the time. Make use of them.

Starting R

Assuming that you have installed R and that you have started the program, you are faced with some text similar to

```
R version 3.1.1 (2014-07-10) -- "Sock it to Me"
Copyright (C) 2014 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin10.8.0 (64-bit)

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

  Natural language support but running in an English locale

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

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

>
```

The > is called the **prompt**. In the examples that follow, the prompt is not something you type. Rather, it is the symbol that R uses to indicate that it is waiting for you to type in the next command. If a command line is too long to fit on a single line a + is automatically inserted by R to indicate the continuation of the prompt on the next line. We will remind you about this when it first occurs in these notes.

One can do basic arithmetic in R. For example, we can add 4 and 5 in the obvious way. Be sure to hit "Return" after typing.

```
> 4 + 5

## [1] 9
```

The lightly shaded background above and the distinctive typeface distinguish R input and output from the explanatory text surrounding it. Material following the prompt > is what we typed. Lines that begin with two hashtags ## are the output that results from an R command. If you are looking at a color version of these materials, you will also see that the various types of R input are color coded.

The other basic mathematical operations work the same way: use - for subtraction, * for multiplication, and / for division. R also has many built in functions. For example, to find the square root of 3, use the `sqrt( )` function:

```
> sqrt(3)

## [1] 1.73205
```

Creating and Accessing Data Sets

Most statistical calculations require a data set containing several numbers. There are several ways to enter a data set. If the data consist of only a few numbers, the `c( )` function can be used. This function combines or concatenates terms. Suppose the heights of ten male faculty were recorded in cm: 171, 177, 178, 175, 202, 180, 192, 182, 195, and 190. These data can be entered into a single object (or variable) called `height`. To do so, type

```
> height <- c(171, 177, 178, 175, 202, 180, 192, 182, 195, 190)
> height

##  [1] 171 177 178 175 202 180 192 182 195 190
```

There are a few things to notice.

- All of the values were assigned to a single object called `height`. Try to make the names of your objects meaningful in the context of the problem or question. We could have simply called the data `x`, but if we looked back at it later we might not know to what `x` referred.

- The assignment operator is an "arrow" formed by typing `<-`. The arrow indicates that we are assigning the values 171, 177, 178, 175, 202, 180, 192, 182, 195, and 190 to the R object (or variable) `height`.

- The values of `height` did not automatically print out. However, typing the name of the object will cause its value(s) to be printed.

- The `[1]` indicates that the output is a vector (a sequence of numbers or other objects) and that the first value printed on the row is actually the first value in `height`.

Many basic functions can be applied directly to entire data sets. For example, to take the square root of each `height` use

```
> sqrt(height)

##  [1] 13.0767 13.3041 13.3417 13.2288 14.2127 13.4164 13.8564 13.4907
##  [9] 13.9642 13.7840
```

Notice that when the values of a function are not stored in a variable, the result is immediately printed. In the second line of the output above, the leading `[9]` indicates that the first entry on this line is the ninth element of `sqrt(height)`.

There are 2.54 cm per inch, so to convert the heights to inches, use

```
> height/2.54             # height in inches

##  [1] 67.3228 69.6850 70.0787 68.8976 79.5276 70.8661 75.5906 71.6535
##  [9] 76.7717 74.8031
```

We will often put comments by the commands. These are indicated by a single hashtag `#`. They are a convenient way to make notes in your computations. Anything following a hashtag, for example `# height in inches` above, is ignored by R.

To compute the average or mean of the heights above, add all of the heights and divide this sum by the number of heights. In R use the `sum( )` function to add all of the entries in a vector.

```
> sum(height)

## [1] 1842
```

To find the number of entries in a vector, use the `length( )` function.

```
> length(height)

## [1] 10
```

So the mean height is

```
> meanHt <- sum(height)/length(height)
> meanHt

## [1] 184.2
```

Note that value of the mean height has been put into the variable `meanHt` so that it can be re-used without having to recalculate the value. To square each of the individual heights use

```
> height^2

##  [1] 29241 31329 31684 30625 40804 32400 36864 33124 38025 36100
```

Order of operations is important. Compare the sum of these squared heights to the square of the sum of the heights. Note the placement of the squaring operation.

```
> sum(height^2)           # sum of the squared heights

## [1] 340196

> sum(height)^2           # square of the sum of the heights

## [1] 3392964

> (sum(height))^2         # square of the sum of the heights, again

## [1] 3392964
```

The two numbers are different. The extra set of parentheses in the third command above clarifies the order of operations: sum, then square.

More complicated calculations may be formed from basic calculations. For example, the **corrected sum of squares** is defined as

$$\sum_{i=1}^{n}(X_i - \overline{X})^2.$$

Recalling that we calculated the `meanHt` earlier, the corrected sum of squares for the height data can be calculated.

```
> sum((height-meanHt)^2)

## [1] 899.6
```

It is easy to carry out basic descriptive statistical operations on data using the many functions built into R as illustrated below.

```
> median(height)                # the median height

## [1] 181

> min(height)                   # the minimum height

## [1] 171

> max(height)                   # the maximum height

## [1] 202
```

Suppose that the arm spans of these same ten faculty were also measured (in the same order) and that the data were entered into R.

```
> span <- c(173, 182, 182, 178, 202, 188, 198, 185, 193, 186)     # in cm
> span

##  [1] 173 182 182 178 202 188 198 185 193 186
```

To determine the difference between the arm span and height for each person by subtracting `height` from `span`, use

```
> difference <- span - height
> difference

##  [1]  2  5  4  3  0  8  6  3 -2 -4
```

It is often convenient to combine data sets that are related. To combine `span`, `height`, and `difference` into a single table, put the data into a so-called data frame (which we name `faculty.dat`) using the `data.frame( )` command.

```
> faculty.dat <- data.frame(span, height, difference)
> faculty.dat

##    span height difference
## 1   173    171          2
## 2   182    177          5
## 3   182    178          4
## 4   178    175          3
## 5   202    202          0
## 6   188    180          8
## 7   198    192          6
## 8   185    182          3
## 9   193    195         -2
## 10  186    190         -4
```

Unless otherwise specified, the column names are the variable names. To add a column for the ratio of `span` to `height`, create a new column using `faculty.dat["ratio"]` and put `span/height` into this column

```
> faculty.dat["ratio"] <- span/height     # note the quotes
> faculty.dat

##    span height difference    ratio
## 1   173    171          2 1.011696
## 2   182    177          5 1.028249
## 3   182    178          4 1.022472
## 4   178    175          3 1.017143
## 5   202    202          0 1.000000
## 6   188    180          8 1.044444
## 7   198    192          6 1.031250
## 8   185    182          3 1.016484
## 9   193    195         -2 0.989744
## 10  186    190         -4 0.978947
```

The various columns of `faculty.dat` can now be referenced by using their names. For example, the `ratio` column is referenced by using the data frame name followed by a $ followed by the column name: `faculty.dat$ratio`. To determine the mean of these data, use

```
> sum(faculty.dat$ratio)/length(faculty.dat$ratio)

## [1] 1.01404
```

Accessing Online Data Files

One of the neat things about R is the ability to access and analyze (large) online data files without ever downloading them to your own computer. With this in mind, we have created online data files for all of the examples and problems in *An Introduction to Biostatistics* that require them. You can download these files to use with R or any other statistical software, or in R you may access the text files from online and do the necessary analysis without ever copying them to your own computer.

The data files for *An Introduction to Biostatistics* use the following naming conventions.

- The data file for Example n of Chapter XY is referred to as `http://waveland.com/Glover-Mitchell/ExampleXY-n.txt`. For instance the data for Example 2 of Chapter 1 is located at `http://waveland.com/Glover-Mitchell/Example01-2.txt`. Note the two-digit chapter reference and the dash between the chapter number and the example number.

- The data file for Problem n of Chapter XY is referred to as `http://waveland.com/Glover-Mitchell/ProblemXY-n.txt`. For instance the data for Problem 5 of Chapter 1 is located at `http://waveland.com/Glover-Mitchell/Problem01-5.txt`.

The data are in columns. Each column has a brief text "header" that describes the data. If a file contains multiple columns, the columns are separated by tabs. If you are reading this on your computer, try clicking on the links to be taken to the actual data files.

EXAMPLE 0.1. To read the tree data in `http://waveland.com/Glover-Mitchell/Example01-2.txt` and store the result in a table in R, use the `read.table( )` command. Since every data file for our text has a header, in the `read.table( )` command set `header = TRUE`. The word `TRUE` must be in capital letters. As a shortcut, one can also use `header = T`. To list the entire table, enter the table's name. Our convention is to name the data table `data.Ex01.2`. While this is a long name, it will help keep straight the current problem of interest. You are likely to use shorter names. Nonetheless, try to make them descriptive.

```
> data.Ex01.2 <- read.table("http://waveland.com/Glover-Mitchell/Example01-2.txt",
+ header = TRUE)
> data.Ex01.2

##    CCH
## 1   17
## 2   19
## 3   31
## 4   39
## 5   48
## 6   56
## 7   68
## 8   73
## 9   73
## 10  75
## 11  80
## 12 122
```

Notice the + on the second line above. This is an instance where the entire command does not fit on a single line. The + indicates that the command continues to the next line.

This is not something that you type and it certainly does not mean addition in this context. If you hit return before a command is finished, R automatically prints + to indicate that the command prompt has continued.

In this example, the header is CCH, which stands for circumference at chest height. The standard R functions can now be applied to these data. For example, the mean and corrected sum of squares for CCH are found in the usual way. (Remember: A column of a data frame is referenced by using the data frame name followed by a $ followed by the column name.)

```
> meanCCH <-sum(data.Ex01.2$CCH)/length(data.Ex01.2$CCH)
> meanCCH

## [1] 58.4167

> sum((data.Ex01.2$CCH - meanCCH)^2)      # corrected sum of squares

## [1] 9812.92
```

For larger data tables, it is convenient to list only the first and/or last few rows of a table. This is done using the head( ) and tail( ) commands. The syntax is head(tableName), which by default lists the first six rows of the table, or head(tableName, n = k) to list the first *k* rows. Similarly for tail( ). For example,

```
> head(data.Ex01.2$CCH)

## [1] 17 19 31 39 48 56

> tail(data.Ex01.2$CCH, n = 2)

## [1]  80 122
```

Help and Quitting

Various types of help are available. Within R you may wish to use the following commands

```
> help.start()   # general help, starts up your web browser
> help(foo)      # help about function 'foo'
> example(foo)   # show an example of function 'foo'
```

That's enough to get you started. Make up a few data sets of your own and try some calculations. Remember to quit R use q().

Command Summary

A little more information about each command used in this introduction is provided here. Most commands have additional options which can be found online using a simple web search, or by consulting one of the references listed earlier, or by using help in R.

c( )

Details: The simplest way to enter small data sets in R is with the c( ) command. This function combines or concatenates terms. To put the values 1, 4, 4, 5, 6 into the data set x, use

```
> x <- c(1, 4, 4, 5, 6)
> x

## [1] 1 4 4 5 6
```

Consecutive numbers such as 3, 4, 5, 6, 7 can be entered into a data set `y` using

```
> y <- c(3:7)
> y

## [1] 3 4 5 6 7
```

Even text data can be entered in this way. The names “Tom”, “Maria”, “Kemberlei”, “Jordan”, and “Chequira” can be put into the data set `students` using

```
> students <- c("Tom", "Maria", "Keberlei", "Jordan", "Chequira")
> students

## [1] "Tom"      "Maria"    "Keberlei" "Jordan"   "Chequira"
```

Note the required quotation marks around each of the text terms.

`data.frame(x, y, ...)`

Details: `data.frame( )` is used to put related data `x, y, ...` into a single table. The data sets must have the same length. To put the variables `x` and `y` from the entry above into `myTable` use

```
> myTable <- data.frame(x,y)
> myTable

##   x y
## 1 1 3
## 2 4 4
## 3 4 5
## 4 5 6
## 5 6 7
```

Notice that the variable names are the column names. Suppose that `x` and `y` actually represented the number of A’s and B’s of the `students` listed earlier. We can put all this information and the corresponding GPA’s (assuming these students earned only A’s and B’s) into a single table. Notice how the columns are renamed.

```
> studentRecord <- data.frame(Name = students, A = x, B = y,
+ GPA = (4*x + 3*y)/(x + y))
> # Name, A, B, and GPA will be the column names with
> # corresponding data from the vectors students, x, and y.
> # The last column, GPA, is calculated from x and y.
> studentRecord
```

```
##       Name A B     GPA
## 1      Tom 1 3 3.25000
## 2    Maria 4 4 3.50000
## 3 Keberlei 4 5 3.44444
## 4   Jordan 5 6 3.45455
## 5 Chequira 6 7 3.46154
```

The various columns of data in a table can be addressed as variables: Use the data frame name followed by a $ followed by the column name. In the table above, GPA is defined only within the data frame `studentRecord`, so it would be addressed as `studentRecord$GPA`. For example, to find the maximum GPA, use

```
> max(studentRecord$GPA)

## [1] 3.5
```

`head(tableName)`, `tail(tableName)`

Details: These commands list the first or last six rows of a table. Use `head(tableName, n = k)` to list just the first *k* rows. Similarly for `tail( )`. See the `read.table( )` entry for examples.

`length(x)`

Details: For an vector or list `x`, this function returns its length, that is, the number of entries in the object.

`read.table("pathName", header = FALSE)`

Details: This command is used to read data files that reside on the internet or on your own computer. By default, `header` is set to `FALSE`. All files for our text contain headers, so set `header = TRUE` when reading them. The pathname (address) must be in quotation marks.

EXAMPLE 0.2. The file `http://waveland.com/Glover-Mitchell/Example00-2.txt` contains the monthly rainfall records (in mm) from 1921 to 1998 at the Eagle Farm Racecourse in Brisbane, Australia. (Source: `http://www.bom.gov.au/climate/data/`.) This table is large so it makes sense to list just the first and last few rows of the table.

```
> data.Ex00.2 <- read.table("http://waveland.com/Glover-Mitchell/Example00-2.txt",
+ header = TRUE)
> head(data.Ex00.2, n = 3)        # display just the first 3 rows

##   Year  Jan   Feb   Mar   Apr  May   Jun   Jul  Aug  Sep  Oct  Nov   Dec
## 1 1921 129.9  16.9 194.7 203.0 19.8 197.9 167.7  4.3 44.8 28.4 57.2 226.4
## 2 1922  67.0 230.1  26.9   8.4 54.0  39.6 118.1  3.4 66.6 37.2 51.9 114.4
## 3 1923  46.7  21.4  87.1 192.4  9.1  73.9  64.5 24.4 36.1 10.9 47.1  62.5
##   Annual
## 1 1291.0
## 2  817.6
## 3  676.1

> tail(data.Ex00.2)               # display the last 6 rows by default
```

```
##    Year   Jan   Feb   Mar   Apr   May  Jun  Jul  Aug   Sep   Oct   Nov
## 73 1993  71.0  44.4  83.1  12.8  22.0  6.2 66.6 41.8  55.4  65.8  77.4
## 74 1994 235.6 101.7 139.0  49.2  65.6  9.4  9.0  2.2  53.0  41.4  41.6
## 75 1995  62.8 330.4  71.3  34.8  42.6 39.4  6.0 19.8  30.8  45.4 183.8
## 76 1996 241.8  52.6  28.6  78.2 678.4 34.2 50.2 41.0  49.2  31.4  82.8
## 77 1997 117.6  54.4  49.2   7.4 209.4 27.4 32.0  2.8  45.0 141.8 114.6
## 78 1998 132.6  36.2  41.4 164.8 140.0 11.2 23.6 91.6 112.6  26.0 104.8
##      Dec Annual
## 73  87.2  633.7
## 74  64.0  811.7
## 75 244.0 1111.1
## 76 117.0 1485.4
## 77 101.6  903.2
## 78  70.5  955.3
```

You can read files on your own computer using the same method. Create and save the appropriate text file. In R type `read.table(" ")` and simply drag the file between the quotation marks. The pathname should automatically appear.

read.csv("pathName.csv", header = TRUE)

Details: This command is used to read data from a spreadsheet. To use this function, save the spreadsheet as a CSV file (comma separated values) with name `pathName.csv`. The command `read.csv("pathName.csv", header = TRUE)` will read the file into your R session. Note: By default, `header = TRUE` in contrast to `read.table( )`. If there are no headers, set `header = FALSE` when reading the file. The pathname (address) must be in quotation marks. See Problem 4 at the end of this chapter for an illustrative example.

sum(x), median(x), max(x), min(x)

Details: These basic statistical functions (and many others) work the same way. To apply the function `fcn( )` to the data in `x` use `fcn(x)`. For example, to find the maximum, minimum, and median `Annual` rainfalls, and then the total `Annual` Brisbane rainfall for the data entered earlier, use

```
> max(data.Ex00.2$Annual)         # max annual rainfall

## [1] 2203.7

> min(data.Ex00.2$Annual)         # min annual rainfall

## [1] 555.1

> median(data.Ex00.2$Annual)      # median annual rainfall

## [1] 1102.4

> sum(data.Ex00.2$Annual)         # total rainfall for 78 years

## [1] 89166
```

There are more than 1,000 entries in the data file. You never had to enter them by hand or store the file to your computer. R makes it simple to analyze such data.

Problems

1. Create two data vectors `x` and `y` that contain the integers 1 to 21 and -10 to 10.

(*a*) Add, subtract, multiply, and divide `x` by `y`. What happens when dividing?

(*b*) Find the mean of each data set.

(*c*) Find the corrected sum of squares for each data set.

(*d*) Find the mean of the product of `x` and `y`.

2. The data set `http://waveland.com/Glover-Mitchell/Problem00-2.txt` lists all cities and villages in New York that had a population of more than 10,000 residents in 2010.

(*a*) How many such cities and villages are there?

(*b*) Find the total population of all of these cities and villages.

(*c*) Find the median population of these cities and villages.

(*d*) Find the mean population of these cities and villages.

(*e*) Find the corrected sum of squares for these data.

(*f*) The total population of New York State was 19,378,102 at the time of the census. What proportion of the population lived outside these cities and towns?

3. The data set `http://waveland.com/Glover-Mitchell/Problem00-3.txt` lists the monthly snowfall for Buffalo, NY for each each snow season from 1940–41 to 2013–14.

(*a*) Read the data and determine its structure. Be careful. Data names are case sensitive.

(*b*) Find the median snowfall for December during this period. How does it compare to the mean for December?

(*c*) Find the mean annual snowfall for this period.

(*d*) What were the maximum and minimum annual snowfalls for this period?

(*e*) How many inches of snow fell on Buffalo in this period?

(*f*) Find the corrected sum of squares for the annual snowfall in this period.

(*g*) While writing this in mid-November, 2014 Buffalo received more than 7 feet of snow in three days. How much snow fell in the snowiest November during the period 1940–41 to 2013–14.

4. This exercise illustrates the use of the `read.csv( )` command to read data from spreadsheets. (See the Command Summary for further details.) It also illustrates how to analyze data that do not reside on your own computer. The CSV file `http://slantedwindows.com/demo/iris.csv` contains data for three different species of irises. The data include sepal length and width and petal length and width.

☞ CSV stands for "comma separated values." If you create a data table in a spreadsheet, be sure to save in CSV format, if you intend to analyze it with R.

(*a*) Read the file and store the result in `iris.dat` using

```
> iris.dat <- read.csv("http://slantedwindows.com/demo/iris.csv", header = TRUE)
```

(*b*) List the data. How many entries are there? What is the structure of the file?

(*c*) Determine the mean sepal length and the median petal width.

(*d*) The `tapply( )` function is described in detail in Chapter 8. It is used to apply a function to groups (here, `species`) within a table. What is the output of the following command:

```
> tapply(iris.data$petalW, iris.dat$species, mean)
```

(*e*) Determine the median for each species using `tapply( )`.

1. *Introduction to Data Analysis*

This chapter introduces some of the basic numerical and statistical functions available in R using the examples from the text *An Introduction to Biostatistics* (Glover and Mitchell, Waveland Press, 2015). This guide is not a replacement for the text where concepts are discussed and explained.

Be sure to read Chapter 0 to familiarize yourself with some of the conventions that are used throughout this guide.

Basic Statistics

EXAMPLE 1.1. The table below gives the circumferences at chest height (CCH) (in cm) and their corresponding depths for 15 sugar maples, *Acer saccharum*, measured in a forest in southeastern Ohio. Determine the sample median.

CCH	18	21	22	29	29	36	37	38	56	59	66	70	88	93	120
Depth	1	2	3	4	5	6	7	8	7	6	5	4	3	2	1

SOLUTION. Using the methods outlined in Chapter 0, read in the data file using the `read.table( )` command and put the result into `data.Ex01.1`. As mentioned previously, `header` is set to `TRUE` because each column of data in these online files has a descriptive header.

```
> data.Ex01.1 <- read.table("http://waveland.com/Glover-Mitchell/Example01-1.txt",
+ header = TRUE)
```

As in Example 0.1, notice the + directly above. This is an instance where an entire command does not fit on a single line. The + indicates that it continues to the next line. This is not something that you type; if you hit return before the command is finished, R automatically prints a + to indicate that the command prompt has continued.

It is useful to list the data to make sure the file has been read properly and that the data are as expected. This is done by entering the data table's name.

```
> data.Ex01.1

##    CCH
## 1   18
## 2   21
## 3   22
## 4   29
## 5   29
## 6   36
## 7   37
## 8   38
## 9   56
## 10  59
## 11  66
## 12  70
## 13  88
## 14  93
## 15 120
```

To determine the median use the function `median( )` applied to the data table just created. To refer to the `CCH` data specifically, use the name of the data table followed immediately by a dollar sign $ that is then followed by the header of the column of data to be accessed: `data.Ex01.1$CCH`.

```
> median(data.Ex01.1$CCH)

## [1] 38
```

The median is 38, which is the depth 8 observation. In this example it may seem silly to find the median using R. However, when faced with a large, unordered data set this is very useful.

EXAMPLE 1.2. The table below gives CCH (in cm) for 12 cypress pines, *Callitris preissii*, measured near Brown Lake on North Stradbroke Island. Determine the median.

CCH	17	19	31	39	48	56	68	73	73	75	80	122
Depth	1	2	3	4	5	6	6	5	4	3	2	1

SOLUTION. Find the median by reading in the data and using the median function.

```
> data.Ex01.2 <- read.table("http://waveland.com/Glover-Mitchell/Example01-2.txt",
+ header = TRUE)
> tail(data.Ex01.2, n = 3)         # list a few values

##    CCH
## 10  75
## 11  80
## 12 122

> median(data.Ex01.2$CCH)

## [1] 62
```

EXAMPLE 1.3. The table that follows gives the weights of two samples of albacore tuna, *Thunnus alalunga* (in kg). Find the means, ranges, variance, and standard deviations of both samples.

Sample 1	Sample 2
8.9	3.1
9.6	17.0
11.2	9.9
9.4	5.1
9.9	18.0
10.9	3.8
10.4	10.0
11.0	2.9
9.7	21.2

SOLUTION. Using R, read and list the the data.

```
> data.Ex01.3 <- read.table("http://waveland.com/Glover-Mitchell/Example01-3.txt",
+ header = TRUE)
> data.Ex01.3
```

```
##   Sample1 Sample2
## 1     8.9     3.1
## 2     9.6    17.0
## 3    11.2     9.9
## 4     9.4     5.1
## 5     9.9    18.0
## 6    10.9     3.8
## 7    10.4    10.0
## 8    11.0     2.9
## 9     9.7    21.2
```

As noted in Chapter 0, the mean is computed using the `sum( )` function to add all the data values and the `length( )` function to determine the number of values in each data set.

```
> sum(data.Ex01.3$Sample1)/length(data.Ex01.3$Sample1)    # mean of sample 1

## [1] 10.1111

> sum(data.Ex01.3$Sample2)/length(data.Ex01.3$Sample2)    # mean of sample 2

## [1] 10.1111
```

☞ R Tip: When a command in R is almost identical to a previous command, use the uparrow to navigate to that previous command and then edit it as necessary. This takes the sting out of using long, descriptive data names.

However, R also has a built in function for this purpose, `mean( )`, that works like the `median( )` or `sum( )` functions

```
> mean(data.Ex01.3$Sample1)

## [1] 10.1111

> mean(data.Ex01.3$Sample2)

## [1] 10.1111
```

Remember to report the means to one more significant digit than the original data. This is simple to do by hand (here both means are equal to 10.11 kg). To demonstrate another of R's functions, use `round( )` being sure to specify the number of digits desired after the decimal point. Two digits are required here, so use

```
> round(mean(data.Ex01.3$Sample1), digits = 2)

## [1] 10.11

> round(mean(data.Ex01.3$Sample2), digits = 2)

## [1] 10.11
```

The difference between the largest and smallest observations in a group of data is called the **range**. In R, applying the `range( )` function to each data set

```
> range(data.Ex01.3$Sample1)

## [1]  8.9 11.2

> range(data.Ex01.3$Sample2)

## [1]  2.9 21.2
```

yields the minimum and maximum values of the data, which is another reasonable way to define the range, but not what we expected. To find the range as we have defined it, use R's `max( )` and `min( )` functions.

```
> max(data.Ex01.3$Sample1) - min(data.Ex01.3$Sample1)     # range

## [1] 2.3

> max(data.Ex01.3$Sample2) - min(data.Ex01.3$Sample2)

## [1] 18.3
```

A more widely used measure of dispersion is the **sample variance**, which is defined as

$$\frac{\sum_{i=1}^{n}(X_i - \overline{X})^2}{n-1}.$$

This is just the corrected sum of squares (see Chapter 0 of these notes) divided by $n-1$, where n is the sample size.

EXAMPLE 1.4. Calculate the variance of the sample mean, standard deviation, and standard error for the data sets in Example 1.3.

SOLUTION. Make use of the corrected sums of squares,

```
> CSS1 <- sum((data.Ex01.3$Sample1 - mean(data.Ex01.3$Sample1))^2)
> CSS2 <- sum((data.Ex01.3$Sample2 - mean(data.Ex01.3$Sample2))^2)
> Var1 <- CSS1/(length(data.Ex01.3$Sample1) - 1)  # variance of Sample 1
> Var2 <- CSS2/(length(data.Ex01.3$Sample2) - 1)  # variance of Sample 2
> Vars <- c(Var1, Var2)   # put both variances in a vector for convenience
> Vars

## [1]  0.641111 49.851111
```

The **sample standard deviation** is the positive square root of the sample variance. Making use of the fact that both variances are in `Vars`, both standard deviations can be calculated in a single command.

```
> sqrt(Vars)                       # the standard deviations of Samples 1 and 2

## [1] 0.800694 7.060532
```

As you might expect, `R` actually provides the commands `var( )` and `sd( )` to calculate both the variance and standard deviation in a single step.

```
> var(data.Ex01.3$Sample1)

## [1] 0.641111

> var(data.Ex01.3$Sample2)

## [1] 49.8511

> sd(data.Ex01.3$Sample1)

## [1] 0.800694

> sd(data.Ex01.3$Sample2)

## [1] 7.06053
```

The standard error, SE $= \frac{s}{\sqrt{n}}$, can now be computed. Remember that the `length( )` function can be used to determine the sample size.

```
> sd(data.Ex01.3$Sample1)/sqrt(length(data.Ex01.3$Sample1))       #standard error

## [1] 0.266898

> sd(data.Ex01.3$Sample2)/sqrt(length(data.Ex01.3$Sample2))

## [1] 2.35351
```

Remember to report the variance, standard deviation, and standard error to the correct number of significant digits.

Tables and Graphs

R can be used to make tables and graphs of data sets.

EXAMPLE 1.5. The following table shows the number of sedge plants, *Carex flacca*, found in 800 sample quadrats in an ecological study of grasses. Each quadrat was 1 m^2. Summarize these data in a table with columns for relative frequency, cumulative frequency, and relative cumulative frequency. Produce a bar graph of the data.

Plants/quadrat (X_i)	**Frequency (f_i)**
0	268
1	316
2	135
3	61
4	15
5	3
6	1
7	1

SOLUTION. Read in the original data containing 800 observations. Instead of listing the entire data set, use the `head( )` and `tail( )` functions to list just the first few rows and the last few rows of the data.

Notice that the data in the file are not presented as in the text. Rather, the raw data are arranged much like they might have been recorded in a field notebook where a researcher made a list of the number of plants/quadrat as the quadrats were encountered. This demonstrates how you could take a tablet into the field to make recordings and then analyze the resulting raw data in R.

```
> data.Ex01.5 <- read.table("http://waveland.com/Glover-Mitchell/Example01-5.txt",
+ header = TRUE)
> head(data.Ex01.5)

##   Carex
## 1     0
## 2     1
## 3     2
## 4     1
## 5     1
## 6     3

> tail(data.Ex01.5, n = 3)

##     Carex
## 798     1
## 799     2
## 800     2
```

If no argument other than the file name is used in the `head( )` or `tail( )` functions, then the first or last six rows of the table are listed. To specify an exact number of rows to be listed, use an additional argument such as `n = 3`, as was done above with the `tail( )` function. In this example, the tail function shows that there are 800 observations in the data set, as expected. Now make use of R's `table( )` function.

table( )

The table( ) function takes one or more specified columns (or rows) of data from a table, sorts it by entry, and lists the number of times each entry occurs, that is, it produces a frequency table.

The function table(data.Ex01.4$Carex) takes each different entry in the Carex column and counts how many times that entry occurs. The result of this process is a list of frequency counts for the observed numbers of Carex per quadrat. Put the result into a "list" named counts.

```
> counts <- table(data.Ex01.5$Carex)
> counts

##
##   0   1   2   3   4   5   6   7
## 268 316 135  61  15   3   1   1
```

These are precisely the frequencies given in the original data table. To make a table from these lists put the counts data into a so-called data frame (which we name carex.table) using the data.frame( ) command (see page 9). Then use the colnames( ) command to give the columns in the table the required names. Finally list carex.table.

```
> carex.table <- data.frame(counts)
> colnames(carex.table) <- c("Plants", "Freq")
> carex.table

##   Plants Freq
## 1      0  268
## 2      1  316
## 3      2  135
## 4      3   61
## 5      4   15
## 6      5    3
## 7      6    1
## 8      7    1
```

To obtain the relative frequencies, divide carex.table$Freq by the total number of observations (quadrats), then multiply by 100. Here we happen to know that the number of quadrats is $n = 800$. In general, use the length function to determine n

```
> n <- length(data.Ex01.5$Carex)          # n = the number of quadrats
> 100*carex.table$Freq/n

## [1] 33.500 39.500 16.875  7.625  1.875  0.375  0.125  0.125
```

To add the relative frequencies as a column to carex.table first create a new column in carex.table using carex.table["RelFreq"] and then place the actual relative frequencies into the column.

```
> carex.table["RelFreq"] <- 100*carex.table$Freq/n
```

Now that you know how the commands work, in the future you can do this in a single step as above. Verify that the relative frequencies were added by listing the table.

```
> carex.table

##   Plants Freq RelFreq
## 1      0  268  33.500
## 2      1  316  39.500
## 3      2  135  16.875
## 4      3   61   7.625
## 5      4   15   1.875
## 6      5    3   0.375
## 7      6    1   0.125
## 8      7    1   0.125
```

To add the cumulative frequencies to the table, use the cumulative sum function `cumsum( )` applied to the `carex.table$Freq` column and insert the result into a new column of `carex.table` called `CumFreq` and then list the result.

```
> carex.table["CumFreq"] <- cumsum(carex.table$Freq)
> carex.table

##   Plants Freq RelFreq CumFreq
## 1      0  268  33.500     268
## 2      1  316  39.500     584
## 3      2  135  16.875     719
## 4      3   61   7.625     780
## 5      4   15   1.875     795
## 6      5    3   0.375     798
## 7      6    1   0.125     799
## 8      7    1   0.125     800
```

To add the relative cumulative frequencies to the table, apply the `cumsum( )` function to the `RelFreq` column. Then insert the result into a new column of `carex.table` called `RelCumFreq` and list the result.

☞ You could also compute the relative cumulative frequencies by taking the cumulative frequency column, dividing by n, and multiplying by 100 as we did for the relative frequencies.

```
> carex.table["RelCumFreq"] <- cumsum(carex.table$RelFreq)
> carex.table

##   Plants Freq RelFreq CumFreq RelCumFreq
## 1      0  268  33.500     268     33.500
## 2      1  316  39.500     584     73.000
## 3      2  135  16.875     719     89.875
## 4      3   61   7.625     780     97.500
## 5      4   15   1.875     795     99.375
## 6      5    3   0.375     798     99.750
## 7      6    1   0.125     799     99.875
## 8      7    1   0.125     800    100.000
```

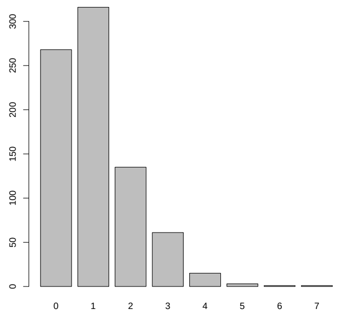


Figure 1.1: The default bar graph for Example 1.5.

To make a bar graph of these data, use R's `barplot( )` command on the *original* `counts` data. When you enter the `barplot( )` command the plot will appear in a new window.

```
> barplot(counts)
```

Another method to create the bar graph is to use the columns from `carex.table`. The syntax in this case is `barplot(height, names.arg = Names)` where `height` is the frequency vector and `names.arg` is the vector of corresponding names. In our case, use

```
> barplot(carex.table$Freq, names.arg = carex.table$Plants)
```

It is good practice to add titles and labels to the plot. To add a title, use `main`. To add x- and y-axis labels use, `xlab` and `ylab` inside the `barplot( )` command. Note that the text of the title and labels must be inside quotation marks.

```
> barplot(counts, main = "Example 1.5", xlab = "Plants/quadrat",
+ ylab = "Frequency")
```

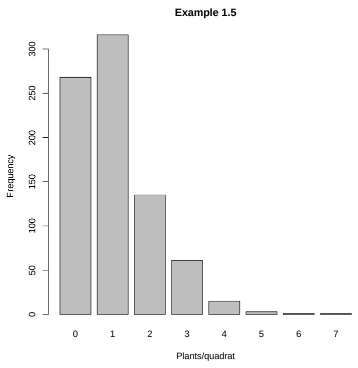


Figure 1.2: A labelled bar graph for Example 1.5.

EXAMPLE 1.6. The following data were collected by randomly sampling a large population of rainbow trout, *Salmo gairdneri*. The variable of interest is weight in pounds. Summarize these data in a table with columns for relative frequency, cumulative frequency, and relative cumulative frequency. Produce a histogram of the data.

X_i (lb)	f_i	f_iX_i	$f_iX_i^2$
1	2	2	2
2	1	2	4
3	4	12	36
4	7	28	112
5	13	65	325
6	15	90	540
7	20	140	980
8	24	192	1536
9	7	63	567
10	9	90	900
11	2	22	242
12	4	48	576
13	2	26	338
Sum	110	780	6158

SOLUTION. The raw data set contains weights of 110 rainbow trout. As usual, first read the data set. Instead of listing the entire data set, use the `head( )` and `tail( )` functions to list a selected number of rows of the data.

```
> data.Ex01.6 <- read.table("http://waveland.com/Glover-Mitchell/Example01-6.txt",
+ header = TRUE)
> head(data.Ex01.6, n = 5)

##   Pounds
## 1     11
## 2     10
## 3      6
## 4      9
## 5      5

> tail(data.Ex01.6, n = 3)

##     Pounds
## 108      7
## 109      6
## 110      7
```

There are 110 observations in the data set, as expected. Use the `table( )` function to produce a list of frequency counts of the different weights in the data set.

☞ Remember to specify that you want the `Pounds` column from `data.Ex01.6`.

```
> counts <- table(data.Ex01.6$Pounds)
> counts

##
##  1  2  3  4  5  6  7  8  9 10 11 12 13
##  2  1  4  7 13 15 20 24  7  9  2  4  2
```

Put `counts` into a data frame (called `trout.table`) using the `data.frame( )` command and use the `colnames( )` command to give the columns in the table the required names.

```
> trout.table = data.frame(counts)
> colnames(trout.table) <- c("Pounds", "Freq")
> tail(trout.table, n = 3)
```

```
##    Pounds Freq
## 11     11    2
## 12     12    4
## 13     13    2
```

Using the same process as in the previous example, make a table with columns for relative frequency, cumulative frequency, and relative cumulative frequency. Obtain the relative frequency by dividing `trout.table$Freq` by the total number of observations n, then multiplying by 100. To add the relative frequencies as a column to the `trout.table`, create a new column in `trout.table` using `trout.table["RelFreq"]` and then place the actual relative frequencies into the column.

```
> n <- length(data.Ex01.6$Pounds) # n = the number of observations
> trout.table["RelFreq"] <- 100*trout.table$Freq/n
```

To add the cumulative frequencies to the table, use the `cumsum( )` function on the `trout.table$Freq` column and insert the result into a new column of `trout.table`.

```
> trout.table["CumFreq"] <- cumsum(trout.table$Freq)
```

To add the relative cumulative frequencies to the table, use the `cumsum( )` function on the `RelFreq` column and then insert the result into a new column of `trout.table`. This completes the table, so list the result.

```
> trout.table["RelCumFreq"] <- cumsum(trout.table$RelFreq)
> trout.table

##    Pounds Freq   RelFreq CumFreq RelCumFreq
## 1       1    2  1.818182       2    1.81818
## 2       2    1  0.909091       3    2.72727
## 3       3    4  3.636364       7    6.36364
## 4       4    7  6.363636      14   12.72727
## 5       5   13 11.818182      27   24.54545
## 6       6   15 13.636364      42   38.18182
## 7       7   20 18.181818      62   56.36364
## 8       8   24 21.818182      86   78.18182
## 9       9    7  6.363636      93   84.54545
## 10     10    9  8.181818     102   92.72727
## 11     11    2  1.818182     104   94.54545
## 12     12    4  3.636364     108   98.18182
## 13     13    2  1.818182     110  100.00000
```

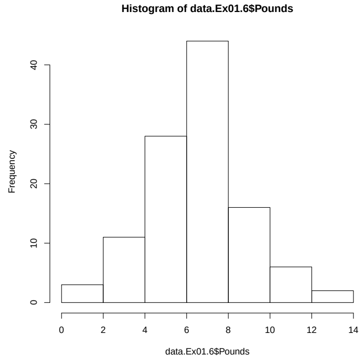


Figure 1.3: The default histogram for the rainbow trout weights in Example 1.6.

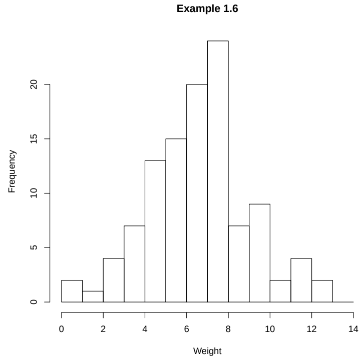


Figure 1.4: A histogram for the rainbow trout weights with breaks for every pound.

To make a histogram of the original frequency data, R has a built-in histogram command `hist( )`.

```
> hist(data.Ex01.6$Pounds)
```

To create breaks in the histogram at every integer from 0 to 14, modify the code just slightly using `breaks = c(0:14)`. Add a title and a label for the x-axis in the usual way.

```
> hist(data.Ex01.6$Pounds, breaks = c(0:14), main = "Example 1.6",
+  xlab = "Weight")
```

EXAMPLE 1.8. The list below gives snowfall measurements for 50 consecutive years (1951–2000) in Syracuse, NY (in inches per year). The data have been rearranged in order of increasing annual snowfall. Create a histogram using classes of width 30 inches and then create a histogram using narrower classes of width 15 inches. (Source: `http://neisa.unh.edu/Climate/IndicatorExcelFiles.zip`)

71.7	73.4	77.8	81.6	84.1	84.1	84.3	86.7	91.3	93.8
93.9	94.4	97.5	97.6	98.1	99.1	99.9	100.7	101.0	101.9
102.1	102.2	104.8	108.3	108.5	110.2	111.0	113.3	114.2	114.3
116.2	119.2	119.5	122.9	124.0	125.7	126.6	130.1	131.7	133.1
135.3	145.9	148.1	149.2	153.8	160.9	162.6	166.1	172.9	198.7

SOLUTION. Read the data and list a few entries

```
> data.Ex01.8 <- read.table("http://waveland.com/Glover-Mitchell/Example01-8.txt",
+ header = TRUE)
> tail(data.Ex01.8, n = 3)

##    Snowfall
## 48    166.1
## 49    172.9
## 50    198.7
```

To create a histogram with breaks every 30 inches, use `breaks = seq(60, 210, by = 30)`, which generates a sequence of numbers starting at 60 and going to 210 by multiples of 30. Add a title and a label for the x-axis in the usual way.

```
> hist(data.Ex01.8$Snowfall, breaks = seq(60, 210, by = 30),
+ main = "Example 1.7a", xlab = "Snowfall (in)")
```

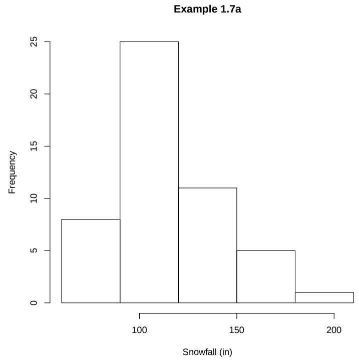


Figure 1.5: A histogram for snowfall in Syracuse, NY with breaks every 30 in.

To create breaks in the histogram every 15 in, modify the code using `by = 15`. The default labels in the first histogram on the x-axis go from 100 to 200. To change it to 60 to 210 is a bit complicated (and is probably not necessary initially). First turn off the default x-axis scale with `xaxt = ’n’`. Then in a second R comand, use the `axis( )` command to take control of the x-axis by setting `side = 1`, put tick marks every 15 units starting at 60 and ending at 210 using `at = seq(60, 210, 15)`, and finally label the ticks using `labels = seq(60, 210, 15)`. The y-axis can be modified in the same way using `side = 2`.

```
> hist(data.Ex01.8$Snowfall, breaks = seq(60, 210, by = 15),
+ main = "Example 1.7b", xlab = "Snowfall (in)", xaxt = 'n')
> axis(side = 1, at = seq(60, 210, by = 15), labels = seq(60, 210, by = 15))
```

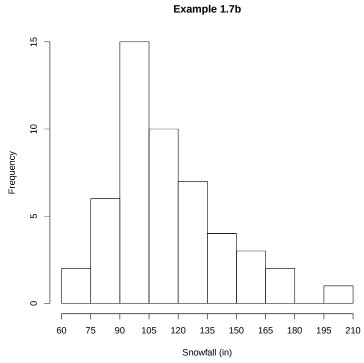


Figure 1.6: A histogram for snowfall in Syracuse, NY with breaks every 15 in.

Quartiles and Box Plots

Five-number summaries are easily carried out in R.

EXAMPLE 1.9. Determine the **five-number summary** for the cypress pine data in Example 1.2.

CCH	17	19	31	39	48	56	68	73	73	75	80	122
Depth	1	2	3	4	5	6	6	5	4	3	2	1

SOLUTION. To produce a five-number summary, read in the data in the usual way and use the `fivenum( )` function.

```
> data.Ex01.9 <- read.table("http://waveland.com/Glover-Mitchell/Example01-9.txt",
+ header = TRUE)
> fivenum(data.Ex01.9$CCH)

## [1]  17  35  62  74 122
```

The five numbers are the minimum, Q_1, the median, Q_3, and the maximum, respectively.

EXAMPLE 1.10. Determine the five-number summary for this sample of 15 weights (in lb) of lake trout caught in Geneva’s Lake Trout Derby in 1994.

Weight		
2.26	3.57	7.86
2.45	1.85	3.88
4.60	4.90	3.60
3.89	2.14	1.52
2.83	1.84	2.12

SOLUTION. Read the data and use the `fivenum( )` function.

```
> data.Ex01.10 <- read.table("http://waveland.com/Glover-Mitchell/Example01-10.txt",
+ header = TRUE)
> fivenum(data.Ex01.10$Weight)

## [1] 1.520 2.130 2.830 3.885 7.860
```

Compare these values to the those given in the text.

Median:			2.83		
Quartiles:		2.12		3.89	
Extremes:	1.52				7.86

There are minor differences in the values of the quartiles calculated by R and those calculated by hand. R uses a method first introduced by Tukey in 1977 while in our text we use a slightly different method. An article by Langford in the *Journal of Statistics Education* (see `http://www.amstat.org/publications/jse/v14n3/langford.html`) indicates that there are at least seven different methods used to determine quartiles! The difference between Tukey's method and Moore's method that we use is the difference between whether the median is included or excluded when calculating the quartiles. While the differences in these calculations are generally small, they can be significant. As Langford mentions, even small differences can be important. He quotes Freund and Perles:

> Before we go into any details, let us point out that the numerical differences between answers produced by the different methods are not necessarily large; indeed, they may be very small. Yet if quartiles are used, say to establish criteria for making decisions, the method of their calculation becomes of critical concern. For instance, if sales quotas are established from historical data, and salespersons in the highest quarter of the quota are to receive bonuses, while those in the lowest quarter are to be fired, establishing these boundaries is of interest to both employer and employee. (Freund and Perles. 1987. A new look at quartiles of ungrouped data. *The American Statistician*, 41(3), 200–203.)

EXAMPLE 1.11. Construct a box plot for the lake trout data in Example 1.10.

SOLUTION. Apply the command `boxplot( )` to the data and add titles or labels just as with bar graphs or histograms.

```
> boxplot(data.Ex01.10$Weight, main = "Example 1.10",
+ ylab = "Weight (lb)")
```

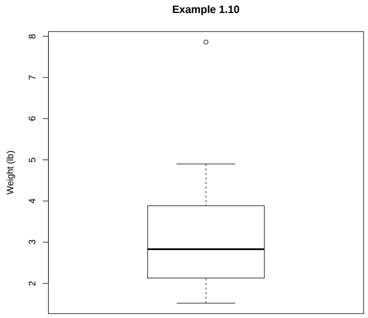


Figure 1.7: A box plot of fish weights for the Geneva Lake Trout Derby.

In the first few examples of this chapter, the depths of each observation were listed. R does not provide a function to calculate depths, but we have provided one, as illustrated below.

PROBLEM 1.0 (Additional Exercise). Calculate the depths of the carapace lengths in Concept Check 2 of Chapter 1.

SOLUTION. First read and list the data file `http://waveland.com/Glover-Mitchell/Problem01-2.txt`.

```
> data.Prob01.2 <- read.table("http://waveland.com/Glover-Mitchell/Problem01-2.txt",
+ header = TRUE)
> data.Prob01.2

##   Length
## 1    4.1
## 2    5.2
## 3    4.3
## 4    5.1
## 5    4.7
## 6    4.5
## 7    3.9
## 8    4.6
## 9    4.3
```

Next download the `depth( )` function using the `source( )` command.

```
> source("http://waveland.com/Glover-Mitchell/depth.txt") # download source file

## Downloaded: depth( ).
```

Now the function is available to use for the rest of your R session. If you close the current session and later open another and want to use this function, you will need to read it in again using `source( )`. Simply apply the `depth( )` function to the data. The result is a table with the original data in ascending order and the corresponding depths.

```
> depth(data.Prob01.2$Length)

##   data.Prob01.2$Length Depth
## 7                  3.9     1
## 1                  4.1     2
## 3                  4.3     3
## 9                  4.3     4
## 6                  4.5     5
## 8                  4.6     4
## 5                  4.7     3
## 4                  5.1     2
## 2                  5.2     1
```

Command Summary

The following are the new commands and functions introduced in Chapter 1.

`barplot(height, names.arg = NULL, main = "Title", xlab = "x-axis label",`
`ylab = "y-axis label")`

Details: `height` is either a vector or matrix of values (frequencies) describing the bars which make up the plot. If `height` is a vector, then the plot consists of a sequence of rectangular bars with heights given by the values in `height`. `names.arg` is a vector of names to be plotted below each bar or group of bars. If this argument is omitted, then the names are taken from the names attribute of `height` if this is a vector, or the column names if it is a matrix. The optional arguments `main = "Title"`, `xlab = "x-axis label"`, and `ylab = "y-axis label"` are used to provide labels for the entire plot, the x-axis, and the y-axis, respectively. Note the labels must be within quotes. There are other optional arguments that can be found online.

```
boxplot(x, y, ..., names, main = "Title", xlab = "x-axis label",
        ylab = "y-axis label")
```

Details: `x, y, ...` are vectors containing the data sets for which the box plots are being drawn. If more than one vector is specified, the box plots are drawn side by side. `names` is a vector of labels, one for each data set, that will be printed under each box plot. The optional arguments `main = "Title"`, `xlab = "x-axis label"`, and `ylab = "y-axis label"` are used to provide labels for the entire plot, the x-axis, and the y-axis, respectively. Note the labels must be within quotes. There are other optional arguments that can be found online.

```
colnames(dataFrame)
```

Details: Used to name the columns in a data frame. For example, `colnames(dataFrame) <- c("Col1", "Col2", "Col3")` renames the columns in `dataFrame` as `Col1`, `Col2`, and `Col3`.

```
depth(data)
```

Details: The `depth( )` function calculates the depths of a column or vector of numeric `data`. The result is a table with the original data in ascending order and the corresponding depths. This function must be downloaded using `source("http://waveland.com/Glover-Mitchell/depth.txt")`.

```
hist(x, breaks = v, main = "Title", xlab = "x-axis label",
     ylab = "y-axis label")
```

Details: `x` is vector containing the data set for which the histogram is being drawn. The optional argument `breaks = v` specifies the break points for the histogram will occur at the values listed in the vector `v`. The optional arguments `main = "Title"`, `xlab = "x-axis label"`, and `ylab = "y-axis label"` are used to provide labels for the entire plot, the x-axis, and the y-axis, respectively. Note that the labels must be within quotes. There are other optional arguments that can be found online.

```
range(dataSet)
```

Details: Returns the minimum and maximum values in `dataSet`. To find the range as defined in *An Introduction to Biostatistics*, use `max(dataSet) - min(dataSet)`.

```
round(x, n = k)
```

Details: Rounds the number `x` to k digits after the decimal point.

```
source("filePathname")
```

Details: The `source( )` command reads an `R` script (set of `R` commands) from a file either on your own computer or from a file on the internet located at `"filePathname"`.

```
table(data)
```

Details: The `table( )` function takes one or more specified columns (or rows) of data from a table, sorts it by entry, and lists the number of times each entry occurs, that is, a frequency table. See Example 1.5.

```
var(dataSet), sd(dataSet)
```

Details: These functions evaluate the variance and standard deviation of the sample `dataSet`.

Problems

1. The data set `http://waveland.com/Glover-Mitchell/Problem01-OldFaithful.txt` lists the time interval (in minutes) between eruptions of Old Faithful for the entire year of 2010.

(*a*) Read the data into a file variable called `OF`.

(*b*) How many eruptions were there in 2010? What function in `R` can you use to determine this?

(*c*) Create a five-number summary of the data.

(*d*) What were the shortest and longest times between eruptions in 2010?

(*e*) Compare the median and mean times between 2010 eruptions.

(*f*) Create a histogram for these data. Label the plot "Distribution of Eruption Intervals for Old Faithful, 2010." Label the *x*-axis as "Minutes between eruptions." (See page 136 for the answer.)

(*g*) Create a box plot for these data.

(*h*) Find the variance and standard deviation for the eruption intervals.

2. Return to data set `http://waveland.com/Glover-Mitchell/Problem00-3.txt` that lists the monthly snowfall for Buffalo, NY for each each snow season from 1940–41 to 2013–14.

(*a*) Print the first and last few rows of the data set.

(*b*) Create a five-number summary of the annual snowfall data.

Names are case sensitive in `R`.

(*c*) Compare the median and mean annual snowfalls for this period

(*d*) Create a histogram of the annual snowfall data. Label the plot "Annual Snowfall Distribution, Buffalo, NY" Label the *x*-axis as "Inches."

(*e*) Create a box plot of the annual snowfall data. Use appropriate labels.

(*f*) Find the variance and standard deviation for the annual snowfall data.

(*g*) Create side-by-side box plots of the monthly snowfalls for November, December, January, and February. Hint: `boxplot( )` will accept more than one data set at a time. Use appropriate labels.

2. *Introduction to Probability*

There are no examples in Chapter 2 that require data sets. Nonetheless, we will use parts of the first four examples to illustrate the use of additional functions in R.

Use of Permutations and Combinations

EXAMPLE 2.1. Eight consumers are asked to rank three brands of Australian beer using 1 for the most preferable and 3 for the least preferable. How many different rankings can occur keeping track of the individual consumers?

SOLUTION. This problem combines a couple of ideas. First each consumer has to rank the three products. That's a permutation question, so there are the symbol $3! = 6$ possible rankings by each consumer. In R we use the `factorial( )` function.

```
> factorial(3)

## [1] 6
```

This ranking process is repeated 8 times, once by each consumer, so there are $(3!)^8$ different rankings. Use ˆ to indicate a power in R.

```
> factorial(3)^8

## [1] 1679616
```

EXAMPLE 2.2. Suppose a die is rolled 3 times and the outcome is recorded for each roll. How many different results can be recorded for the set of three rolls?

SOLUTION. A three-stage experiment with six possible outcomes with repetition allowed:

```
> 6^3

## [1] 216
```

How many sets of results will have a different side up for each roll?

SOLUTION. This is a three-stage experiment without repetition. R has no built in function to calculate the number of such permutations. However, from the definition of ${}_nP_k$ we have ${}_6P_3 = \frac{6!}{(6-3)!}$.

```
> factorial(6)/factorial(6-3)

## [1] 120
```

EXAMPLE 2.3. An insect toxicologist would like to test the effectiveness of three new pyrethroid insecticides on populations of European corn borer moths, *Ostrinia nubilalis*, but she has 7 different geographically isolated strains available. If each strain could be used more than once, how many different tests could she perform?

SOLUTION. A three-stage experiment with repetition permitted: so

```
> 7^3     # ways to test each pyrethroid once

## [1] 343
```

If each strain can be used only once, how many ways could she perform her tests?

SOLUTION. The answer is a permutation, ${}_7P_3$ or

```
> factorial(7)/factorial(7-3)

## [1] 210
```

If the experiment is redesigned to test a single pyrethroid with three of the strains, how many different experiments can be done?

SOLUTION. This reduces to 7 strains choose 3. The answer is a combination, $\binom{7}{3}$. Use

```
> choose(7,3)

## [1] 35
```

EXAMPLE 2.4. In a project to investigate the water quality of Lake Erie, 20 samples were taken. Unknown to the investigators, 5 of the sample containers were contaminated before use. If 10 of the samples are selected at random, what is the probability that 2 of these selected samples will be contaminated?

SOLUTION. We use the counting techniques developed earlier to solve a classical probability problem.

$$P(2 \text{ of the chosen 10 contaminated}) = \frac{\binom{5}{2}\binom{15}{8}}{\binom{20}{10}},$$

so in R use

```
> choose(5,2)*choose(15,8)/choose(20,10)

## [1] 0.348297
```

If ten of the samples are selected at random, what is the probability that none will be contaminated?

SOLUTION. This time we use

$$P(0 \text{ of the chosen 10 contaminated}) = \frac{\binom{5}{0}\binom{15}{10}}{\binom{20}{10}}$$

So in R use

```
> choose(5,0)*choose(15,10)/choose(20,10)

## [1] 0.0162539
```

Command Summary

The following are the new commands and functions introduced in Chapter 2.

choose(n,k)

Details: Returns the value ${}_nC_k = \frac{n!}{k!(n-k)}$ for any non-negative integers n and k with $n \geq k$.

factorial(n)

Details: Returns the value $n!$ for any non-negative integer.

3. Probability Distributions

We now turn to the techniques needed to predict the actions of various **discrete** and **continuous random variables** that are fundamental in the life sciences.

Discrete Random Variables

EXAMPLE 3.1. A fair 6-sided die is rolled with the discrete random variable X representing the number obtained per roll. Give the probability density function for this variable.

SOLUTION. This problem is easier to do by hand. But we use it to illustrate a new `R` command.

`rep(x, times = n)`

Details: `x` is an object (such as a number or vector) and `times` is a non-negative integer representing the number of repetitions of `x`.

Since the die is fair, each of the six outcomes, 1 through 6, is equally likely, so the pdf is

```
> f <- rep(1/6, times = 6)        # or rep(1/6, 6)
> f

## [1] 0.166667 0.166667 0.166667 0.166667 0.166667 0.166667
```

EXAMPLE 3.2. A fair 6-sided die is rolled twice with the discrete random variable X representing the sum of the numbers obtained on both rolls. Give the density function of this variable.

SOLUTION. This problem is relatively easy to do by hand. But here is a way to do it using `R` that illustrates some new functions. First put the possible outcomes into a vector that will be our random variable `X`. Then use the `outer( )` function in `R` to make a table whose entries are the sums of the corresponding sides of the die.

`outer(X, Y, FUN = "*")`

Details: `X` and `Y` are two vectors and `FUN` specifies a function to use on all possible combinations of pairs of values from `X` and `Y`. For example, `outer(X, Y, FUN = "+")` produces a table containing all possible sums of pairs of numbers from `X` and `Y`.

Finish using the `table( )` function (see Chapter 1) to determine the frequency of each outcome.

```
> X <- c(1:6)
> X

## [1] 1 2 3 4 5 6

> two.rolls <- outer(X, X, FUN = "+") # all possible sums of two rolls
> two.rolls

##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    2    3    4    5    6    7
## [2,]    3    4    5    6    7    8
## [3,]    4    5    6    7    8    9
## [4,]    5    6    7    8    9   10
## [5,]    6    7    8    9   10   11
## [6,]    7    8    9   10   11   12

> frequency <- table(two.rolls)   # create the frequency table
> g <- frequency/36       # divide by 36 for the probability (relative frequency)
> round(g, digits = 3)    # round the output to three digits

## two.rolls
##     2     3     4     5     6     7     8     9    10    11    12
## 0.028 0.056 0.083 0.111 0.139 0.167 0.139 0.111 0.083 0.056 0.028
```

EXAMPLE 3.3. Find the expected values for the random variable X in Examples 3.1 and 3.2.

SOLUTION. For Example 3.1,

$$\mu = E(X) = \sum_{x=1}^{6} xf(x)$$

In R use the random variable X from the previous example and the density function f from Example 3.1.

```
> Expected.X <- sum(X*f)
> Expected.X                      # the expected value of X

## [1] 3.5
```

For rolling a die twice as in Example 3.2, first create the random variable of outcomes and then multiply by the density function found previously.

```
> Y <- c(2:12)                    # possible outcomes for two rolls
> Y

##  [1]  2  3  4  5  6  7  8  9 10 11 12

> Expected.Y <- sum(Y*g)          # density g from the previous example
> Expected.Y                      # the expected value of two rolls

## [1] 7
```

EXAMPLE 3.4. Find the expected value of X^2 in Examples 3.1 and 3.2.

SOLUTION. For Example 3.1, making use of earlier calculations

```
> Expected.X.sq <- sum(X^2*f)     # f is the density function from Example 3.1
> Expected.X.sq                   # the expected value of X^2

## [1] 15.1667
```

For Example 3.2,

```
> Expected.Y.sq <- sum(Y^2*g)     # g is the density function for 2 rolls
> Expected.Y.sq                   # the expected value of Y^2

## [1] 54.8333
```

EXAMPLE 3.5. Find the variances for X in Examples 3.1 and 3.2.

SOLUTION. For Example 3.1, $\sigma^2 = E(X^2) - [E(X)]^2$. Using the earlier calculations

```
> varX <- Expected.X.sq - (Expected.X)^2
> varX

## [1] 2.91667
```

For Example 3.2,

```
> varY <- Expected.Y.sq - (Expected.Y)^2
> varY

## [1] 5.83333
```

EXAMPLE 3.6. Breeding pairs of the common robin, *Turdus migratorius*, typically produce a clutch of 3 to 6 eggs. Assume the estimated density function for the random variable X (eggs per clutch) is given in the table below. Based on this information, what is $E(X)$ and what is σ_X?

No. of eggs: x	3	4	5	6
Density: $f(x)$	0.35	0.45	0.16	0.04

SOLUTION. The mean or expected value is $\mu = E(X) = \sum_{x=3}^{6} xf(x)$. The variance is $\sigma^2 = E(X^2) - [E(X)]^2$. Create the appropriate random variable and density functions and use the `sum( )` function.

```
> Eggs <- c(3:6)                  # the sample space
> Eggs

## [1] 3 4 5 6

> f <- c(0.35, 0.45, 0.16, 0.04)
> Expected.Eggs <- sum(Eggs*f)    # mean (expected) number of eggs hatched
> Expected.Eggs

## [1] 3.89

> Expected.Eggs.sq <- sum(Eggs^2*f)
> var.Eggs <- Expected.Eggs.sq - (Expected.Eggs)^2        # variance
> var.Eggs

## [1] 0.6579

> sqrt(var.Eggs)                                          # standard deviation

## [1] 0.81111
```

EXAMPLE 3.7. Find the CDF for Example 3.6, where X was the random variable eggs per clutch for the common robin.

SOLUTION. The cumulative distribution function is easily tabulated from the density function. Simply add all values of f for the values of X less than or equal to x. That's done with the `cumsum( )` function (see Chapter 1).

```
> CDF <- cumsum(f)        # f is the density function in Example 3.6
> CDF

## [1] 0.35 0.80 0.96 1.00
```

We can combine the pdf and CDF into a single table, as one might do for a report. First bind the rows to together using the `rbind( )` command and then add the column names. Always use quote marks around the labels of each column.

```
> Robin <- rbind(f, CDF)  # this binds the two row vectors together
> colnames(Robin) <- c("3", "4", "5", "6")        # mind the quotes
> Robin

##        3    4    5    6
## f   0.35 0.45 0.16 0.04
## CDF 0.35 0.80 0.96 1.00
```

What is the probability of a robin's nest containing 4 *or fewer eggs?*

SOLUTION. This is easy to do by hand, but let's practice some R syntax. Use the `Robin` table. The rows and columns can be referred to by their names, which must be in quotes. $P(X \leq 4) = F(4)$ is

```
> Robin["CDF", "4"]

## [1] 0.8
```

What is the probability of finding strictly between 3 *and* 6 *eggs in a clutch?*

SOLUTION. $P(3 < X < 6) = F(5) - F(3)$ is

```
> Robin["CDF", "5"] - Robin["CDF", "3"]   # remember the quotes

## [1] 0.61
```

The Binomial Distribution

The most important discrete variable distribution in biology is the **binomial distribution**. A binomial random variable probability density function is characterized by the two parameters n, the number of trials or sample size, and p, the probability of success on a trial. R has built in density and cumulative density functions for binomial random variables.

`dbinom(x, n, p = probability)`

Details: `x` is the number of successes; `n` specifies the number of trials; `p = probability` specifies the probability of success, with default `p = 0.5`.

`pbinom(x, n, p = probability)`

Details: The arguments are the same as for `dbinom( )`.

The function `dbinom( )` is used for the pdf for the binomial and `pbinom( )` is used for the CDF.

EXAMPLE 3.8. Assuming that sex determination in human babies follows a binomial distribution, find the density function and CDF for the number of females in families of size 5.

SOLUTION. $P(\text{female}) = P(\text{success}) =$ `p = 0.5`. The number of trials is `n = 5`. Individual values of the pdf and CDF can be computed using

```
> dbinom(3, 5, p = 0.5)           # 3 females in 5 trials

## [1] 0.3125

> pbinom(3, 5, p = 0.5)           # 3 or fewer females in 5 trials

## [1] 0.8125
```

For the entire pdf and CDF, use vector inputs for all possible numbers of successes: `c(0:5)`.

```
> pdf = dbinom(c(0:5), 5, p = 0.5)                # entire pdf
> CDF = pbinom(c(0:5), 5, p = 0.5)                # entire CDF
> pdf

## [1] 0.03125 0.15625 0.31250 0.31250 0.15625 0.03125

> CDF

## [1] 0.03125 0.18750 0.50000 0.81250 0.96875 1.00000
```

We can combine the pdf and CDF into a single table. Bind the rows to together using the `rbind( )` command and then add the column names. Always use quote marks around the labels of each column.

```
> Binom <- rbind(pdf, CDF)        # this binds the two row vectors together
> colnames(Binom) <- c("0", "1", "2", "3", "4", "5")      # mind the quotes
> Binom

##           0       1      2      3       4       5
## pdf 0.03125 0.15625 0.3125 0.3125 0.15625 0.03125
## CDF 0.03125 0.18750 0.5000 0.8125 0.96875 1.00000
```

EXAMPLE 3.9. A particular strain of inbred mice has a form of muscular dystrophy that has a clear genetic basis. In this strain the probability of appearance of muscular dystrophy in any one mouse born of specified parents is $\frac{1}{4}$. If 20 offspring are raised from these parents, find the following probabilities.

(*a*) Fewer than 5 will have muscular dystrophy;

(*b*) Five will have muscular dystrophy;

(*c*) Fewer than 8 and more than 2 will have muscular dystrophy.

SOLUTION. The number of trials is `n = 20` and the probability of success is `p = 1/4`.

(*a*) To determine the probability that fewer than 5 will have muscular dystrophy, use $P(X < 5) = F(4)$.

```
> pbinom(4, 20, p = 1/4)  # pbinom is the CDF

## [1] 0.414842
```

(*b*) To determine the probability that 5 will have muscular dystrophy, use the pdf.

```
> dbinom(5, 20, p = 1/4)  # dbinom is the pdf

## [1] 0.202331
```

(*c*) To determine the probability that fewer than 8 and more than 2 will have muscular dystrophy, use $P(2 < X < 8) = F(7) - F(2)$.

```
> pbinom(7, 20, p = 1/4) - pbinom(2, 20, p = 1/4)

## [1] 0.806928
```

EXAMPLE 3.10. Suppose it is known that the probability of recovery from infection with the Ebola virus is 0.10. If 16 unrelated people are infected, find the following:

(*a*) The expected number who will recover;

(*b*) The probability that 5 or fewer recover;

(*c*) The probability that at least 5 will recover;

(*d*) The probability that exactly 5 will recover.

SOLUTION. The number of trials is `n = 16` and the probability of success is `p = 1/10`.

(*a*) The expected number who will recover is

$$E(X) = \mu = np = 16(0.10) = 1.6.$$

(*b*) The probability that 5 or fewer recover is

```
> pbinom(5, 16, p = 1/10)         # CDF: P(X <= 5) = F(5)

## [1] 0.996703
```

(*c*) The probability that at least 5 will recover is

```
> 1 - pbinom(4, 16, p = 1/10)     # P (X >= 5) = 1 - F(4)

## [1] 0.017004

> pbinom(4, 16,  p = 1/10, lower.tail = FALSE)    # another method

## [1] 0.017004
```

The cumulative density function has an additional argument called `lower.tail`. By default, it is set to `TRUE` so that we get the ordinary density function. Using `pbinom(x, n, p = probability, lower.tail = FALSE)`, the function returns the value `1 - pbinom(x, n, p = probability)`, that is, $1 - F(x)$, which is the same as $P(X > x) = P(X \geq x + 1)$.

(*d*) The probability that exactly 5 will recover is

```
> dbinom(5, 16, p = 1/10)         # pdf: P(X = 5) = f(5)

## [1] 0.0137072
```

The Poisson Distribution

The discrete random variable describing the number of occurrences of an event in a continuous interval of time or space, sometimes called rare, random events, arises from what are known as Poisson processes. The expected number of events during any one period is the same as during any other period of the same length and is denoted by μ. To be able to use a Poisson distribution the value of μ must be known or determined from a sample. R has built in density and cumulative density functions, `dpois( )` and `ppois( )`, respectively.

`dpois(x, mu)`

ppois(x, mu, lower.tail = TRUE)

Details: `x` is the number of successes and `mu` specifies the expected number (mean) of successes in one period. The cumulative density function has an additional argument called `lower.tail`. By default, it is set to `TRUE` so that the left tail of the density function is used. Using `ppois(n, mu.est, lower.tail = FALSE)`, the function returns the value `1 - ppois(n, mu.est)` $= 1 - F(n)$, which is the same as $P(X > n)$ or the right tail of the function.

EXAMPLE 3.11. A radioactive source emits decay particles at an average rate of 4 particles per second. Find the probability that 2 particles are emitted in a 1-second interval.

SOLUTION. Here `mu = 4` particles per second. We want

```
> dpois(2, 4)     # pdf: P(X = 2) = f(2)

## [1] 0.146525
```

EXAMPLE 3.12. An ichthyologist studying the spoonhead sculpin, *Cottus ricei*, catches specimens in a large bag seine that she trolls through the lake. She knows from many years experience that on average she will catch 2 fish per trolling run. Find the probabilities of catching

(*a*) No fish on a particular run.

(*b*) Fewer than 6 fish on a particular run;

(*c*) Between 3 and 6 fish on a particular run.

SOLUTION. `mu = 2` fish/run.

(*a*) For no fish use

```
> dpois(0, 2)     # pdf: P(X= 0) = f(0)

## [1] 0.135335
```

(*b*) For fewer than 6 fish, $P(X < 6) = F(5)$. Use

```
> ppois(5, 2)     # CDF: F(5)

## [1] 0.983436
```

(*c*) For between 3 and 6 fish, $P(3 < X < 6) = F(5) - F(3)$. Use

```
> ppois(5, 2) - ppois(3, 2)

## [1] 0.126313
```

EXAMPLE 3.13. A group of forest ecology students survey several 10-m × 10-m plots in a subtropical rainforest. They find a mean of 30 trees per plot. Under the assumption that the trees are randomly distributed, what is the probability of finding no more than 3 trees in a 1-m^2 plot? Of finding exactly 1 tree in a 1-m^2 plot? At least 2 trees?

SOLUTION. If there were a mean of 30 trees in 100 m^2, the expected number in a 1-m^2 observation plot is

```
> mu.est <- 30/100
> mu.est

## [1] 0.3
```

The probability of finding at most 3 trees in a 1-m^2 plot is

```
> ppois(3, mu.est)        # CDF: F(3)

## [1] 0.999734
```

The probability of finding exactly one tree is

```
> dpois(1, mu.est)        # pdf: f(1)

## [1] 0.222245
```

Using complements, the probability of finding at least two trees is

```
> 1 - ppois(1, mu.est)    # P(X >= 2) = 1 - F(1)

## [1] 0.0369363

> ppois(1, mu.est, lower.tail = FALSE)    # another method

## [1] 0.0369363
```

EXAMPLE 3.14. A certain birth defect occurs with probability $p = 0.0001$. Assume that $n =$ 5000 babies are born at a particular large, urban hospital in a given year.

(*a*) What is the probability that there is at least 1 baby born with the defect?

(*b*) What is the probability that there will be no more than 2 babies born with the defect?

SOLUTION. This is a binomial problem with `n = 5000` and `p = 0.0001`. In R, one should simply make use of the binomial function. However, since n is very large and $np \leq 10$, a Poisson approximation is possible. Here `mu = np = 0.5`. Let's compare the two results.

(*a*) the probability that there is at least 1 baby with the defect is

```
> 1 - pbinom(0, 5000, p = 0.0001)

## [1] 0.393485

> 1 - ppois(0, 1/2)       # compare to the Poisson approximation

## [1] 0.393469
```

(*b*) and the probability that there are no more than 2 with the defect is

```
> pbinom(2, 5000, p = 0.0001)

## [1] 0.985618

> ppois(2, 1/2)           # compare to the Poisson approximation

## [1] 0.985612
```

The Normal Distribution

The normal distributions depend on two parameters, the mean μ and the standard deviation σ. In R the functions `dnorm( )` and `pnorm( )` are used for the pdf and CDF of various normal distributions.

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

pnorm(x, mean = mu, sd = sigma, lower.tail = TRUE)

Details: `x` is a value of the normal random variable X; `mean` specifies the mean of the distribution; `sd` specifies the standard deviation. The default values are `mean = 0` and `sd = 1` that specify the standard normal distribution. The argument `lower.tail` is by default `TRUE`. When set to `FALSE`, `pnorm(x, mean = mu, sd = sigma, lower.tail = FALSE)` returns `1 - pnorm(x, mean = mu, sd = sigma)` or the upper tail of the distribution.

EXAMPLE 3.16. Suppose that the scores on an aptitude test are normally distributed with a mean of 100 and a standard deviation of 10. (Some of the original IQ tests were purported to have these parameters.) What is the probability that a randomly selected score is below 90?

SOLUTION. We must find $P(X < 90) = F(90)$. There is no need to transform to the standard normal distribution. Simply use `mean = 100` and `sd = 10`.

```
> pnorm(90, mean = 100, sd = 10)  # CDF

## [1] 0.158655
```

What is the probability of a score between 90 *and* 115*?*

SOLUTION. We wish to find $P(90 < X < 115)$.

```
> pnorm(115, mean = 100, sd = 10) - pnorm(90, mean = 100, sd = 10)

## [1] 0.774538
```

What is the probability of a score of 125 *or higher?*

SOLUTION. We want $P(X \geq 125)$, the upper tail.

```
> pnorm(125, mean = 100, sd = 10, lower.tail = FALSE)

## [1] 0.00620967
```

EXAMPLE 3.17. Suppose that diastolic blood pressure X in hypertensive women centers about 100 mmHg and has a standard deviation of 16 mmHg and is normally distributed. Find $P(X < 90)$, $P(X > 124)$, $P(96 < X < 104)$. Then find x so that $P(X \leq x) = 0.95$.

SOLUTION. This time `mean = 100` and `sd = 16`. Use

```
> pnorm(90, mean = 100, sd = 16)                                # P(X < 90)

## [1] 0.265986

> pnorm(124, mean = 100, sd = 16, lower.tail = FALSE )          # P(X > 124)

## [1] 0.0668072

> pnorm(104, mean = 100, sd = 16) - pnorm(96, mean = 100, sd = 16) # P(96 < X < 104)

## [1] 0.197413
```

Finally, find x so that $P(X \leq x) = 0.95$. Well, R has a function for that, too.

qnorm(p, mean = mu, sd = sigma)

Details: `p` is the particular cumulative probability of the normal random variable X that we are trying to achieve; `mean` specifies the mean of the distribution; `sd` specifies the standard deviation. The default values are `mean = 0` and `sd = 1` that specify the standard normal distribution.

```
> qnorm(0.95, mean = 100, sd = 16)

## [1] 126.318
```

This means that approximately 95% of these hypertensive women will have diastolic blood pressures less than 126.3 mmHg.

EXAMPLE 3.18. Return to Example 3.9 with a genetic form of muscular dystrophy in mice. This example was solved using binomial probabilities with $p = 0.25$ and $n = 20$. Suppose a larger sample of progeny are generated and we would like to know the probability of fewer than 15 with muscular dystrophy in a sample of 60.

SOLUTION. While it is difficult to calculate this binomial by hand, in R use

```
> pbinom(14, 60, p = 1/4) # fewer than 15 mice have muscular dystrophy

## [1] 0.450569
```

By hand, one would use the normal approximation to the binomial. Since $p = 0.25$ and $n = 60$, it follows that $np = \mu = 60(0.25) = 15$ and $n(1-p) = 60(0.75) = 45$; the normal approximation is acceptable. Since $\sigma^2 = np(1-p) = 60(0.25)(0.75) = 11.25$, then $\sigma = \sqrt{11.25}$. The normal approximation is

```
> pnorm(14.5, mean = 15, sd = sqrt(11.25)) # use 14.5 for continuity correction

## [1] 0.440749
```

When using R, it makes sense to use the exact distribution rather than an approximation, especially since using the exact distribution requires no intermediate calculations.

Command Summary

`dbinom(x, n, p = probability)`

`pbinom(x, n, p = probability, lower.tail = TRUE)`

Details: `x` is the number of successes; `n` specifies the number of trials; `p = probability` specifies the probability of success, with default `p = 0.5`. The function `pbinom( )` has an additional argument `lower.tail` that is by default `TRUE`. If `lower.tail = FALSE`, the function returns `1 - pbinom(x, n, p)` or the upper tail of the distribution. `dbinom( )` is the pdf and `pbinom( )` is the CDF.

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

`pnorm(x, mean = mu, sd = sigma, lower.tail = TRUE)`

Details: `x` is a value of the normal random variable X; `mean` specifies the mean of the distribution; `sd` specifies the standard deviation. The default values are `mean = 0` and `sd = 1` that specify the standard normal distribution. The function `pnorm( )` has an additional argument `lower.tail` that is by default `TRUE`. If `lower.tail = FALSE`, the function returns `1 - pnorm(x, mu, sd)` or the upper tail of the distribution. `dnorm( )` is the pdf and `pnorm( )` is the CDF.

`qnorm(p, mean = mu, sd = sigma)`

Details: `p` is the particular cumulative probability of the normal random variable X that we are trying to achieve; `mean` specifies the mean of the distribution; `sd` specifies the standard deviation. The default values are `mean = 0` and `sd = 1` that specify the standard normal distribution.

`dpois(x, mu)`

`ppois(x, mu, lower.tail = TRUE)`

Details: `x` is the number of successes of interest and `mu` specifies the expected number (mean) of successes in one period. The function `ppois( )` has an additional argument `lower.tail` that is by default `TRUE`. If `lower.tail = FALSE`, the function returns `1 - ppois(x, mu)` or the upper tail of the distribution. `dpois( )` is the pdf and `ppois( )` is the CDF.

`outer(X, Y, FUN = "*")`

Details: `X` and `Y` are two vectors and `FUN` specifies a function to use on all possible combinations of pairs of values from `X` and `Y`. For example, `outer(X, Y, FUN = "+")` produces a table containing all possible sums of pairs of numbers from `X` and `Y`.

`rbind(x, y, ...)`; *`cbind(x, y, ...)`*

Details: `x, y, ...` are vectors of the same length that will be bound together as rows (columns) in a matrix.

`rep(x, times = n)`

Details: `x` is an object (such as a number or vector) and `times` is a non-negative integer representing the number of repetitions of `x`.

4. *Sampling Distributions*

A **simple random sample** is a sample of size n drawn from a population of size N in such a way that every possible sample of size n has the same probability of being selected. Variability among simple random samples drawn from the same population is called **sampling variability**. The probability distribution that characterizes some aspect of the sampling variability, usually the mean but not always, is called a **sampling distribution**. These sampling distributions allow us to make objective statements about population parameters without measuring every object in the population.

EXAMPLE 4.2. Suppose that in a genetics experiment with fruit flies, *Drosophila melanogaster*, a very large progeny is generated from parents heterozygous for a recessive allele causing vestigial wings. Recall from your study of elementary genetics that a cross between heterozygous parents should produce offspring in a 3:1 ratio of dominant to recessive phenotypes:

$$\text{Vv} \times \text{Vv} \longrightarrow \tfrac{3}{4}\text{V}_{-}\ \left(\tfrac{2}{4}\text{Vv};\ \tfrac{1}{4}\text{VV: normal wings}\right), \quad \tfrac{1}{4}(\text{vv: vestigial wings}).$$

Mendelian principles lead to the expectation of 25% vestigial-winged offspring. How many of a random sample of 20 of the progeny would you expect to be vestigial-winged?

SOLUTION. For the binomial with $p = 0.25$ and $n = 20$,

$$E(X) = np = 20(0.25) = 5,$$

so the long-term theoretical average expected in samples of size 20 is 5. But the probability of exactly 5 in a random sample of 20 is

```
> dbinom(5, 20, p = 0.25)

## [1] 0.202331
```

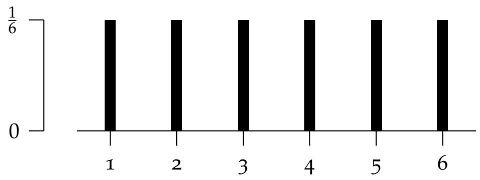
Distribution of the Sample Mean

Perhaps the most important sampling distribution in applied statistics is the distribution of the **sample mean**. Construction of a sampling distribution for the sample mean or any other statistic is a very arduous task for samples of any appreciable size. We do examples here for their heuristic value and not to demonstrate acceptable testing procedures.

EXAMPLE 4.3. Consider a population of size $N = 6$ consisting of the 6 sides of a fair die. The sample space is $\{1, 2, 3, 4, 5, 6\}$. As we demonstrated in Chapter 3, this population has the following parameters:

$$\mu = \frac{\sum X_i}{N} = \frac{21}{6} = 3.5, \qquad \sigma^2 = \frac{\sum (X_i - \mu)^2}{N} = \frac{17.5}{6}.$$

Note the population distribution is a uniform distribution.



Distribution of population

As an exercise in sampling distribution construction, let's draw all possible samples of size $n = 2$ from this population that correspond to rolling the die twice. For each sample drawn calculate the sample mean $\overline{X}$.

SOLUTION. Any of the 6 faces can occur on either roll, so this process corresponds to sampling with replacement. Therefore, the total number of possible samples is $6 \times 6 = 36$. (In general, sampling with replacement n times from a population of size N generates N^n possible samples.) In the previous chapter we generated all possible samples of size $n = 2$ using the `outer( )` function.

```
> X <- c(1:6)              # the sample space
> two.rolls.sum <- outer(X, X, FUN = "+") # sum from two rolls
> two.rolls.sum            # the 36 possible sample sums

##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    2    3    4    5    6    7
## [2,]    3    4    5    6    7    8
## [3,]    4    5    6    7    8    9
## [4,]    5    6    7    8    9   10
## [5,]    6    7    8    9   10   11
## [6,]    7    8    9   10   11   12
```

Since the sample size is $n = 2$, we obtain the sample means by dividing the sample sums in `two.rolls.sum` by 2.

```
> two.rolls.means <- two.rolls.sum/2
> two.rolls.means

##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]  1.0  1.5  2.0  2.5  3.0  3.5
## [2,]  1.5  2.0  2.5  3.0  3.5  4.0
## [3,]  2.0  2.5  3.0  3.5  4.0  4.5
## [4,]  2.5  3.0  3.5  4.0  4.5  5.0
## [5,]  3.0  3.5  4.0  4.5  5.0  5.5
## [6,]  3.5  4.0  4.5  5.0  5.5  6.0

> # the mean of the sampling distribution
> mean.two.rolls <- mean(two.rolls.means)
> mean.two.rolls

## [1] 3.5

> # the variance of the sampling distribution
> var.two.rolls <- sum((two.rolls.means - mean.two.rolls)^2)/length(two.rolls.means)
> var.two.rolls

## [1] 1.45833
```

We finish by creating a bar plot of the distribution of the means of the two rolls of the die. Use the `table( )` function to create a frequency table of the `two.roll.means`

```
> counts <- table(two.rolls.means)
> counts

## two.rolls.means
##   1 1.5   2 2.5   3 3.5   4 4.5   5 5.5   6
##   1   2   3   4   5   6   5   4   3   2   1
```

```
> barplot(counts) # create the bar graph
```

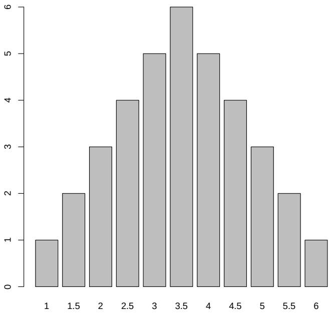


Figure 4.8: The sampling distribution of $\overline{X}$.

EXAMPLE 4.4. The mean blood cholesterol concentration of a large population of adult males (50–60 years old) is 200 mg/dl with a standard deviation of 20 mg/dl. Assume that blood cholesterol measurements are normally distributed. What is the probability that a randomly selected individual from this age group will have a blood cholesterol level below 250 mg/dl?

SOLUTION. Apply `pnorm( )` as in Chapter 3.

```
> pnorm(250, mean = 200, sd = 20)

## [1] 0.99379
```

What is the probability that a randomly selected individual from this age group will have a blood cholesterol level above 225 *mg/dl?*

SOLUTION. Apply `pnorm( )` with the optional argument `lower.tail = FALSE` to determine the upper tail above 225 of the distribution.

```
> pnorm(225, mean = 200, sd = 20, lower.tail = FALSE)

## [1] 0.10565
```

What is the probability that the mean of a sample of 100 *men from this age group will have a value below* 204 *mg/dl?*

SOLUTION. This question requires the understanding and application of Theorem 4.1 from the text. Sample means have a sampling distribution with $\mu_{\overline{X}} = \mu_X = 200$ mg/dl and

$$\sigma_{\overline{X}} = \frac{\sigma_X}{\sqrt{n}} = \frac{20}{\sqrt{100}} = 2.0 \text{ mg/dl},$$

and because the X's were normally distributed, the $\overline{X}$'s will also be normally distributed. This time use `pnorm( )` with `mean = 200` and `sd = 2.0`.

```
> pnorm(204, mean = 200, sd = 2.0)

## [1] 0.97725
```

If a group of 25 *older men who are strict vegetarians have a mean blood cholesterol level of* 188 *mg/dl, would you say that vegetarianism significantly lowers blood cholesterol levels?*

SOLUTION. Use

```
> pnorm(188, mean = 200, sd = 20/sqrt(25))

## [1] 0.0013499
```

Yes, this would be a very rare event if the mean were 200 mg/dl.

EXAMPLE 4.5. Portions of prepared luncheon meats should have pH values with a mean of 5.6 and a standard deviation of 1.0. The usual quality control procedure is to randomly sample each consignment of meat by testing the pH value of 25 portions. The consignment is rejected if the pH value of the sample mean exceeds 6.0. What is the probability of a consignment being rejected?

SOLUTION. Note here there is no mention of the shape of the distribution of pH values for individual portions. They may or may not be normally distributed. The question, however, involves a sample mean, and according to Theorem 4.1 we can assume $\mu_{\overline{X}} = \mu_X = 5.6$ and $\sigma_{\overline{X}} = \frac{\sigma_x}{\sqrt{n}} = \frac{1.0}{\sqrt{25}}$. Also because of the Central Limit Theorem, these $\overline{X}$'s are at least approximately normally distributed. Use

```
> pnorm(6, mean = 5.6, sd = 1.0/sqrt(25), lower.tail = FALSE)

## [1] 0.0227501
```

Only 2.28% of the consignments will be rejected using the quality control procedure.

Confidence Intervals for the Population Mean

EXAMPLE 4.6. Suppose a particular species of understory plants is known to have a variance in heights of 16 cm^2 ($\sigma^2 = 16$ cm^2). If this species is sampled with the heights of 25 plants averaging 15 cm, find the 95% confidence interval for the population mean.

SOLUTION. Here $n = 25$, $\overline{X} = 15$ cm, $\sigma^2 = 16$ cm^2, and $\sigma = 4$ cm. Enter all of this into R and calculate the standard error.

```
> n <- 25                    # sample size
> mx <- 15                   # sample mean
> sd <- 4                    # population sd
> se <- sd/sqrt(n)           # standard error
> se

## [1] 0.8
```

Notice that to determine a confidence interval requires four different numbers:

1. a point estimate, here the sample mean $\overline{X}$;
2. a measure of variability, here the standard error of the mean, $\frac{\sigma}{\sqrt{n}}$;
3. a desired level of confidence $1 - \alpha$, in this case $1 - \alpha = 0.95$, so $\alpha = 0.05$;
4. and the sampling distribution of the point estimate, here the *standard normal distribution*, which provides the confidence factor b with which to adjust the variability for the desired level of confidence, in this case $F(b) = (1 - \frac{\alpha}{2}) = 0.975$.

Using this language, the endpoints of the confidence interval have the form

$$\text{point estimate} \pm \text{(confidence factor)(standard error)}. \tag{4.1}$$

Calculate the factor b using the qnorm( ) function and put it all together as in (4.1).

```
> qnorm(0.975)                          # with the defaults: mean = 0, sd = 1

## [1] 1.95996

> L1 <- mx - qnorm(0.975)*se            # lower endpt
> L2 <- mx + qnorm(0.975)*se            # upper endpt
> c(L1, L2)                             # print both endpoints at once

## [1] 13.432 16.568
```

We are 95% confident that the values 13.43 and 16.57 capture the parametric mean, μ. Let's recalculate this one more time where we think of `qnorm(0.975)*se` as the entire `error` term that we will add or subtract from the mean. This will give us some highly re-usable code and allow us to focus on the key idea: determining the error.

```
> error <- qnorm(0.975)*se
> L1 <- mx - error        # lower endpt
> L2 <- mx + error        # upper endpt
> c(L1, L2)       # print both endpoints at once

## [1] 13.432 16.568
```

What if we want to be more confident (say, 99%) that we've included μ in our interval?

SOLUTION. Now $\alpha = 1 - 0.99 = 0.01$ so $1 - \frac{\alpha}{2} = 0.995$ in the error term. This is the only change required.

```
> error <- qnorm(0.995)*se
> L1 <- mx - error
> L2 <- mx + error
> c(L1, L2)

## [1] 12.9393 17.0607
```

Confidence Intervals for the Mean When the Variance is Unknown

If the value of σ is not known and must be estimated, the distribution of the variable obtained by replacing σ by its estimator is called the ***t* distribution**. `R` provides the same sort of functions to determine t distribution values as it does for the normal distributions.

`dt(x, df)`

`pt(x, df, lower.tail = TRUE)`

Details: The function `dt( )` is used for the pdf and `pt( )` is used for the CDF. `x` is a value of the t statistic and `df` specifies the degrees of freedom. The optional argument `lower.tail` is by default set to `TRUE` so that the left tail of the density function is used. Using `lower.tail = FALSE`, the function returns the value `1 - pt(x, df)`, which is the upper or the right tail of the function.

`qt(p, df, lower.tail = TRUE)`

Details: `p` is the desired probability to be achieved; `df` specifies the degrees of freedom, and the optional argument `lower.tail` specifies that the probability is calculated in the lower tail of the distribution when `TRUE`, which is the default. For an upper tail calculation set `lower.tail = FALSE`.

EXAMPLE 4.7. With $n = 15$, find t_0 such that $P(-t_0 \leq t \leq t_0) = 0.90$.

SOLUTION. Use `qt( )` to determine the required t values.

```
> qt(0.05, df = 14)       # 0.05 = alpha/2

## [1] -1.76131

> qt(0.95, df = 14)       # 1 - alpha/2
```

```
## [1] 1.76131

> qt(0.05, df = 14, lower.tail = FALSE)   # another method

## [1] 1.76131
```

Both values can be calculated at once by using `c(0.05, 0.95)` as the input.

```
> qt(c(0.05, 0.95), df = 14)

## [1] -1.76131  1.76131
```

Find t_0 *such that* $P(-t_0 \leq t \leq t_0) = 0.95$.

SOLUTION. Again df $= 14$, but now $\frac{\alpha}{2} = \frac{1-0.95}{2} = 0.025$.

```
> qt(c(0.025, 0.975), df = 14)

## [1] -2.14479  2.14479
```

EXAMPLE 4.8. Return to the measurements of understory plants introduced in Example 4.6. Without any prior knowledge of the plant heights, the ecologist samples 25 plants and measures their heights. He finds that the sample has a mean of 15 cm ($\overline{X} = 15$ cm) and a sample variance of 16 cm^2 ($s^2 = 16$ cm^2). Note here the sample size, sample mean, and sample variance are all determined by the sampling process. What is the 95% confidence interval for the population mean μ?

SOLUTION. Use a process similar to that for confidence intervals when the variance was known, but use the t distribution since the variance is estimated from the sample.

```
> n <- 25                                 # sample size
> mx <- 15                                # sample mean
> sd <- sqrt(16)                          # population sd
> se <- sd/sqrt(n)                        # standard error
> error <-  qt(0.975, df = n - 1)*se      # error
> L1 <- mx - error                        # lower endpt
> L2 <- mx + error                        # upper endpt
> c(L1, L2)

## [1] 13.3489 16.6511
```

The plant ecologist is 95% confident that the population mean for heights of these understory plants is between 13.349 and 16.651 cm. If he chooses a 99% confidence interval, a single change is required to the error term.

```
> error <-  qt(0.995, df = n - 1)*se
> L1 <- mx - error
> L2 <- mx + error
> c(L1, L2)

## [1] 12.7624 17.2376
```

Confidence Intervals for the Population Variance

Just as we would like to know the population mean μ without measuring every individual in the population, we would also like to have some clear understanding of the population variance σ^2. For a sample we can calculate the point estimate s^2, but each sample will generate a different s^2 and we don't know the sampling distribution that would be generated by the repeated sampling of the population.

To construct a confidence interval for the population variance, we need a random variable that involves this parameter in its expression and whose sampling distribution is well characterized. Fortunately,

$$\frac{(n-1)s^2}{\sigma^2}$$

is such a random variable. If all possible samples of size n are drawn from a normal population with a variance equal to σ^2 and for each of these samples the value $\frac{(n-1)s^2}{\sigma^2}$ is computed, these values will form a *sampling distribution* called a $\boldsymbol{\chi^2}$ **distribution** (chi-square) with $\nu = n-1$ degrees of freedom. These distributions each have

$$\mu = E(\chi^2) = \nu \quad \text{and} \quad \text{Var}(\chi^2) = 2\nu.$$

R provides the same sort of functions to determine χ^2 distribution values as it does for the t distributions.

dchisq(x, df)

pchisq(x, df, lower.tail = TRUE)

Details: The function dchisq( ) is used for the pdf and pchisq( ) is used for the CDF of the distribution. x is a value of the χ^2 statistic and df specifies the degrees of freedom. The optional argument lower.tail is by default set to TRUE so that the left tail of the density function is used. Using lower.tail = FALSE, the function returns the value 1 - pchisq(x, df), which is the upper or the right tail of the function.

qchisq(p, df, lower.tail = TRUE)

Details: p is the desired probability to be achieved; df specifies the degrees of freedom, and the optional argument lower.tail specifies that the probability is calculated in the lower tail of the distribution when TRUE, which is the default. For an upper tail calculation set lower.tail = FALSE.

EXAMPLE 4.9. A horticultural scientist is developing a new variety of apple. One of the important traits, in addition to taste, color, and storability, is the uniformity of the fruit size. To estimate the variability in weights she samples 25 mature fruit and calculates a sample variance for their weights, $s^2 = 4.25$ g^2. Develop 95% and 99% confidence intervals for the population variance from her sample.

SOLUTION. The endpoints of a $(1-\alpha)100\%$ confidence interval for the population variance σ^2 are given by

$$L_1 = \frac{(n-1)s^2}{\chi^2_{1-\frac{\alpha}{2}}} \qquad \text{and} \qquad L_2 = \frac{(n-1)s^2}{\chi^2_{\frac{\alpha}{2}}}.$$

where the chi-square distribution has $n-1$ degrees of freedom.

```
> n <- 25                           # sample size
> df <- n - 1                       # degrees of freedom
> var <- 4.25                       # variance
> L1 <- df*var/qchisq(0.975, df)  # lower endpt for 95% conf int
> L2 <- df*var/qchisq(0.025, df)  # upper endpt for 95% conf int
> c(L1, L2)                         # 95% confidence interval for the variance
```

```
## [1] 2.59120 8.22504

> L1.99 <- df*var/qchisq(0.995, df)       # 1 - alpha/2 = 0.005
> L2.99 <- df*var/qchisq(0.005, df)
> c(L1.99, L2.99)                         # 99% confidence interval

## [1]  2.23888 10.31738
```

Confidence Intervals for a Population Proportion

Confidence intervals for many other parametric statistics can be developed, if the sampling distribution is known or can be reasonably approximated. Consider now the confidence interval for p, the population proportion in a binomial distribution.

EXAMPLE 4.10. A local epidemiologist wishes to determine the rate of breast cancer in women under age 35 in a rural county in Ireland. She surveys a random sample of 1200 women in this age group and determines that exactly 6 have had this form of cancer sometime during their lifetime. She wishes to use this information to estimate the population rate p of breast cancer and determine a confidence interval for this estimate.

SOLUTION. Recall that a good approximation for the $(1-\alpha)100\%$ confidence limits of a population proportion when $n\hat{p} > 5$ and $n(1-\hat{p}) > 5$ is given by

$$L_1 = \hat{p} - z_{1-\frac{\alpha}{2}}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \qquad \text{and} \qquad L_2 = \hat{p} + z_{1-\frac{\alpha}{2}}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}.$$

The process should be familiar now. Determine the point estimate and the error.

```
> n <- 1200                         # sample size
> p <- 6/1200                       # sample proportion
> se <-sqrt(p*(1 - p)/n)            # standard error
> error <-  qnorm(0.975)*se         # error
> L1 <- p - error                   # lower endpt
> L2 <- p + error                   # upper endpt
> c(L1, L2)                         # 95% confidence interval

## [1] 0.00100925 0.00899075
```

Choosing Sample Sizes

In some situations a researcher may wish to design a study that will produce a confidence interval for the population proportion p of a certain pre-specified width. This can be accomplished by using a sufficiently large sample size, n.

EXAMPLE 4.11. Using county medical records, the epidemiologist now wishes to determine the five-year survival rate, p, of all women diagnosed with breast cancer in the 1970s. She would like to determine p to within 2%. More precisely, she wishes to construct a 95% confidence interval whose endpoints are within 2% of the five-year survival rate. How large of a sample will she need to do this?

SOLUTION. Recall that a $(1-\alpha)100\%$ confidence interval for a population proportion p will have a margin of error no greater than the pre-specified value m, if the sample size is

$$n = \left(\frac{z_{1-\frac{\alpha}{2}}}{2m}\right)^2.$$

Since the epidemiologist wants a 95% confidence interval with a 2% margin of error, $\alpha = 0.05$ and $m = 0.02$. Thus,

```
> n <- (qnorm(0.975)/(2*0.02))^2  # required sample size
> n

## [1] 2400.91
```

Command Summary

`dt(x, df)`

`pt(x, df, lower.tail = TRUE)`

Details: The function `dt( )` is used for the pdf and `pt( )` is used for the CDF. `x` is a value of the *t* statistic and `df` specifies the degrees of freedom. The optional argument `lower.tail` is by default set to `TRUE` so that the left tail of the density function is used. Using `lower.tail = FALSE`, the function returns the value `1 - pt(x, df)`, which is the upper or the right tail of the function.

`qt(p, df, lower.tail = TRUE)`

Details: `p` is the desired probability to be achieved; `df` specifies the degrees of freedom, and the optional argument `lower.tail` specifies that the probability is calculated in the lower tail of the distribution when `TRUE`, which is the default. For an upper tail calculation set `lower.tail = FALSE`.

`dchisq(x, df)`

`pchisq(x, df, lower.tail = TRUE)`

Details: The function `dchisq( )` is used for the pdf and `pchisq( )` is used for the CDF of the distribution. `x` is a value of the χ^2 statistic and `df` specifies the degrees of freedom. The optional argument `lower.tail` is by default set to `TRUE` so that the left tail of the density function is used. Using `lower.tail = FALSE`, the function returns the value `1 - pchisq(x, df)`, which is the upper or the right tail of the function.

`qchisq(p, df, lower.tail = TRUE)`

Details: `p` is the desired probability to be achieved; `df` specifies the degrees of freedom, and the optional argument `lower.tail` specifies that the probability is calculated in the lower tail of the distribution when `TRUE`, which is the default. For an upper tail calculation set `lower.tail = FALSE`.

6. *One-Sample Tests of Hypothesis*

Hypotheses Involving the Mean (μ)

We know from Chapter 4 that sample means have a sampling distribution known as the t distribution, and that probabilities for this distribution for various sample sizes are well known. This knowledge and R makes hypothesis tests on the parametric mean relatively straightforward, as illustrated in the following examples.

The function `t.test( )` is used to carry out a t test. We present a simple form of `t.test( )` in this chapter and include additional options in the next chapter.

`t.test(x, mu = 0, alternative = "two.sided", conf.level = 0.95)`

Details: `x` is the vector or table column containing the sample data. The optional argument `mu` gives the hypothesized value of μ; the default is `mu = 0`. The optional argument `alternative` gives the alternative hypothesis for the difference between $\overline{X}$ and μ. The default is `"two-sided"` with the other possible choices being `"less"` or `"greater"`. Finally, the argument `conf.level` gives the confidence level to be used in the t test; the default value of `0.95` is equivalent to $\alpha = 0.05$.

EXAMPLE 6.1. A forest ecologist, studying regeneration of rainforest communities in gaps caused by large trees falling during storms, read that stinging tree, *Dendrocnide excelsa*, seedlings will grow 1.5 m/yr in direct sunlight in such gaps. In the gaps in her study plot she identified 9 specimens of this species and measured them in 2005 and again 1 year later. Listed below are the changes in height for the 9 specimens. Do her data support the published contention that seedlings of this species will average 1.5 m of growth per year in direct sunlight?

1.9   2.5   1.6   2.0   1.5   2.7   1.9   1.0   2.0

SOLUTION. The ecologist is looking for deviations from 1.5 m in either direction, so this is a two-tailed test:

$$H_0\colon \mu_d = 1.5 \text{ m/year}$$
$$H_a\colon \mu_d \neq 1.5 \text{ m/year.}$$

Read and list the data in `http://waveland.com/Glover-Mitchell/Example06-1.txt`.

```
> data.Ex06.1 <- read.table("http://waveland.com/Glover-Mitchell/Example06-1.txt",
+ header = TRUE)
> data.Ex06.1

##   Difference
## 1        1.9
## 2        2.5
## 3        1.6
## 4        2.0
## 5        1.5
## 6        2.7
## 7        1.9
## 8        1.0
## 9        2.0
```

To carry out a t test on these data, use `t.test( )` with `mu = 1.5`, the default two-sided hypothesis, and the default `conf.level = 0.95` (or $\alpha = 0.05$).

```
> t.test(data.Ex06.1$Difference, mu = 1.5)

##
##  One Sample t-test
##
## data:  data.Ex06.1$Difference
## t = 2.3534, df = 8, p-value = 0.04643
## alternative hypothesis: true mean is not equal to 1.5
## 95 percent confidence interval:
##  1.50805 2.29195
## sample estimates:
## mean of x
##       1.9
```

The `t.test( )` function provide two ways to answer whether μ significantly differs from 1.5 m/year. The P value is 0.046, which is smaller than $\alpha = 0.05$. So there is evidence to reject the null hypothesis in favor of the alternative that the mean is different from 1.5 m/year. Equivalently, `t.test( )` provides a confidence interval for the mean μ. If the hypothesized value (here we expected $\mu = 1.5$) falls within the confidence interval, then the null hypothesis is retained. Otherwise, the null hypothesis is rejected. In this example, the 95 percent confidence interval $[1.508, 2.292]$ does not contain the expected value of μ. There is evidence to reject the null hypothesis in favor of the alternative that the mean is different from 1.5 m/year.

EXAMPLE 6.2. Documents from the whaling industry of the Windward Islands in the Lesser Antilles indicate that in the 1920s short-finned pilot whales weighed an average of 360 kg. Cetologists believe that overhunting and depletion of prey species has caused the average weight of these whales to drop in recent years. A modern survey of this species in the same waters found that a random sample of 25 individuals had a mean weight of 336 kg and a standard deviation of 30 kg. Are pilot whales significantly lighter today than during the 1920s?

SOLUTION. The question anticipates a deviation from the claimed value *in a particular direction*, that is, a decrease in weight. This is a left-tailed test with hypotheses

$$H_0\colon \mu \geq 360 \text{ kg}$$
$$H_a\colon \mu < 360 \text{ kg}.$$

R does not provide a function to carry out a t test when the original data are not given. However, we have created a variation on the `t.test( )` function called

```
t.test2(mx, sx, nx, mu = 0, alternative = "two.sided", conf.level = 0.95)
```

Details: `mx`, `sx`, and `nx` are the mean, standard deviation, and sample size for the sample `x`. The optional argument `mu` is the expected value of the mean. The default value is `mu = 0`. The optional argument `alternative` is the alternative hypothesis. The default is `"two-sided"` with the other possible choices being `"less"` or `"greater"`. Finally, the argument `conf.level` gives the confidence level to be used in the t test; the default value is `0.95`, which is equivalent to $\alpha = 0.05$. There are additional arguments that are discussed in the next chapter.

You can download this function using the `source( )` command. The function only needs to be downloaded once per session.

```
>  source("http://waveland.com/Glover-Mitchell/t.test2.txt")

## Downloaded: t.test2( ).
```

In this example the mean, standard deviation, and sample size are `mx = 336`, `sx = 30`, and `nx = 25`, respectively. The alternative hypothesis above is that $\mu < 360$. This means that `mu = 360` and `alternative = "less"`.

```
> t.test2(mx = 336, sx = 30, nx = 25, mu = 360, alternative = "less")

##
##  One Sample t-test
##
## data:  mx = 336, sx = 30, and nx = 25
## t = -4, df = 24, p-value = 0.0002635
## alternative hypothesis: true mean is less than 360
## 95 percent confidence interval:
##     -Inf 346.265
## sample estimates:
## mean of x
##       336
```

`t.test2( )` provides the same two ways as `t.test( )` to answer whether $\mu = 360$. The P value of 0.00026, which is much smaller than $\alpha = 0.05$, indicates that the null hypothesis should be rejected. Equivalently, the 95 percent confidence interval for μ is $(-\infty, 346.265]$. This interval does not include the hypothesized mean of 360. There is evidence that the mean mass of the pilot whales is smaller today than in the 1920s.

The confidence interval here is one-sided to match the t test. We have not discussed one-sided confidence intervals in the text. Briefly, a one-sided 95 percent confidence interval gives the values of μ which lie in the lower (or upper) 95 percent of the t distribution determined by the sample. In other words, the confidence interval describes all values of μ that would *not* lead to rejection of H_0 in the corresponding one-sided t test. In this case $\mu = 360$ is not among the values, so H_0 is rejected.

EXAMPLE 6.3. To test the effectiveness of a new spray for controlling rust mites in orchards, a researcher would like to compare the average yield for treated groves with the average displayed in untreated groves in previous years. A random sample of 16 one-acre groves was chosen and sprayed according to a recommended schedule. The average yield for this sample of 16 groves was 814 boxes of marketable fruit with a standard deviation of 40 boxes. Yields from one acre groves in the same area without rust mite control spraying have averaged 800 boxes over the past 10 years. Do the data present evidence to indicate that the mean yield is sufficiently greater in sprayed groves than in unsprayed groves?

SOLUTION. Anticipating a particular change, an *increase* in yield, means that a right-tailed (`alternative = "greater"`) test should be carried out:

$$H_0\colon \mu \leq 800 \text{ boxes}$$
$$H_a\colon \mu > 800 \text{ boxes.}$$

Without the original data, use `t.test2( )` with sample mean `mx = 814`, sample standard deviation `sx = 40`, sample size `nx = 16`, and hypothesized mean `mu = 800`.

```
> t.test2(mx = 814, sx = 40, nx = 16, mu = 800, alternative = "greater")

##
##  One Sample t-test
##
## data:  mx = 814, sx = 40, and nx = 16
## t = 1.4, df = 15, p-value = 0.09093
## alternative hypothesis: true mean is greater than 800
## 95 percent confidence interval:
##  796.469     Inf
## sample estimates:
## mean of x
##       814
```

The P value of 0.09093 is greater than $\alpha = 0.05$ indicating that the null hypothesis cannot be rejected. Equivalently, the 95 percent confidence interval for μ is $[796.469, \infty)$. This interval contains the hypothesized mean of 800 boxes, so there is not sufficient evidence that mean yield is significantly greater in sprayed groves than in unsprayed groves.

Hypotheses Involving the Variance (σ^2)

Sometimes the question asked about a sample is not its central tendency but its variability or scatter. This question requires one to frame very different H_0's and H_a's and to rely on a different index or test statistic to differentiate between them.

EXAMPLE 6.5. Economy-sized boxes of cornflakes average 40 oz with a standard deviation of 4.0 oz. In order to improve quality control a new packaging process is developed that you hope will significantly decrease variability. Thirty boxes packed by the new process are weighed and have a standard deviation of 3.0 oz. Is there evidence that the cornflakes boxes are significantly more uniform when packed by the new process?

SOLUTION. Recall that H_0 and H_a are written in terms of the *variance* because we have a sampling distribution for variances but not for standard deviations:

$$H_0\colon \sigma^2 \geq 16.0 \text{ oz}^2$$
$$H_a\colon \sigma^2 < 16.0 \text{ oz}^2.$$

Here $n = 30$ and $s^2 = 9.0$ oz^2. The statistic to test H_0 is

$$\chi^2 = \frac{(n-1)s^2}{\sigma^2},$$

and it approximates the **chi-square** statistic that was introduced when the confidence interval for σ^2 was developed in Chapter 4. This statistic will be close to $n-1$ when s^2 is close to the hypothesized value σ^2. Values of χ^2 close to or above $n-1$ support H_0. If s^2 is significantly smaller than σ^2, the χ^2 value would be much smaller than $n-1$ and H_a would be supported. To carry out this calculation in R, first evaluate the test statistic which we will call `chi.sq`.

```
> chi.sq <- (30-1)*9/16
> chi.sq

## [1] 16.3125
```

To obtain the corresponding P value, instead of looking up the result in a table, use the function `pchisq( )` to do the look-up. (See Chapter 4.) In this example a left-tailed test is being used, so the default `lower.tail = TRUE` is used.

```
> pchisq(chi.sq, df = 29)

## [1] 0.0281868
```

The P value is 0.0282, which is less than $\alpha = 0.05$. Reject H_0, while being willing to accept the risk of a Type I error at 0.0282.

EXAMPLE 6.6. Dugongs are large, docile marine mammals that normally occur in herds of about 50 animals. Because the populations are being restricted by human activity along the southeastern coast of Queensland the groups are now somewhat smaller and, therefore, thought to be more highly inbred. Inbreeding often leads to an increase in the variability of the size of animals among the various herds. Healthy dugong populations have a mean weight of 350 kg and a variance of 900 kg^2. A sample of 25 Moreton Bay dugongs had a variance of 1600 kg^2. Is the Moreton Bay population significantly more variable than standard populations?

SOLUTION. Anticipating deviations that are larger than the claimed value makes this a right-tailed test:

$$H_0\colon \sigma^2 \leq 900 \text{ kg}^2$$
$$H_a\colon \sigma^2 > 900 \text{ kg}^2.$$

Here $n = 25$ and $s^2 = 1600$ kg^2. Under H_0 we expect the χ^2 statistic to be $n - 1 = 24$ or smaller. Carry out the calculations in R as in the previous example. Since the test is right-tailed or upper-tailed, set `lower.tail = FALSE`.

```
> chi.sq <- (25-1)*1600/900
> chi.sq

## [1] 42.6667

> pchisq(chi.sq, df = 24, lower.tail = FALSE)

## [1] 0.0108548
```

The P value is 0.0109, which is less than $\alpha = 0.05$. The test statistic is extreme enough to reject H_0 and accept H_a. The analysis supports the view that the Moreton Bay population is significantly more variable than standard populations.

EXAMPLE 6.7. A geneticist interested in human populations has been studying growth patterns in American males since 1900. A monograph written in 1902 states that the mean height of adult American males is 67.0 inches with a standard deviation of 3.5 inches. Wishing to see if these values have changed over the twentieth century the geneticist measured a random sample of 28 adult American males and found that $\overline{X} = 69.4$ inches and $s = 4.0$ inches. Are these values significantly different at the $\alpha = 0.01$ level from the values published in 1902?

SOLUTION. There are two questions here—one about the mean and a second about the standard deviation or variance. Two questions require two sets of hypotheses and two tests.

For the question about means, the hypotheses are

$$H_0\colon \mu = 67.0 \text{ inches}$$
$$H_a\colon \mu \neq 67.0 \text{ inches.}$$

Use `t.test2( )`, where the mean, standard deviation, and sample size for the sample are `mx = 69.4`, `sx = 4.0`, and `nx = 28`, respectively. The test is two-sided so the default `alternative = "two.sided"` is used. The hypothesized mean is `mu = 67` and finally, since $\alpha = 0.01$, set `conf.level = 0.99`.

```
> t.test2(mx = 69.4, sx = 4.0, nx = 28, mu = 67, conf.level = 0.99)

##
##  One Sample t-test
##
## data:  mx = 69.4, sx = 4, and nx = 28
## t = 3.1749, df = 27, p-value = 0.003726
## alternative hypothesis: true mean is not equal to 67
## 99 percent confidence interval:
##  67.3056 71.4944
## sample estimates:
## mean of x
##      69.4
```

The P value is 0.0037 and is smaller than $\alpha = 0.01$. Equivalently, the hypothesized mean of 67.0 in does not lie in the 99 percent confidence interval $[67.306, 71.494]$. Reject H_0 and say that the modern mean is significantly different from that reported in 1902.

For the question about variances, the hypotheses are

$$H_0\colon \sigma^2 = 12.25 \text{ inches}^2$$
$$H_a\colon \sigma^2 \neq 12.25 \text{ inches}^2.$$

The question about variability is answered with a chi-square statistic. Here $n = 28$ and since $s = 4.0$, $s^2 = 16.0$. So

```
> chi.sq <- (28-1)*(4.0)^2/12.25
> chi.sq

## [1] 35.2653
```

This is a two-tailed test. We want to know whether this value of the chi-square statistic is in the extreme tail of either end of the distribution. However, the `pchisq( )` function only calculates the probabilities for a lower tail or an upper tail. The calculated value of the chi square statistic is 35.27, which exceeds the expected value $E(\chi^2) = \text{df} = 27$. So the calculated value is in the *upper tail* of the distribution. To calculate a probability for a two-sided test, double this upper-tail probability.

```
> 2*pchisq(chi.sq, df = 27, lower.tail = FALSE)

## [1] 0.264544
```

The P value is 0.265, so this indicates that the result is not a rare event. For this example we would conclude that the mean height of adult American males is greater now than reported in 1902, but the variability in heights is not significantly different today than in 1902.

The One-Sample Sign Test

In some investigations, a population median is known or is suspected from earlier work. A question arises as to whether some new population is, in fact, identical to the known population. In the text you saw that the sign test can be used to answer such questions. `R` does not directly provide a sign test function. However, we have provided one that you may use.

`sign.test(x, md = 0, alternative = "two.sided", conf.level = 0.95)`

Details: `x` is the vector or table column containing the sample data. The optional argument `md` gives the hypothesized value of the median M to which the individual observations will be compared; the default is `md = 0`. The optional argument `alternative` gives the alternative hypothesis for the difference between an observed value and M. The default is `"two-sided"` with the other possible choices being `"less"` or `"greater"`. Finally, the argument `conf.level` gives the confidence level to be used in the sign test; the default value of `0.95` is equivalent to $\alpha = 0.05$.

EXAMPLE 6.8. Southeastern Queensland is home to more than 40 species of frogs. Among them are several species of *Crinia*, which are small, highly variable brownish or grayish froglets usually found in soaks, swamps, or temporary ponds. One way to distinguish these froglets is by their lengths. A population of froglets from a single pond appear to be of the same species. You tentatively identify them as clicking froglets *C. signifera*, a species known to have a median length of 30 mm. To be honest, you are not really sure that's what they are. A random sample of 18 froglets is captured and their lengths (in mm) are recorded in the table below. Is it reasonable to assume that these are clicking froglets?

24.1	22.9	23.0	26.1	25.0	30.8	27.1	23.2	22.8
23.7	24.6	30.3	23.9	21.8	28.1	25.4	31.2	30.9

SOLUTION. A two-tailed test is appropriate since you know nothing about these froglets, so `alternative = "two.sided"`, which is the default. H_0: $M = 30$ mm versus H_a: $M \neq 30$ mm. So `md = 30`. Download the function using the `source( )` command. This only needs to be done once per session.

```
> source("http://waveland.com/Glover-Mitchell/sign.txt")

## Downloaded: sign.test( ).
```

Read in and list the data file `http://waveland.com/Glover-Mitchell/Example06-8.txt`.

```
> data.Ex06.8 <- read.table("http://waveland.com/Glover-Mitchell/Example06-8.txt",
+ header = TRUE)
> head(data.Ex06.8, n = 2)

##   Length
## 1   24.1
## 2   22.9

> tail(data.Ex06.8, n = 2)

##    Length
## 17   31.2
## 18   30.9
```

Carry out the test with $\alpha = 0.05$, which is equivalent to the default `conf.level = 0.95`.

```
> sign.test(data.Ex06.8$Length, md = 30, alternative = "two.sided")

##
##  One-sample Sign-Test
##
## data:  x =  data.Ex06.8$Length
## s = 4, p-value = 0.03088
## alternative hypothesis: true median of x is not equal to 30
## 95 percent confidence interval:
##  23.2 28.1
## sample estimates:
## median of x
##        24.8
```

The P value is less than $\alpha = 0.05$. Likewise, the 95 percent confidence interval based on the sign test is $[23.2, 28.1]$ does not contain the hypothesized median of 30 mm. Based on the evidence, it is reasonable to conclude that these are *not* clicking froglets.

EXAMPLE 6.9. In June of 1996 in the Finger Lakes region of New York, a series of warm days in the middle of the month caused people to comment that it was “unusually” warm. The median normal maximum temperature for Ithaca in June is 75°F. Do the data (the daily maximum temperatures for all 30 days in June) support the notion that June was unusually warm? (Based on data reported in The Ithaca Climate Page, `http://snow.cit.cornell.edu/climate/ithaca/moncrt_06-96.html`, July 2000.)

72	78	79	75	74	70	77	83	85	85
84	85	80	78	82	83	83	77	68	69
76	76	80	72	73	70	70	78	78	78

SOLUTION. In this example there is reason to believe that temperatures were higher than normal. Use the `sign.test( )` with H_0: $M \leq 75°$ versus H_a: $M > 75°$, and with $\alpha = 0.05$. Set `alternative = "greater"` and `md = 75`.

```
> data.Ex06.9 <- read.table("http://waveland.com/Glover-Mitchell/Example06-9.txt",
+ header = TRUE)
> head(data.Ex06.9, n = 2)

##   Temperature
## 1          72
## 2          78

> tail(data.Ex06.9, n = 3)

##    Temperature
## 28          78
## 29          78
## 30          78

> sign.test(data.Ex06.9$Temperature, md = 75, alternative = "greater")

##
##  One-sample Sign-Test
##
## data:  x =  data.Ex06.9$Temperature
## s = 20, p-value = 0.03071
## alternative hypothesis: true median of x is greater than 75
## 95 percent confidence interval:
##   76 Inf
## sample estimates:
## median of x
##          78
```

The P value is 0.0307 and is smaller than $\alpha = 0.05$. Also, the 95 percent confidence interval based on the sign test is $[76, \infty)$ does not contain the hypothesized median of 75°. There is sufficient evidence to reject H_0. June of 1996 was unusually warm.

Confidence Intervals Based on the Sign Test

To compute confidence intervals based on the sign test for the median of a population one simply uses `sign.test( )` with the appropriate confidence level.

EXAMPLE 6.10. Find a 95% confidence interval for the froglets in Example 6.8.

SOLUTION. Look back at Example 6.8. To determine a 95 percent confidence interval use the `sign.test( )` function with its default options `conf.level = 0.95` and `alternative = "two.sided"` (so neither has to be entered). Remember that the expected median was 30 mm.

```
> sign.test(data.Ex06.8$Length, md = 30)

##
##  One-sample Sign-Test
##
## data:  x =  data.Ex06.8$Length
## s = 4, p-value = 0.03088
## alternative hypothesis: true median of x is not equal to 30
## 95 percent confidence interval:
```

```
##  23.2 28.1
## sample estimates:
## median of x
##        24.8
```

The 95 percent confidence interval based on the sign test is $[23.2, 28.1]$. This interval does not contain 30 mm, which was the median length for the clicking froglet, *Crinia signifera*. This is why in Example 6.8 we rejected the hypothesis that these were clicking froglets. But beeping froglets, *C. parinsignifera* have a median length of 25 mm, which does lie in the interval. How would you interpret this?

The One-Sample WilcoxonSigned-Rank Test

Like the sign test, the purpose of the one-sample Wilcoxon signed-rank test is to test the null hypothesis that a particular population has a hypothesized median M_0. This is carried out in R using the `wilcox.test( )` function. We present a simple form of the function in this chapter and will include additional optional arguments in the next chapter.

```
wilcox.test(x, mu = 0, alternative = "two.sided", exact = NULL,
               correct = TRUE, conf.int = FALSE, conf.level = 0.95)
```

Details: `x` is the sample data. The optional argument `mu` gives the value of the median; the default is `mu = 0`. (Recall: Under the assumption of the Wilcoxon signed-rank test the data are symmetric so the median and the mean should be the same, so μ is used.) The optional argument `alternative` gives the alternative hypothesis. The default is `"two-sided"` with the other possible choices being `"less"` or `"greater"`. By default the argument `exact` specifies that an exact P value is computed, if the sample contains less than 50 values and there are no ties. Otherwise, a normal approximation is used. The argument `correct` specifies whether to apply the continuity correction in the normal approximation for the P value and defaults to `TRUE`. The argument `conf.int` specifies whether a confidence interval should be computed (`TRUE`) or not (`FALSE`), the latter being the default. Finally, the argument `conf.level` gives the confidence level to be used in the test; the default value of `0.95` is equivalent to $\alpha = 0.05$.

EXAMPLE 6.11. The bridled goby, *Arenigobius frenatus*, occupies burrows beneath rocks and logs on silty or muddy tidal flats in Moreton Bay. Ventral fins join at the base to form a single disc—characteristic of the family Gobiidae. The median length is thought to be 80 mm, and the distribution of lengths is thought to be symmetric. Suppose a small sample was collected and the following lengths (in mm) were recorded.

Researchers assumed because the sample was collected at a location adjacent to a ferry port where there were higher than normal levels of disturbance and pollution that the bridled gobies would be somewhat stunted, that is, shorter than 80 mm. Do the data collected support this contention?

63.0 82.1 81.8 77.9 80.0 72.4 69.5 75.4 80.6 77.9

SOLUTION. Read the data from `http://waveland.com/Glover-Mitchell/Example06-11.txt`.

```
> data.Ex06.11 <- read.table("http://waveland.com/Glover-Mitchell/Example06-11.txt",
+ header = TRUE)
> data.Ex06.11
```

```
##    Length
## 1    63.0
## 2    82.1
## 3    81.8
## 4    77.9
## 5    80.0
## 6    72.4
## 7    69.5
## 8    75.4
## 9    80.6
## 10   77.9
```

Use `alternative = "less"` with `conf.level = 0.95`. The expected median is 80 mm, so set `mu = 80`. One of the gobies measured exactly 80 mm. In a calculation by hand, this observation would be eliminated. Using R the observation is automatically eliminated, but a warning message is generated unless you set `exact = FALSE`.

```
> wilcox.test(data.Ex06.11$Length, mu = 80, alternative = "less", exact = FALSE)

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  data.Ex06.11$Length
## V = 7, p-value = 0.03727
## alternative hypothesis: true location is less than 80
```

The P value is 0.037, which is less than $\alpha = 0.05$. Reject H_0. There is evidence to support that the gobies at this location are stunted.

EXAMPLE 6.13. Adult heights are known to be symmetrically distributed. We measured the heights of 10 male faculty in the science division at Hobart and William Smith Colleges. The median male adult height in the U.S. is supposedly 178 cm. Do the data collected support this assertion? Set up the appropriate two-tailed hypothesis and determine whether H_0 can be rejected at the $\alpha = 0.05$ level.

171 175 177 178 180 182 190 192 195 202

SOLUTION. The null hypothesis is H_0: $M = 178$ versus the two-sided alternative H_a: $M \neq 178$. Read the data from `http://waveland.com/Glover-Mitchell/Example06-13.txt`.

```
> data.Ex06.13 <- read.table("http://waveland.com/Glover-Mitchell/Example06-13.txt",
+ header = TRUE)
> data.Ex06.13

##    Height
## 1     171
## 2     175
## 3     177
## 4     178
## 5     180
## 6     182
## 7     190
## 8     192
## 9     195
## 10    202
```

Use the defaults `alternative = "two.sided"` and `conf.level = 0.95`. The expected median is 178 mm, so set `mu = 178`. One of the measurements is actually 178 cm, so `exact = FALSE`. Let's also compute a confidence interval, so set `conf.int = TRUE`.

```
> wilcox.test(data.Ex06.13$Height, mu = 178, conf.int = TRUE, exact = FALSE)

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  data.Ex06.13$Height
## V = 36, p-value = 0.1235
## alternative hypothesis: true location is not equal to 178
## 95 percent confidence interval:
##  176.0 193.5
## sample estimates:
## (pseudo)median
##            185
```

The P value is 0.1235, which is greater than $\alpha = 0.05$. The 95 percent confidence interval based on the Wilcoxon signed-rank test is $[176, 193.5]$. This interval contains 178 cm, which is the median height for males. Using either the P value or the confidence interval, we cannot reject H_0.

Bonus Example: Mozart and the Golden Ratio

EXAMPLE 6.14. Did Mozart use the golden ratio in his compositions? Let a and b be two quantities, with $a < b$. Recall that a and b are in the **golden ratio** if the ratio $\frac{b}{a}$ is the same as the ratio of their sum to the larger of the two quantities, $\frac{a+b}{b}$. This ratio is denoted by φ, where $\varphi = \frac{1+\sqrt{5}}{2} \approx 1.61803$. In an article by J. Putz (The golden section and the piano sonatas of Mozart. *Mathematics Magazine*, **68**(4): 275–282), the author notes that sonatas are typically divided into two sections: the Exposition (part a) in which the theme is introduced and the Development and Recapitulation (part b, longer) in which the theme is developed and revisited. The data below are the lengths of these sections (in measures) for the 29 piano sonatas of Mozart, listed by their Köchel catalog number.

Köchel	**a**	**b**
279, I	38	62
279, II	28	46
279, III	56	102
280, I	56	88
280, II	24	36
28Q, III	77	113
281, 1	40	69
281, II	46	60
282, I	15	18
282, III	39	63
283, I	53	67
283, II	14	23
283, III	102	171
284, I	51	76
309, I	58	97
311, 1	39	73
310, I	49	84
330, I	58	92
330, III	68	103
332, I	93	136
332, III	90	155
333, I	63	102
333, II	31	50
457, I	74	93
533, I	102	137
533, II	46	76
545, I	28	45
547a, I	78	118
570, I	79	130

Does the ratio $\frac{b}{a}$ in Mozart's sonatas differ significantly from the golden ratio φ? Use a t test with $\alpha = 0.05$.

SOLUTION. Let `ratio` denote the ratio of $\frac{b}{a}$. The question can be framed as the following pair of hypotheses:

$$H_0\colon \mu_{\text{ratio}} = \varphi$$
$$H_a\colon \mu_{\text{ratio}} \neq \varphi.$$

Read and list the data in `http://waveland.com/Glover-Mitchell/Example06-14.txt`.

```
> data.Ex06.14 <- read.table("http://waveland.com/Glover-Mitchell/Example06-14.txt",
+ header = TRUE)
> head(data.Ex06.14, n = 3)

##    Kochel  a   b
## 1   279,I 38  62
## 2  279,II 28  46
## 3 279,III 56 102
```

To carry out a t test on these data, use `t.test( )` with `mu = 1.61803`, the default two-sided hypothesis, and the default confidence level 0.95 (or $\alpha = 0.05$). First we need to form the ratios.

```
> data.Ex06.14["ratio"] <- data.Ex06.14$b/data.Ex06.14$a
> head(data.Ex06.14, n = 2)

##   Kochel  a  b   ratio
## 1  279,I 38 62 1.63158
## 2 279,II 28 46 1.64286

> tail(data.Ex06.14, n = 2)

##    Kochel  a   b   ratio
## 28 547a,I 78 118 1.51282
## 29  570,I 79 130 1.64557
```

The few ratios listed do look close to φ. Now carry out the test.

```
> t.test(data.Ex06.14$ratio, mu = 1.61803)

##
##  One Sample t-test
##
## data:  data.Ex06.14$ratio
## t = -1.7671, df = 28, p-value = 0.08811
## alternative hypothesis: true mean is not equal to 1.61803
## 95 percent confidence interval:
##  1.50068 1.62668
## sample estimates:
## mean of x
##   1.56368
```

The P value is 0.088, which is greater than $\alpha = 0.05$, so there is insufficient evidence to reject the null hypothesis that the ratio $\frac{b}{a}$ equals φ. Equivalently, the 95 percent confidence interval [1.500682, 1.622662] contains the expected μ value of $\varphi \approx 1.61803$. The null hypothesis cannot be rejected: Mozart's sonata ratios do not appear to be different from φ.

What mistake was made in carrying out this analysis?

Answer: The collection of Mozart piano sonatas is a *population*, not a sample. The population ratio is 1.5637, not φ.

7. *Tests of Hypothesis Involving Two Samples*

Comparing Two Variances

In R the function `var.test( )` is used for an F test comparison of the variances of two samples. The syntax for the testing variances is

```
var.test(x, y, ratio = 1, alternative = "two.sided", conf.level = 0.95)
```

> Details: `x` and `y` are the vectors containing the two samples. The optional argument `ratio` is the null hypothesis; the default value is `ratio = 1`, if not specified. The optional argument `alternative` gives the alternative hypothesis for the test. The default for `alternative` is `"two-sided"` with the other possible choices being `"less"` or `"greater"`. The optional argument `conf.level` gives the confidence level to be used in the test; the default value of `0.95` is equivalent to $\alpha = 0.05$.

EXAMPLE 7.1. Among the few reptilian lineages to emerge from the Mesozoic extinction are today's reptiles. One of these, the tuatara, *Sphenodon punctatum*, of New Zealand, is the sole survivor of a group that otherwise disappeared 100 million years ago. The mass (in g) of random samples of adult male tuatara from two islets in the Cook Strait are given below. Is the variability in mass of adult males different on the two islets?

Location A		**Location B**
510	790	650
773	440	600
836	435	600
505	815	575
765	460	452
780	690	320
235		660

SOLUTION. The question can be framed as the following pair of hypotheses:

$$H_0\colon \sigma_A^2 = \sigma_B^2$$
$$H_a\colon \sigma_A^2 \neq \sigma_B^2.$$

Read and list the data in `http://waveland.com/Glover-Mitchell/Example07-1.txt`.

```
> data.Ex07.1 <- read.table("http://waveland.com/Glover-Mitchell/Example07-1.txt",
+ header = TRUE)
> data.Ex07.1

##      A   B
## 1  510 650
## 2  773 600
## 3  836 600
## 4  505 575
## 5  765 452
## 6  780 320
## 7  235 660
## 8  790  NA
## 9  440  NA
## 10 435  NA
## 11 815  NA
## 12 460  NA
## 13 690  NA
```

The term `NA`, "not available", indicates that there is no data at that position in the table. (Blank entries are not allow in data frames; each row must have the same number of entries.) To carry out an F test on these data, use `var.test( )` with the defaults `ratio = 1`, `alternative = two.sided`, and `conf.level = 0.95` or $\alpha = 0.05$.

```
> var.test(data.Ex07.1$A, data.Ex07.1$B)

##
##  F test to compare two variances
##
## data:  data.Ex07.1$A and data.Ex07.1$B
## F = 2.5173, num df = 12, denom df = 6, p-value = 0.2661
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.469106 9.385360
## sample estimates:
## ratio of variances
##            2.51734
```

The `var.test( )` function provides two ways to answer whether the variances of the populations are significantly different. The P value is 0.226, which is greater than $\alpha = 0.05$, so there is insufficient evidence to reject the null hypothesis that the variances are equal. Equivalently, `var.test( )` provides a confidence interval for the F value. As usual, if the hypothesized value (here we expect the F ratio to be 1) falls within the confidence interval, then the null hypothesis is retained. Otherwise, the null hypothesis is rejected. In this example, the 95 percent confidence interval $[0.469, 9.385]$ contains the expected F value of 1. So the null hypothesis is not rejected. Continue to assume that the variances are equal.

EXAMPLE 7.2. In a fruit-processing plant the machine that is normally used to package frozen strawberries produces packages with an average of 250 g/box. A new machine is being considered that will process much faster, but may produce more variable results. In order to investigate this concern, the quality control supervisor measured the contents of 50 boxes produced by each machine and found $s_O^2 = 25\text{ g}^2$ and $s_N^2 = 64\text{ g}^2$. From her results, are her suspicions supported?

SOLUTION. The hypotheses are

$$\begin{aligned} H_0&: \sigma_N^2 \leq \sigma_O^2 \qquad E(F) \leq 1 \\ H_a&: \sigma_N^2 > \sigma_O^2 \qquad E(F) > 1. \end{aligned}$$

Carry out an F test to determine whether the variances are equal. Since the original data are not given, the `var.test( )` function cannot be used. This problem was meant to be done by hand. However, we have created a function called `f.test2( )` that may be used here.

`f.test2(sx, nx, sy, ny, alternative = "two.sided", conf.level = 0.95)`

Details: `sx` and `sy` are the standard deviations of the two samples and `nx` and `ny` are the corresponding sample sizes. The optional argument `alternative` gives the alternative hypothesis for the test. The default for `alternative` is `"two-sided"` with the other possible choices being `"less"` or `"greater"`. The optional argument `conf.level` gives the confidence level to be used in the test; the default value of `0.95` is equivalent to $\alpha = 0.05$.

Notice that the test uses the standard deviations of the samples as inputs, not the variances.

Download the function using the `source( )` command. The function only needs to be downloaded once per session.

In this example, the `alternative = "greater"` because $H_a: \sigma_N^2 > \sigma_O^2$ or $E(F) > 1$. Additionally, `conf.level = 0.95`, which is the default.

```
> source("http://waveland.com/Glover-Mitchell/f.test2.txt")

## Downloaded: f.test2( ).

> f.test2(sx = sqrt(64), nx = 50, sy = sqrt(25), ny = 50, alternative = "greater")

##
##  F test to compare two variances
##
## data:  sx = 8, nx = 50; and sy = 5, ny = 50
## F = 2.56, num df = 49, denom df = 49, p-value = 0.0006496
## alternative hypothesis: true ratio of variances is greater than 1
## 95 percent confidence interval:
##  1.59274     Inf
## sample estimates:
## ratio of variances
##               2.56
```

`f.test2( )` provides the same two ways as `var.test( )` to answer whether the variances are significantly different. The P value is approximately 0.00065, which is much smaller than $\alpha = 0.05$, so there is strong evidence to reject the null hypothesis that the variance of the new machine is less than or equal to the variance of the old machine. Equivalently, `f.test2( )` provides the confidence interval for the F value for the specified alternative hypothesis. In this example, the 95% confidence interval $[1.593, \infty)$ does not contain the expected F value of 1. So the null hypothesis is rejected.

Testing the Difference Between Two Means of Independent Samples

In Chapter 6, the function `t.test( )` was used to compare a sample mean, $\overline{X}$, to a theoretical mean, μ. This same function with its optional arguments can be used to test whether two sample means are different. The function can handle both paired and unpaired data, as well as equal or unequal variances. The general syntax is

```
t.test(x, y, mu = 0, alternative = "two.sided", paired = FALSE,
       var.equal = FALSE, conf.level = 0.95)
```

Details: `x` and `y` are vectors containing the two samples. The vector `y` is optional and is not used for a one-sample test as in Chapter 6. The optional argument `mu` is the hypothesized value of μ in a one-sample test or the hypothesized difference between μ_x and μ_y when comparing two samples; the default value is 0. The optional argument `alternative` is the alternative hypothesis. The default is `"two-sided"` with the other possible choices being `"less"` or `"greater"`. The argument `paired` indicates whether the data are paired or unpaired and defaults to `FALSE` or unpaired data, if not specified. The argument `var.equal` specifies whether the variances associated with `x` and `y` are equal or unequal, defaulting to `FALSE` or unequal, if not specified. Finally, the argument `conf.level` gives the confidence level to be used in the t test; the default value of `0.95` is equivalent to $\alpha = 0.05$.

EXAMPLE 7.3. Returning to the New Zealand tuataras in Example 7.1, we now ask the question: Does the average mass of adult males differ between Location A and Location B?

SOLUTION. Make use of `data.Ex07.1` read in earlier. The samples are independent (not paired), the first observation at Location A has no special relationship to any observation at Location B. The hypothesis of equal variances was tested in Example 7.1 and H_0 was accepted. The new hypotheses about the means are

$$H_0\colon \mu_A = \mu_B$$
$$H_a\colon \mu_A \neq \mu_B,$$

which leads to a two-sided test with the default $\mu_1 - \mu_2 = 0$. Use the default `conf.level = 0.95` or $\alpha = 0.05$. Example 7.1 showed that the variances of the two populations should be thought of as equal. So set `var.equal = TRUE`. The t test on these data, using the options described, is given by

```
> t.test(data.Ex07.1$A, data.Ex07.1$B, var.equal = TRUE)

##
##  Two Sample t-test
##
## data:  data.Ex07.1$A and data.Ex07.1$B
## t = 0.8217, df = 18, p-value = 0.422
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -104.297  238.297
## sample estimates:
## mean of x mean of y
##       618       551
```

The P value is 0.422, which is greater than $\alpha = 0.05$, so there is insufficient evidence to reject the null hypothesis that the means are equal. Equivalently, the 95 percent confidence interval is $[-104.297, 238.297]$. The hypothesized value is $\mu_1 - \mu_2 = 0$, which lies within this interval. So the null hypothesis is not rejected. Continue to assume that means of the masses of the two populations are equal.

EXAMPLE 7.4. An ornithologist studying various populations of the Australian magpie, *Gymnorhina tibicen*, mist-netted 25 adults from an isolated rural area and 36 from an urban picnic area. She measured the total body lengths in centimeters and reported the following summary statistics:

	Rural	**Urban**
$\overline{X}$	38 cm	35 cm
s	4 cm	3 cm
n	25	36

Because picnickers often feed the magpies, it is thought that the urban population might be somewhat stunted due to the consumption of processed rather than natural foods. Completely analyze the data given to see if it supports this hypothesis.

SOLUTION. The samples are independent, but do they come from populations with the same variance? Perform an F test with $\alpha = 0.05$ to answer this question. The hypotheses are

$$H_0\colon\ \sigma_r^2 = \sigma_u^2$$
$$H_a\colon\ \sigma_r^2 \neq \sigma_u^2.$$

Since the original data are not given, the function `var.test( )` cannot be used. Instead use `f.test2( )` as in Example 7.2 with `sx = 4`, `nx = 25`, `sy = 3`, and `ny = 36`.

```
> f.test2(sx = 4, nx = 25, sy = 3, ny = 36)
```

```
##
##  F test to compare two variances
##
## data:  sx = 4, nx = 25; and sy = 3, ny = 36
## F = 1.7778, num df = 24, denom df = 35, p-value = 0.1182
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.862306 3.863724
## sample estimates:
## ratio of variances
##            1.77778
```

The P value is 0.118, which is greater than $\alpha = 0.05$. Alternatively, observe that the expected F value of 1 is in the confidence interval. The null hypothesis that the two variances are equal cannot be rejected.

Now the question about mean size can be addressed. Are the magpies smaller in urban areas than in rural areas?

$$H_0\colon\ \mu_r \le \mu_u$$
$$H_a\colon\ \mu_r > \mu_u.$$

This requires a one-tailed t test because we anticipate a deviation from equality in a particular direction.

As noted in the previous chapter, `R` does not provide a function to carry out a t test when the original data are not given. Just as in the previous chapter, you may use `t.test2( )` with additional arguments to carry out the analysis.

```
t.test2(mx, sx, nx, my, sy, ny, mu = 0, alternative = "two.sided",
        paired = FALSE, var.equal = FALSE, conf.level = 0.95)
```

Details: `mx`, `sx`, and `nx` are the mean, standard deviation, and sample size for the `x` sample and `my`, `sy`, and `ny` are the corresponding values for the `y` sample. All of the arguments except `mx`, `sx`, and `nx` are optional. If `my`, `sy`, and `ny` are not present, then a one-sample test is carried out as in Chapter 6. The optional argument `mu` is the expected value of the difference in the means for a two-sample test (or the expected value of the mean in a one-sample test). The default value is `mu = 0`. The optional argument `alternative` is the alternative hypothesis. The default is `"two-sided"` with the other possible choices being `"less"` or `"greater"`. The argument `paired` indicates whether the data are paired or unpaired and defaults to `FALSE` or unpaired data, if not specified. The argument `var.equal` specifies whether the variances associated with `x` and `y` are equal or unequal, defaulting to `FALSE` or unequal, if not specified. Finally, the argument `conf.level` gives the confidence level to be used in the t test; the default value of `0.95` is equivalent to $\alpha = 0.05$.

Download the function, if you have not done so earlier in this session, using the `source( )` command. The function only needs to be downloaded once per session.

```
>  source("http://waveland.com/Glover-Mitchell/t.test2.txt")

## Downloaded: t.test2( ).
```

In this example, the mean, standard deviation, and sample size for the first sample are `mx = 38`, `sx = 4.0`, and `nx = 25`, respectively. Similarly, for the second sample `my = 35`, `sy = 3.0`, and `ny = 36`. The alternative hypothesis is that the mean of the rural `x` sample is greater than that of the urban `y` sample. The F test above indicated that `var.equal = TRUE`.

```
> t.test2(mx = 38, sx = 4.0, nx= 25, my = 35, sy = 3.0, ny = 36,
+ alternative = "greater", var.equal = TRUE)

##
```

```
##  Two Sample t-test
##
## data:  mx = 38, sx = 4, nx = 25; and my = 35, sy = 3, ny = 36
## t = 3.3478, df = 59, p-value = 0.0007113
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  1.50253     Inf
## sample estimates:
## mean of x mean of y
##        38        35
```

The P value of 0.0007, which is much smaller than $\alpha = 0.05$, indicates that the null hypothesis should be rejected. Equivalently, the 95 percent confidence interval for $\mu_1 - \mu_2$ is $[1.503, \infty)$. This interval does not include 0, so the null hypothesis should be rejected. There is evidence that the mean mass of the rural magpies is greater than that of the urban magpies.

EXAMPLE 7.5. As part of a larger study of the effects of amphetamines, a pharmacologist wished to investigate the claim that amphetamines reduce overall water consumption. She injected 15 standard lab rats with an appropriate dose of amphetamine and 10 with a saline solution as controls. Over the next 24 hours she measured the amount of water consumed by each rat and expressed the results in ml/kg body weight:

	Amphetamine	**Saline**
n	15	10
$\overline{X}$	115	135
s	40	15

Does the amphetamine significantly suppress water consumption? Use an alpha level of 0.05 for any tests of hypotheses.

SOLUTION. First carry out a preliminary F test with $\alpha = 0.05$ for the variances:

$$H_0\colon \sigma_A^2 = \sigma_S^2$$
$$H_a\colon \sigma_A^2 \neq \sigma_S^2.$$

```
> f.test2(sx = 40, nx = 15, sy = 15, ny = 10)

##
##  F test to compare two variances
##
## data:  sx = 40, nx = 15; and sy = 15, ny = 10
## F = 7.1111, num df = 14, denom df = 9, p-value = 0.005652
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##   1.87235 22.82169
## sample estimates:
## ratio of variances
##            7.11111
```

The tiny P value of 0.0057 indicates that the null hypothesis should be rejected. There is strong evidence that the two variances are not equal.

The hypotheses for the means are

$$H_0\colon \mu_A \geq \mu_S$$
$$H_a\colon \mu_A < \mu_S.$$

Since only the summary data are provided, use the function `t.test2( )` to carry out the analysis. The results of the F test indicate that `var.equal = FALSE`. From reviewing the hypotheses for the means, `alternative = "less"`.

```
> t.test2(mx = 115, sx = 40, nx = 15, my = 135, sy = 15, ny = 10,
+ alternative = "less", var.equal = FALSE)

##
##  Welch Two Sample t-test
##
## data:  mx = 115, sx = 40, nx = 15; and my = 135, sy = 15, ny = 10
## t = -1.7598, df = 19.2, p-value = 0.04719
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf -0.35863
## sample estimates:
## mean of x mean of y
##       115       135
```

Since the P value is 0.047 is smaller than $\alpha = 0.05$, reject H_0. The data support the claim that amphetamines significantly suppress water consumption.

Confidence Intervals for $\mu_1 - \mu_2$

A confidence interval for $\mu_1 - \mu_2$ at the $(1 - \alpha)100\%$ level of confidence can be calculated by using the `t.test( )` or `t.test2( )` functions. This is illustrated below.

EXAMPLE 7.6. The magpie body lengths in Example 7.4 for the rural population were deemed significantly longer than the body lengths for the urban population. Calculate a 95% confidence interval for the difference in mean size.

SOLUTION. Only summary data were provided, so use `t.test2( )`. In Example 7.4 the variances of both populations were shown to be equal. The argument `alternative = "two.sided"` is optional since it is the default.

```
> t.test2(mx = 38, sx = 4.0, nx = 25, my = 35, sy = 3.0, ny = 36,
+ alternative = "two.sided", var.equal = TRUE)

##
##  Two Sample t-test
##
## data:  mx = 38, sx = 4, nx = 25; and my = 35, sy = 3, ny = 36
## t = 3.3478, df = 59, p-value = 0.001423
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.20691 4.79309
## sample estimates:
## mean of x mean of y
##        38        35
```

Among the results is the 95% confidence interval for the difference in the means: $[1.207, 4.793]$. This differs only slightly from the text answer due to rounding in the hand calculations. To get just the confidence interval, add `$conf.int` to the end of the command. This will print only the confidence interval information, albeit in an ugly form. For example, to determine a 99% confidence interval use

```
> t.test2(mx = 38, sx = 4.0, nx = 25, my = 35, sy = 3.0, ny = 36,
+ var.equal = TRUE, conf.level = 0.99)$conf.int

## [1] 0.614799 5.385201
## attr(,"conf.level")
## [1] 0.99
```

The increase in confidence necessitates a wider interval: $[0.615, 5.835]$.

EXAMPLE 7.7. Calculate the 95% confidence interval from Example 7.5 for the reduction in water consumption in lab rats when they are treated with amphetamines.

SOLUTION. In Example 7.5 the variances of both populations couldn't be assumed to be equal. Set `var.equal = FALSE` or leave the argument out since it is the default.

```
> t.test2(mx = 115, sx = 40, nx = 15, my = 135, sy = 15, ny = 10,
+ var.equal = FALSE)$conf.int

## [1] -43.77075   3.77075
## attr(,"conf.level")
## [1] 0.95
```

We are 95% confident that the reduction in the amount of water consumed lies in the interval $[-43.77, 3.77]$ ml/kg. Notice that 0 (implying no difference) is included in this confidence interval. What does this indicate?

The Difference Between Two Means with Paired Data

Careful design of experiments, including the use of pairing, can reduce or remove a great deal of the variability in data that often obscures meaningful differences.

EXAMPLE 7.8. Watching an infomercial on television you hear the claim that without changing your eating habits, a particular herbal extract when taken daily will allow you to lose 5 lb in 5 days. You decide to test this claim by enlisting 12 of your classmates into an experiment. You weigh each subject, ask them to use the herbal extract for 5 days and then weigh them again. From the results recorded below, test the infomercial's claim of 5 lb lost in 5 days.

Subject	Weight before	Weight after
1	128	120
2	131	123
3	165	163
4	140	141
5	178	170
6	121	118
7	190	188
8	135	136
9	118	121
10	146	140
11	212	207
12	135	126

SOLUTION. Read and list the data in `http://waveland.com/Glover-Mitchell/Example07-8.txt`.

```
> data.Ex07.8 <- read.table("http://waveland.com/Glover-Mitchell/Example07-8.txt",
+ header = TRUE)
> tail(data.Ex07.8, n = 3)
```

```
##    Subject Before After
## 10      10    146   140
## 11      11    212   207
## 12      12    135   126
```

Because the data are paired we are not directly interested in the values presented above, but are instead interested in the *differences* in the pairs of numbers. These differences are automatically calculated by `t.test( )` by setting `paired = TRUE`. The infomercial claim of a 5-lb loss can be written as

$$H_0\colon \mu_B - \mu_A \geq 5 \text{ lb}$$
$$H_a\colon \mu_B - \mu_A < 5 \text{ lb}.$$

Now carry out a paired t test on the two samples with `mu = 5`.

```
> t.test(data.Ex07.8$Before, data.Ex07.8$After, mu = 5, alternative = "less",
+ paired = TRUE)

##
##  Paired t-test
##
## data:  data.Ex07.8$Before and data.Ex07.8$After
## t = -0.9837, df = 11, p-value = 0.1732
## alternative hypothesis: true difference in means is less than 5
## 95 percent confidence interval:
##     -Inf 5.96323
## sample estimates:
## mean of the differences
##                 3.83333
```

The P value is 0.1732, which is greater than $\alpha = 0.05$, so H_0 cannot be rejected. Another way to see this is that the confidence interval for the difference of the means is $(-\infty, 5.963]$ and contains 5, the hypothesized value of the difference.

EXAMPLE 7.9. An experiment was conducted to compare the performance of two varieties of wheat, A and B. Seven farms were randomly chosen for the experiment and the yields in metric tons per hectare for each variety on each farm were recorded.

Farm	Yield of variety A	Yield of variety B
1	4.6	4.1
2	4.8	4.0
3	3.2	3.5
4	4.7	4.1
5	4.3	4.5
6	3.7	3.3
7	4.1	3.8

Carry out a hypothesis test to decide whether the mean yields are the same for the two varieties.

SOLUTION. The experiment was designed to test both varieties on each farm because different farms may have significantly different yields due to differences in soil characteristics, microclimate, or cultivation practices. "Pairing" the data points accounts for most of the "between farm" variability and should make any differences in yield due solely to wheat variety. Read and list the data.

```
> data.Ex07.9 <- read.table("http://waveland.com/Glover-Mitchell/Example07-9.txt",
+ header = TRUE)
> data.Ex07.9
```

```
##   Farm   A   B
## 1    1 4.6 4.1
## 2    2 4.8 4.0
## 3    3 3.2 3.5
## 4    4 4.7 4.1
## 5    5 4.3 4.5
## 6    6 3.7 3.3
## 7    7 4.1 3.8
```

The hypotheses are

$$H_0\colon \mu_A = \mu_B$$
$$H_a\colon \mu_A \neq \mu_B.$$

Use `t.test( )` with `paired = TRUE` and the defaults `mu = 0` and `alternative = "two.sided"`.

```
> t.test(data.Ex07.9$A, data.Ex07.9$B, paired = TRUE)

##
##  Paired t-test
##
## data:  data.Ex07.9$A and data.Ex07.9$B
## t = 1.9442, df = 6, p-value = 0.09986
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.0775667  0.6775667
## sample estimates:
## mean of the differences
##                     0.3
```

The P value is 0.0999, which is greater than $\alpha = 0.05$, so H_0 cannot be rejected. Also the confidence interval $[-0.776, 0.678]$ contains 0, the hypothesized value of μ_d. From the data given we cannot say that the yields of varieties A and B are significantly different.

*The Wilcoxon Rank-Sum (Mann-Whitney **U**) Test*

If there are reasons to believe that the distributions involved are normal, then tests on the differences in the means of the two populations are carried out by using one form or another of the t test, depending on whether variances are equal or not. But often we don't know anything about the populations or perhaps we know that the distributions are not normal. In this case, the most common distribution-free hypothesis test used is the **Wilcoxon rank-sum test**, also known as the **Mann-Whitney *U* test**. This is carried out in R using the `wilcox.test( )` function.

```
wilcox.test(x, y = NULL, alternative = "two.sided", mu = 0,
              paired = FALSE, exact = NULL, correct = TRUE,
             conf.int = FALSE, conf.level = 0.95)
```

Details: `x` and `y` are vectors containing the two samples. The vector `y` is optional and is not used when carrying out a one-sample Wilcoxon signed-rank test. The optional argument `mu` is the value of the hypothesized median in a one-sample test, or the hypothesized difference between M_x and M_y when comparing two medians; the default value is `0`. The optional argument `alternative` is the alternative hypothesis. The default is `"two-sided"` with the other possible choices being "less" or "greater". The argument paired indicates whether the data are paired or unpaired and defaults to FALSE or unpaired data, if not specified. By default the argument exact specifies whether an exact P value is computed if the samples contain less than 50 values and there are no ties. Otherwise, a normal approximation is used. The argument correct specifies whether to apply the continuity correction in the normal approximation for the P value and defaults to TRUE. The argument conf.int specifies whether a confidence interval should be computed (TRUE) or not (FALSE), the latter being the default. Finally, the argument conf.level gives the confidence level to be used in the test; the default value of 0.95 is equivalent to $\alpha = 0.05$.

EXAMPLE 7.10. Two color morphs (green and red) of the same species of sea star were found at Polka Point, North Stradbroke Island. One might suppose that the red color morph is more liable to predation and, hence, those found might be smaller than the green color morph. The following data are the radial lengths of two random samples of these sea stars (in mm). Are red color morphs significantly smaller than green color morphs?

Red	108	64	80	92	40	
Green	102	116	98	132	104	124

SOLUTION. Read and list the data in http://waveland.com/Glover-Mitchell/Example07-10.txt.

```
> data.Ex07.10 <- read.table("http://waveland.com/Glover-Mitchell/Example07-10.txt",
+ header = TRUE)
> data.Ex07.10

##   Red Green
## 1 108   102
## 2  64   116
## 3  80    98
## 4  92   132
## 5  40   104
## 6  NA   124
```

To test whether red color morphs tend to be smaller than the green, the appropriate test is left-tailed with H_0: $M_R \geq M_G$ versus H_a: $M_R < M_G$. The test is done at the $\alpha = 0.05$ level. Use wilcox.test( ) with alternative = "less" and the defaults mu = 0 and conf.level = 0.95 (equivalently, $\alpha = 0.05$).

```
> wilcox.test(data.Ex07.10$Red, data.Ex07.10$Green, alternative = "less")

##
##  Wilcoxon rank sum test
##
## data:  data.Ex07.10$Red and data.Ex07.10$Green
## W = 3, p-value = 0.01515
## alternative hypothesis: true location shift is less than 0
```

The small P value of 0.015 is less than $\alpha = 0.05$, so there is little evidence to support the null hypothesis. Accept H_a, the red color morphs are, in fact, significantly smaller than the green color morphs.

Note on the test statistic W in R. The careful reader may have noticed a substantial difference in the test statistic computed in the text and by R. This isn't roundoff error. In the text, the test statistic is the sum of the red sea star ranks,

$$W_R = 1 + 2 + 3 + 4 + 8 = 18.$$

Yet R reports that $W = 3$. This latter value is more commonly referred to as the Mann-Whitney U statistic. That is, $U = 3$. When the rank sum of the smaller sample size (say n) is calculated, the minimum possible sum is

$$1 + 2 + 3 + \cdots + n = \frac{n(n+1)}{2}.$$

In this example the minimum value would be

$$1 + 2 + 3 + 4 + 5 = 15.$$

The statistic U is the difference between this minimum value and the actual rank sum. That is, the statistics W and U are related by

$$W = U + \frac{n(n+1)}{2}.$$

In the case at hand, the sample size of the red sea stars is $n = 5$, so the example,

$$U + \frac{n(n+1)}{2} = U + \frac{5(6)}{2} = 3 + 15 = 18.$$

The rank sum W is easier to compute by hand so it is used in the text.

The Sign Test and Paired Data

Earlier we applied the sign test to a sample drawn from a population with a known or hypothesized median (see Section 6.4). A more common situation is a *paired comparison* in which subjects are matched in pairs and the outcomes are compared within each matched pair. The `sign.test( )` function from Chapter 6 can be used to answer questions about median difference with paired observations, as in the example below.

EXAMPLE 7.11. Triglycerides are blood constituents that are thought to play a role in coronary artery disease. An experiment was conducted to see if regular exercise could reduce triglyceride levels. Researchers measured the concentration of triglycerides in the blood serum of 7 male volunteers before and after participation in a 10-week exercise program. The results are shown below. Note the variation in levels from one person to the next, both before and after. But an overall pattern of reduction after exercise is clear. An obvious question to ask is: "How much did exercise reduce triglyceride levels?" This is asking for either a point estimate or a confidence interval for the median difference.

Before	0.87	1.13	3.14	2.14	2.98	1.18	1.60
After	0.57	1.03	1.47	1.43	1.20	1.09	1.51

SOLUTION. First download the function, if you have not already done so, using the `source( )` command. Then read the data.

```
> source("http://waveland.com/Glover-Mitchell/sign.txt")

## Downloaded: sign.test( ).

> data.Ex07.11 <- read.table("http://waveland.com/Glover-Mitchell/Example07-11.txt",
+ header = TRUE)
> data.Ex07.11

##   Before After
## 1   0.87  0.57
## 2   1.13  1.03
## 3   3.14  1.47
## 4   2.14  1.43
## 5   2.98  1.20
## 6   1.18  1.09
## 7   1.60  1.51
```

In this case a right-tailed test, H_a: $M_{\text{Bef}} > M_{\text{Aft}}$, is called for because researchers are testing whether regular exercise lowers triglyceride levels. So set `alternative = "greater"`. The null hypothesis is H_0: $M_{\text{Bef}} \leq M_{\text{Aft}}$, the triglyceride levels before exercise are no greater than those after. Use the `sign.test( )` with the defaults `mu = 0` and `conf.level = 0.95` or $\alpha = 0.05$. When two data files are specified, `sign.test( )` automatically carries out a paired test.

```
> sign.test(data.Ex07.11$Before, data.Ex07.11$After, alternative = "greater")

##
##  Paired-samples Sign-Test
##
## data:  x = data.Ex07.11$Before and y = data.Ex07.11$After
## s = 7, p-value = 0.007813
## alternative hypothesis: true median difference of x - y is greater than 0
## 95 percent confidence interval:
##  0.09  Inf
## sample estimates:
## median of x - y
##             0.3
```

The tiny P value of 0.0078 is less than $\alpha = 0.05$, so there is little evidence to support the null hypothesis. There is strong evidence that there is a drop in the median triglyceride levels after exercise.

EXAMPLE 7.12. Zonation is a process of adaptation that can occur when various organisms compete for resources in a small area and can be observed in microhabitats such as the intertidal shoreline of a bay. Moving a distance of a meter or two perpendicular to the waterline can drastically change the amounts of moisture, temperature, and salinity. Different organisms tolerate different extremes and zonation occurs.

While at North Stradbroke Island, students carried out exercises to quantify these processes. The intertidal zone (the shore between the lowest low tide and the highest high tide) was divided into five equal-width bands parallel to the waterline. Square-meter quadrats were used to repeatedly sample the various bands for marine organisms. One subset of the data was used to study the vertical zonation of the intertidal zone by various gastropods, such as *Bembicium*, *Austrocochlea*, and *Nerita*. The data below show the distribution of *Bembicium* at 12 locations where counts were made at the mean water mark and at one zone directly above it. Is there a difference in the median numbers of *Bembicium* in the two zones?

Above mid-shore	81	20	25	62	41	28	43	50	48	31	39	24
Mid-shore	135	15	31	81	59	72	66	101	36	92	59	41

SOLUTION. In this case a two-tailed test is called for since prior to the exercise students knew little about the natural history of the organisms. The null hypothesis is H_0: There is no difference in the median numbers of *Bembicium* in the two zones ($M_{\text{Above}} = M_{\text{Mid}}$). The alternative is H_a: There is evidence for a difference in the median numbers ($M_{\text{Above}} \neq M_{\text{Mid}}$). Read and list the data in `http://waveland.com/Glover-Mitchell/Example07-12.txt` and use the `sign.test( )` with the defaults `alternative = "two.sided"` and `conf.level = 0.95`.

```
> data.Ex07.12 <- read.table("http://waveland.com/Glover-Mitchell/Example07-12.txt",
+ header = TRUE)
> head(data.Ex07.12, n = 3)

##   Above Mid
## 1    81 135
## 2    20  15
## 3    25  31

> sign.test(data.Ex07.12$Above, data.Ex07.12$Mid)
```

```
##
##  Paired-samples Sign-Test
##
## data:  x = data.Ex07.12$Above and y = data.Ex07.12$Mid
## s = 2, p-value = 0.03857
## alternative hypothesis: true median difference of x - y is not equal to 0
## 95 percent confidence interval:
##  -51  -6
## sample estimates:
## median of x - y
##           -19.5
```

The P value is 0.0386 and is less than $\alpha = 0.05$, so there is a difference in the median numbers of *Bembicium* per square meter in these two zones.

The Wilcoxon Signed-Rank Test for Paired Data

The Wilcoxon signed-rank test `wilcox.test( )` is easily extended to paired data using the optional argument `paired = TRUE`.

EXAMPLE 7.13. Body height and arm span are related quantities; taller people generally have longer arms. One rule of thumb says that adult height is nearly equal to adult span. The arm spans of the 10 science faculty in Example 6.13 were also measured. Do the data support the contention? (Height and span data are symmetric.)

Height	171	175	177	178	180	182	190	192	195	202
Span	173	178	182	182	188	185	186	198	193	202

SOLUTION. Test the null hypothesis H_0: $M_{H-S} = 0$ versus the two-sided alternative hypothesis, H_a: $M_{H-S} \neq 0$ at the $\alpha = 0.05$ level using the Wilcoxon signed-rank test because of the symmetry assumption. Read and list the data in `http://waveland.com/Glover-Mitchell/Example07-13.txt`.

```
> data.Ex07.13 <- read.table("http://waveland.com/Glover-Mitchell/Example07-13.txt",
+ header = TRUE)
> data.Ex07.13

##    Height Span
## 1     171  173
## 2     175  178
## 3     177  182
## 4     178  182
## 5     180  188
## 6     182  185
## 7     190  186
## 8     192  198
## 9     195  193
## 10    202  202
```

Though not necessary, it is often instructive to form the paired differences between the samples.

```
> differences.Ex07.13 <- data.Ex07.13$Height - data.Ex07.13$Span
> differences.Ex07.13

##  [1] -2 -3 -5 -4 -8 -3  4 -6  2  0
```

Notice that there are ties in unsigned differences (2 and -2, two -3's, and 4 and -4). Because the presence of ties makes it impossible to compute an exact P value, this requires setting `exact = FALSE` in `wilcox.test( )`. Use the defaults `mu = 0` for the hypothesized median difference and `alternative = "two.sided"`.

```
> wilcox.test(data.Ex07.13$Height, data.Ex07.13$Span, paired = TRUE, exact = FALSE)

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  data.Ex07.13$Height and data.Ex07.13$Span
## V = 7, p-value = 0.07479
## alternative hypothesis: true location shift is not equal to 0
```

The P value is 0.075, which is greater than $\alpha = 0.05$, so do not reject H_0. We cannot reject the claim that adult height and arm span are nearly equal.

Notes: Suppose that you did not compute the height-span differences first and did not see that there were tied unsigned differences. It is entirely possible that you would analyze the data without setting `exact = FALSE`. Here's what happens.

```
> wilcox.test(data.Ex07.13$Height, data.Ex07.13$Span, paired = TRUE)

## Warning in wilcox.test.default(data.Ex07.13$Height, data.Ex07.13$Span, paired
= TRUE): cannot compute exact p-value with ties
## Warning in wilcox.test.default(data.Ex07.13$Height, data.Ex07.13$Span, paired
= TRUE): cannot compute exact p-value with zeroes

##
##  Wilcoxon signed rank test with continuity correction
##
## data:  data.Ex07.13$Height and data.Ex07.13$Span
## V = 7, p-value = 0.07479
## alternative hypothesis: true location shift is not equal to 0
```

Warning messages are generated that ties were present as were 0-differences, so an exact P value cannot be computed. Instead the function acts as if you had set `exact = FALSE` and continues the analysis. So you still get the right answer, but your professor may be disappointed.

A bigger issue is that if you did not set `paired = TRUE`, the data would be analyzed as if they were independent samples. This means that the Wilcoxon rank sum test would have been performed.

```
> wilcox.test(data.Ex07.13$Height, data.Ex07.13$Span, mu = 0, exact = FALSE)

##
##  Wilcoxon rank sum test with continuity correction
##
## data:  data.Ex07.13$Height and data.Ex07.13$Span
## W = 40, p-value = 0.4717
## alternative hypothesis: true location shift is not equal to 0
```

Notice that the P value is much larger now, 0.4717 versus 0.07479. In a different experiment, this difference in P values could easily be the difference between rejecting and accepting a null hypothesis. Designing experiments with paired data may remove variability and potentially allowing researchers to make finer discriminations.

8. $\boldsymbol{k}$-Sample Tests of Hypothesis: ANOVA

ANOVA in R

R can be used to carry out analysis of variance using the `aov( )` function.

`aov(y ∼ g, data = mydataframe)`

Details: `y` is the column of response measurements (data) and `g` is the corresponding column of treatment levels (groups) that were applied. The argument `data = mydataframe` is the data frame containing the data (measurements with corresponding treatment levels) to be analyzed.

The following examples illustrate how to use this function.

EXAMPLE 8.1. A gerontologist investigating various aspects of the aging process wanted to see whether staying "lean and mean," that is, being under normal body weight, would lengthen life span. She randomly assigned newborn rats from a highly inbred line to one of three diets: (1) unlimited access to food, (2) 90% of the amount of food that a rat of that size would normally eat, or (3) 80% of the amount of food that a rat of that size would normally eat. She maintained the rats on three diets throughout their lives and recorded their lifespans (in years). Is there evidence that diet affected life span in this study?

Unlimited	90% diet	80% diet
2.5	2.7	3.1
3.1	3.1	2.9
2.3	2.9	3.8
1.9	3.7	3.9
2.4	3.5	4.0

SOLUTION. Read and list the data in `http://waveland.com/Glover-Mitchell/Example08-1.txt`.

```
> data.Ex08.1 <- read.table("http://waveland.com/Glover-Mitchell/Example08-1.txt",
+ header = TRUE)
> data.Ex08.1

##    Lifespan      Diet
## 1       2.5 Unlimited
## 2       3.1 Unlimited
## 3       2.3 Unlimited
## 4       1.9 Unlimited
## 5       2.4 Unlimited
## 6       2.7   90%Diet
## 7       3.1   90%Diet
## 8       2.9   90%Diet
## 9       3.7   90%Diet
## 10      3.5   90%Diet
## 11      3.1   80%Diet
## 12      2.9   80%Diet
## 13      3.8   80%Diet
## 14      3.9   80%Diet
## 15      4.0   80%Diet
```

Note that the format of the data differs from the text. Each row consists of a response (`Lifespan`) to a treatment level (`Diet`). We carry out the ANOVA using the `aov( )` command and put the result in the table `aov.Ex08.1`. Note that we are carrying out an analysis of variance on `Lifespan` by the `Diet` factors. The arguments of `aov( )` reflect this by using the header names from the appropriate columns in the data frame `data.Ex08.1`.

```
> aov.Ex08.1<- aov(Lifespan ~ Diet, data = data.Ex08.1)
```

To actually see the results of the ANOVA, use the `summary( )` command.

```
> summary(aov.Ex08.1)

##             Df Sum Sq Mean Sq F value Pr(>F)
## Diet         2   3.15   1.573     7.7 0.0071 **
## Residuals   12   2.45   0.204
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note the two asterisks ** at the end of the `Diet` row in the table. Using the significance codes in the output, this means that the F test is significant at the 0.01 level. Indeed, the P value is given in the table as `Pr(>F) = 0.00707`.

To see the means by treatment use the `tapply( )` function. This command is used to apply a function (here the `mean`) to groups (here, `Diet` type) within a table or data frame. In other words, `tapply( )` can be thought of as shorthand for "table apply."

```
tapply(x, g, FUN = function)
```

Details: `x` is typically a vector (column) of data and `g` is a list of corresponding factors of same length as `x`. `FUN` is the `function` to be applied to each group of cells defined by the categorical variables in `g`.

```
> tapply(data.Ex08.1$Lifespan, data.Ex08.1$Diet, mean)

##   80%Diet   90%Diet Unlimited
##      3.54      3.18      2.44
```

The box plot of the data provides an indication of whether any treatments (`Diets`) are significantly different. We might not take the time to do this by hand, but `R` makes it easy.

```
> boxplot(Lifespan ~ Diet, data = data.Ex08.1)
```

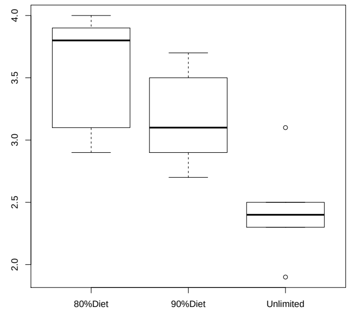


Figure 8.9: The box plot of `Lifespan` by `Diet` indicates there is a difference between the mean lifespans of 80% and unlimited diets. Other comparisons are less clear.

The F test was significant (see the ANOVA table above) and the box plot indicates that there is a difference between some mean lifespans. To carry out Bonferroni-Holm pairwise t tests to locate any differences in mean lifespans use

```
pairwise.t.test(x, g, p.adjust.method = "holm", conf.level = 0.95)
```

Details: `x` is the measurement or response data, `g` contains the corresponding treatment levels, and `p.adjust.method = "holm"` adjusts the P values of the sequence of tests according to the method selected. The abbreviation `p.adj` may be used in place of `p.adjust.method`. The optional argument `conf.level` gives the confidence level to be used in the test; the default value of `0.95` is equivalent to $\alpha = 0.05$. There are additional arguments for this function that are not needed at this stage.

In this example, `data.Ex08.1$Lifespan` is the response data and `data.Ex08.1$Diet` contains the corresponding treatment (group).

```
> pairwise.t.test(data.Ex08.1$Lifespan, data.Ex08.1$Diet, p.adj = "holm")

##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  data.Ex08.1$Lifespan and data.Ex08.1$Diet
##
##           80%Diet 90%Diet
## 90%Diet   0.232   -
## Unlimited 0.007   0.047
##
## P value adjustment method: holm
```

The output of `pairwise.t.test( )` is presented in tabular form. The entry in the column `80%Diet` and row `90%Diet` is 0.232. This is the P value for the the test of the difference in means, and so is not significant at the $\alpha = 0.05$ level. Similar remarks apply to the other entries. The table indicates that the unlimited diet is different from both the 80% and 90% diets; the corresponding P values are 0.007 and 0.047, respectively. The latter is barely significant.

The listed P values are to be compared directly to the value of α with no further adjustment. Whereas in the text we compared the computed P value p_j to $\frac{\alpha}{m+1-j}$, `pairwise.t.test( )` instead multiplies p_j by $m+1-j$ and now the comparison should be to α. That is, in R we are comparing $(m+1-j)p_j$ to α.

Another form of paired comparisons can be carried out using Tukey's Honest Significant Difference test. The corresponding R function is `TukeyHSD( )`.

`TukeyHSD(fit, ordered = TRUE, conf.level = 0.95)`

> Details: `fit` is the output of the `aov( )` function (ANOVA) carried out earlier on the relevant data frame. The argument `ordered` specifies whether to order the means in the analysis. The default is `FALSE`, but to more closely match the procedure in text, use `ordered = TRUE`. The optional argument `conf.level` gives the confidence level to be used in the test; the default value of `0.95` is equivalent to $\alpha = 0.05$.

In this example, the output from the earlier ANOVA resides in `aov.Ex08.1`. The pairwise comparisons are carried out at the $\alpha = 0.05$ level using the default `conf.level = 0.95` as follows.

```
> TukeyHSD(aov.Ex08.1, ordered = TRUE)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##     factor levels have been ordered
##
## Fit: aov(formula = Lifespan ~ Diet, data = data.Ex08.1)
##
## $Diet
##                   diff        lwr     upr    p adj
## 90%Diet-Unlimited 0.74 -0.0227167 1.50272 0.057470
## 80%Diet-Unlimited 1.10  0.3372833 1.86272 0.006070
## 80%Diet-90%Diet   0.36 -0.4027167 1.12272 0.443277
```

The results table provides a 95% confidence interval for the difference in each pair of means and the corresponding P value. The means are significantly different at the $\alpha = 0.05$ level only if 0 is not in the confidence interval. The mean of the 80% diet is different from the mean of the unlimited diet because the confidence interval $[0.337, 1.863]$ does not contain 0. The corresponding P value is 0.006. Using the Tukey test, 90% and unlimited diets just fail being significantly different in this small study because $p = 0.057$ and the confidence interval is $[-0.023, 1.502]$. This result is different than with the Bonferroni-Holm test. The Tukey test is more conservative than the Bonferroni-Holm test. There is a smaller chance of making a Type I error but a larger chance of making a Type II error (accepting a false null hypothesis). In this case, a larger sample size might have given a clearer result.

EXAMPLE 8.2. A paleontologist studying Ordovician fossils collected four samples of the trilobite *Paradoxides* (members of an extinct subphylum of the Arthropoda) from the same rock strata, but from different locations. He measured the cephalon length (in mm) of all the specimens he collected and recorded the data below. Are there significant differences in cephalon length among these sites?

	Site			
	I	**II**	**III**	**IV**
n	8	5	11	15
$\overline{X}$	7.0	9.0	8.0	5.0
s	2.1	2.1	2.2	2.2
$\sum X_i$	56.0	45.0	88.0	75.0

R does not provide a way to carry out an ANOVA from the summary statistics of an experiment rather than the actual data. This exercise is really meant to be done by hand so that you can check your understanding of how ANOVA calculations actually work. Nonetheless, we provide three functions that will allow us to carry out an ANOVA and mean separation.

aov2(SSTreat, SSError, n)

Details: SSTreat and SSError are the sums of squares required for ANOVA and n is the vector of treatment sample sizes. This function typically requires the preliminary calculations of SS_{Treat} and SS_{Error}.

pairwise.holm.t.test(meanTreat, n, g, MSE)

pairwise.tukey.test(meanTreat, n, g, MSE)

Details: For both functions meanTreat is a vector containing the means of the treatments; n is a corresponding vector containing the treatment sample sizes, g is the vector containing the corresponding treatment names (which must be in quotes), and MSE is the mean square error from the ANOVA in aov2( ).

All three of these functions may be downloaded at once using the source( ) command. See the example below.

SOLUTION. To carry out the analysis, we will use aov2( ) since we have only the summary statistics for the experiment and not the original data. To use aov2( ), one must first determine SS_{Treat} and SS_{Error}. Let's see how this can be done in R.

```
> # Preparation
> # Create a vector n containing the Treatment sample sizes
> n <- c(8, 5, 11, 15)
> # Create a vector T containing the Treatment totals
> T <- c(56.0, 45.0, 88.0, 75.0)
> # Create a vector containing the Treatment means
> meanTreat <- T/n        # Each treatment total is divided by its sample size
> meanTreat

## [1] 7 9 8 5

> # Create the vector of Treatment names
> g <- c("I", "II", "III", "IV")
> # Create a vector s containing the standard deviations of each Treatment sample
> s <- c(2.1, 2.1, 2.2, 2.2)
```

Now we are ready to carry out the ANOVA. Calculate SSTreat and SSError using the formulæ in the text:

$$\text{SS}_{\text{Treat}} = \sum_i \left[\frac{T_{i.}^2}{n_i} \right] - \frac{T_{..}^2}{N}.$$

```
> # Note the order of the summing and squaring operations
> SSTreat <- sum(T^2/n) - (sum(T))^2/sum(n)
> SSTreat

## [1] 88.9231

> # Remember SSError is the sum of (n-1)s^2
> SSError = sum((n-1)*s^2)
> SSError

## [1] 164.67

> # Download aov2( ) and carry out the ANOVA
> source("http://waveland.com/Glover-Mitchell/aov2.txt")

## Downloaded: aov2( ), pairwise.holm.t.test( ), and pairwise.tukey.test( ).

> aov2(SSTreat, SSError, n)

## ANOVA with data as given
##
##              SS df       MS       F      cv     Pr>(F)
## Treat  88.9231  3 29.64103 6.30009 2.87419 0.00156398
## Error 164.6700 35  4.70486      NA      NA         NA
```

The F test is significant at the 0.05 level. Indeed, the P value is `Pr(>F) = 0.00156`. Mean separation is in order. The text uses the Tukey test. The appropriate source file was down loaded with `aov2( )`.

```
> pairwise.tukey.test(meanTreat, n, g, MSE = 4.70486)     # MSE from aov2

## Tukey pairwise comparisons (adjusted for any unequal sample sizes)
##
## Data: From given summary statistics
##
## q critical value for k = 4 means: q(0.05, 4, 35) = 3.814
##
## q statistics
##
##         IV     I   III
## I   2.979    NA    NA
## III 4.927 1.403    NA
## II  5.050 2.287 1.209
##
## Corresponding p values
##
##           IV        I     III
## I   0.171065       NA      NA
## III 0.007038 0.754796      NA
## II  0.005574 0.382303 0.82781
```

The test indicates the following relationship among the means:

IV	**I**	**III**	**II**
5.0^a	$7.0^{a,b}$	8.0^b	9.0^b

If you preferred to use the Bonferroni-Holm test, use

```
> pairwise.holm.t.test(meanTreat, n, g, MSE = 4.70486)    # MSE from aov2

## Pairwise comparisons using Bonferroni-Holm t tests)
##
## Data: From given summary statistics
##
## p values (order and then compare directly to alpha)
##
##             I        II       III
## II  0.344318        NA        NA
## III 0.655838 0.6558383        NA
## IV  0.169744 0.0063456 0.0067323
```

The results are the same, though the P values are different. In general, the Tukey test is more conservative than the Bonferroni-Holm test, there is a smaller chance of making a Type I error but a larger chance of making a Type II error (accepting a false null hypothesis). You should choose one or the other when you plan your analysis. Do not do both.

EXAMPLE 8.3. An endocrinologist studying genetic and environmental effects on insulin production of pancreatic tissue raised five litters of experimental mice. At age 2 months he sacrificed the mice, dissected out pancreatic tissue and treated the tissue specimens with glucose solution. The amount of insulin released from these specimens was recorded in pg/ml. Are there significant differences in insulin release among the litters?

Litter				
1	2	3	4	5
9	2	3	4	8
7	6	5	10	10
5	7	9	9	12
5	11	10	8	13
3	5	6	10	11

SOLUTION. In this example the litters are a *random sample* of mouse litters. The goal of the study is to determine if significant between-litter variability exists that could be attributed to genetic differences among litters. The individual litters are not of interest. The endocrinologist is only interested in a general statement that there is significant variability among litters (a significant genetic component) or not. This kind of study is analyzed as a random-effects or **Model II ANOVA**. For this experiment the hypotheses are

$$H_0\text{: There is not significant variability among litters, } \sigma^2_{\text{litters}} = 0$$
$$H_a\text{: There is significant variability among litters, } \sigma^2_{\text{litters}} > 0.$$

The preliminary analyses of both Model I and Model II ANOVAs are identical, but Model II ANOVAs do not require mean separation. To do the analysis, read and list at least some of the data in `http://waveland.com/Glover-Mitchell/Example08-3.txt`.

```
> data.Ex08.3 <- read.table("http://waveland.com/Glover-Mitchell/Example08-3.txt",
+ header = TRUE)
> head(data.Ex08.3, n = 2)

##   Insulin Litter
## 1       9      I
## 2       7      I

> tail(data.Ex08.3, n = 2)

##    Insulin Litter
## 24      13      V
## 25      11      V
```

The two columns are the measurement of the response (`Insulin`) and the treatment levels (`Litter`). Carry out the ANOVA using the `aov( )` command and put the result in the table `aov.Ex08.3`. Use the `summary( )` command to print out the ANOVA table.

```
> aov.Ex08.3 <- aov(Insulin ~ Litter, data = data.Ex08.3)
> summary(aov.Ex08.3)

##             Df Sum Sq Mean Sq F value Pr(>F)
## Litter       4   83.8   20.96    3.07   0.04 *
## Residuals   20  136.4    6.82
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The asterisk ∗ at the end of the `Litter` row in the ANOVA table means that the *F* test is significant at the 0.05 level. The *P* value is 0.04. Therefore there is significant variability among the litters. The Model II ANOVA, a random effects study, has a single question in mind that is *answered* with the global *F* test. Mean separation techniques are *not* done in conjunction with this type of ANOVA.

EXAMPLE 8.4. The values in the following table are measured maximum breadths of male Egyptian skulls from different epochs (based on data from *The Ancient Races of the Thebaid*, by Thomson and Randall-Maciver, 1905, Clarendon Press). Changes in head shape over time suggest that interbreeding occurred with immigrant populations. Use a 0.05 significance level to test the claim that the different epochs do not all have the same mean.

4000 BCE	1850 BCE	150 CE
131	129	128
138	134	138
125	136	136
129	137	139
132	137	141
135	129	142
132	136	137
134	138	145
138	134	137

SOLUTION. Read the data file `http://waveland.com/Glover-Mitchell/Example08-4.txt` and list a few rows.

```
> data.Ex08.4 <- read.table("http://waveland.com/Glover-Mitchell/Example08-4.txt",
+ header = TRUE)
> tail(data.Ex08.4, n = 3)

##    Breadth   Era
## 25     137 150CE
## 26     145 150CE
## 27     137 150CE
```

Carry out the ANOVA of `Breadth` by `Era` using `aov( )`.

```
> aov.Ex08.4 <- aov(Breadth ~ Era, data = data.Ex08.4)
> summary(aov.Ex08.4)

##             Df Sum Sq Mean Sq F value Pr(>F)
## Era          2    139    69.4    4.05  0.031 *
## Residuals   24    411    17.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The single ∗ at the end of the Era row in the table means that the F test is significant at the 0.05 level. The P value is 0.031.

To apply the mean function to each of the treatments use the tapply( ) function.

```
> tapply(data.Ex08.4$Breadth, data.Ex08.4$Era, mean)

##   150CE 1850BCE 4000BCE
## 138.111 134.444 132.667
```

To create a box plot of Breadth by Era use

```
> boxplot(Breadth ~ Era, data = data.Ex08.4)
```

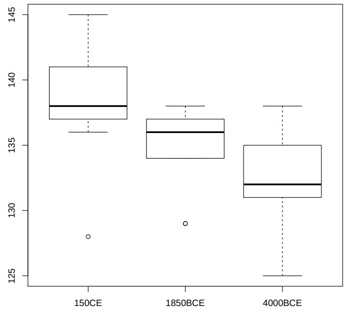


Figure 8.10: The box plot of the Breadths by Era indicates there may be a difference between the means.

Since the F test was significant, carry out Bonferroni-Holm pairwise t tests to locate differences in the mean Breadth for the various Eras. Use the pairwise.t.test( ) with the probability adjustment set to "holm".

```
> pairwise.t.test(data.Ex08.4$Breadth, data.Ex08.4$Era, p.adj = "holm")

##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  data.Ex08.4$Breadth and data.Ex08.4$Era
##
##         150CE 1850BCE
## 1850BCE 0.14  -
## 4000BCE 0.03  0.37
##
## P value adjustment method: holm
```

The entry in column 150CE and row 1850BCE is 0.14, which is the P value for the the test of the difference in means. This difference is not significant at the $\alpha = 0.05$ level. Similar remarks apply to the other entries. The table indicates that only the 150 CE and 4000 BCE eras have significantly different mean breadths.

Alternatively, the paired comparisons can be carried out using the Tukey test using TukeyHSD( ) applied to the results of the ANOVA, aov.Ex08.4.

```
> TukeyHSD(aov.Ex08.4, ordered = TRUE)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##     factor levels have been ordered
##
## Fit: aov(formula = Breadth ~ Era, data = data.Ex08.4)
##
## $Era
##                    diff      lwr      upr    p adj
## 1850BCE-4000BCE 1.77778 -3.09455  6.65010 0.638624
## 150CE-4000BCE   5.44444  0.57212 10.31677 0.026465
## 150CE-1850BCE   3.66667 -1.20566  8.53899 0.166403
```

The results table provides a 95% confidence interval for the difference in each pair of means and the corresponding P value. Only the 150 CE and 4000 BCE eras have significantly different mean breadths. The result is the same as with the Bonferroni-Holm t tests.

The Kruskal-Wallis Test in R

The function kruskal.test( ) simplifies carrying out the Kruskal-Wallis test.

kruskal.test(y ∼ g, data = mydataframe)

Details: `y` is the column of response measurements (data) and `g` is the corresponding column of treatment levels (groups) that were used. The argument `data = mydataframe` is the data frame containing the data (measurements with corresponding treatment levels) to be analyzed.

EXAMPLE 8.5. Fruit bats are important dispersal agents of many species of figs. For example, some species of fruit bats travel distances exceeding 10 km per night and thus are capable of dispersing fig seeds over long distances. Researchers in the Philippines collected uneaten fig "fragments" generated when bats were feeding, masses of masticated but ejected fig ("ejecta"), and fecal masses ("splats") beneath five fig trees, *Ficus chrysalis*, in the mountains of Negros Island in the Philippines. Germination tests were conducted on the seeds in these masses and the percentages of seeds germinating were recorded. Was there a significant difference in the germination rates of fig seeds depending how the bat interacted with the seeds? (Based on ideas in: Utzurrum, R and P. Heideman. 1991. Differential ingestion of viable vs nonviable *Ficus* seeds by fruit bats. *Biotropica* **23**(3): 311–312.)

Fragments	Ejecta	Splats
47	3	88
50	20	90
55	46	91
68	54	91
	76	92
	87	96

SOLUTION. Read the data file `http://waveland.com/Glover-Mitchell/Example08-5.txt`.

```
> data.Ex08.5 <- read.table("http://waveland.com/Glover-Mitchell/Example08-5.txt",
+ header = TRUE)
> head(data.Ex08.5, n = 2)

##   Rate Treatment
## 1   47  Fragment
## 2   50  Fragment

> tail(data.Ex08.5, n = 2)

##    Rate Treatment
## 15   92     Splat
## 16   96     Splat
```

The two columns are the response measurement or germination `Rate` and the `Treatment` levels. Carry out the Kruskal-Wallis test using the `kruskal.test( )` command. The analysis is to determine whether germination `Rate` varies by `Treatment`. The arguments of the `kruskal.test( )` command are the header names from the appropriate columns in the data table `data.Ex08.5`.

```
> kruskal.test(Rate ~ Treatment, data = data.Ex08.5)

##
##  Kruskal-Wallis rank sum test
##
## data:  Rate by Treatment
## Kruskal-Wallis chi-squared = 10.6775, df = 2, p-value = 0.004802
```

The small *P* value (less than 0.05) indicates a significant result. The question now is which treatments differ significantly. Paired comparisons are needed.

Before carrying out the comparisons, let's actually look at the ranks corresponding to the germination rates. This is accomplished by applying the `rank( )` function to `data.Ex08.5$Rate`.

Add the ranks to the original data by creating a new column in the table `data.Ex08.5` using `data.Ex08.5["Rank"]` and then place the rank data into the column.

```
> data.Ex08.5["Rank"] <- rank(data.Ex08.5$Rate)
> data.Ex08.5

##    Rate Treatment Rank
## 1    47  Fragment  4.0
## 2    50  Fragment  5.0
## 3    55  Fragment  7.0
## 4    68  Fragment  8.0
## 5     3    Ejecta  1.0
## 6    87    Ejecta 10.0
## 7    20    Ejecta  2.0
## 8    76    Ejecta  9.0
## 9    46    Ejecta  3.0
## 10   54    Ejecta  6.0
## 11   88     Splat 11.0
## 12   90     Splat 12.0
## 13   91     Splat 13.5
## 14   91     Splat 13.5
## 15   92     Splat 15.0
## 16   96     Splat 16.0
```

Calculate the means of the ranks by `Treatment` using the `tapply( )` command:

```
> tapply(data.Ex08.5$Rank, data.Ex08.5$Treatment, mean)

##   Ejecta Fragment    Splat
##  5.16667  6.00000 13.50000
```

It appears that the germination rate for `Splats` may be different than the other two rates. Though R does not provide a function to carry out the paired comparisons described in the text, we provide `pairwise.kruskal.test( )` for this task. Use the `source( )` function to download it.

`pairwise.kruskal.test(x, g)`

Details: `x` is the measurement or response data and `g` is the vector of corresponding treatment levels (groups). The test is two-sided.

In this example, `data.Ex08.5$Rate` is the response measured and `data.Ex08.5$Treatment` contains the treatment levels.

```
> source("http://waveland.com/Glover-Mitchell/pairwise.kruskal.txt")
> pairwise.kruskal.test(data.Ex08.5$Rate, data.Ex08.5$Treatment)

## Dunn's Multiple Comparisons for Kruskal-Wallis test (adjusted for any ties)
##
## Data: data.Ex08.5$Rate and data.Ex08.5$Treatment
##
## p values (compare directly to alpha)
##
##            Ejecta Fragment
## Fragment 1.000000       NA
## Splat    0.007242 0.043786
```

The output of `pairwise.kruskal.test( )` is presented in tabular form. The entry in the column `Ejecta` and row `Splat` is 0.0072. This is the P value for the test of the difference in the means of these two treatments, and is significant at the $\alpha = 0.05$ level. Similar remarks apply to the other entries. The table indicates that the mean germination rate for the `Splat` group differs significantly from the other two.

The listed P values are to be compared directly to the value of α with no further adjustment. Whereas in the text we compared the computed P value p to $\frac{\alpha}{\binom{k}{2}}$, `pairwise.kruskal.test( )` instead multiplies p by $\binom{k}{2}$ and now the comparison should be to α. That is, in R we are comparing $\binom{k}{2}p$ to α.

EXAMPLE 8.6. Watching television has been blamed for a myriad of problems, including contributing to the obesity "epidemic" in the United States. A recent experiment with Cornell undergraduates indicated that the type of TV programming watched could exacerbate the problem. Participants watching more stimulating programs featuring frequent camera cuts and high sound variation typical in action-adventure movies consumed nearly twice as much food during the program as those watching programs with less visual and audio variation. (Tal et al. Watch what you eat: Action-related television content increases food intake. *JAMA Intern Med*. Published online September 01, 2014. doi:10.1001/jamainternmed.2014.4098.)

You decide to repeat this experiment with the help of 24 classmates. You randomly assign your peers to three groups of 8. Each group is assigned to watch 30 minutes of the same videos that were used in the Cornell experiment: *The Island*, the PBS interview program *Charlie Rose*, and a silent version of *The Island*. During the viewing, your peers had access to a generous personal supply of snacks (M&Ms, cookies, carrot sticks, and grapes). The calories consumed by each student are recorded below.

The Island	Rank	Silent	Rank	Charlie Rose	Rank
280	13	167	2	142	1
285	14	226	7	196	3
297	16	233	8	219	4
298	17	240	10	220	5
304	19	262	12	224	6
325	20	294	15	236	9
332	21	344	23	260	11
333	22	351	24	299	18
Sum	142	Sum	101	Sum	57

SOLUTION. Read the data file `http://waveland.com/Glover-Mitchell/Example08-6.txt` and list the last few rows to make sure all of the data are present.

```
> data.Ex08.6 <- read.table("http://waveland.com/Glover-Mitchell/Example08-6.txt",
+ header = TRUE)
> tail(data.Ex08.6, n = 3)

##    Calories Video
## 22      236  Rose
## 23      260  Rose
## 24      299  Rose
```

Carry out the Kruskal-Wallis test using

```
> kruskal.test(Calories ~ Video, data = data.Ex08.6)

##
##  Kruskal-Wallis rank sum test
##
## data:  Calories by Video
## Kruskal-Wallis chi-squared = 9.035, df = 2, p-value = 0.01092
```

The small P value of 0.01092 indicates a significant result. Paired comparisons are needed to locate the differences.

As in Example 8.5, get a sense of the mean ranks by video. Use the `rank( )` function applied to `data.Ex08.6$Calories` to rank the calories. Create a new column in the data table using `data.Ex08.6["Rank"]` and then place the actual rank data into this column. Apply the `mean` function to the ranks grouped by factors using the `tapply( )` command.

```
> data.Ex08.6["Rank"] <- rank(data.Ex08.6$Calories)
> tapply(data.Ex08.6$Rank, data.Ex08.6$Video, mean)

## Island   Rose Silent
## 17.750  7.125 12.625
```

The Island and the *Charlie Rose* videos appear to have different means. Carry out paired comparisons as in Example 8.5 using the `pairwise.kruskal.test( )`. The response variable is `data.Ex08.6$Calories` and `data.Ex08.6$Video` contains the treatment levels.

```
> pairwise.kruskal.test(data.Ex08.6$Calories, data.Ex08.6$Video)

## Dunn's Multiple Comparisons for Kruskal-Wallis test (adjusted for any ties)
##
## Data: data.Ex08.6$Calories and data.Ex08.6$Video
##
## p values (compare directly to alpha)
##
##          Island     Rose
## Rose   0.007962       NA
## Silent 0.441536 0.359385
```

The only P value in the table that is less than $\alpha = 0.05$ occurs for the comparison of the `Island` and `Rose` treatments; consequently the median calories consumed during these videos differ significantly.

9. *Two-Factor Analysis*

Randomized Complete Block Design ANOVA

EXAMPLE 9.1. One very effective way to catch salmon commercially is by gill netting. Traditionally gill netters hang long monofilament nets vertically over the fishing grounds. As the salmon encounter these nets they become entangled in the mesh by their gill covers and quickly die. Unfortunately many other species also become ensnared in these nets. In particular, diving sea birds such as common murres and rhinoceros auklets fall prey to gill nets. In addition, seals and dolphins can be easily trapped in the almost invisible monofilament nets. The sea birds and marine mammals caught and killed in gill nets are euphemistically called "by-catch" in the fishing industry. Suppose as part of a conservation project on by-catch reduction, three modified techniques are used on boats fishing for salmon with gill nets.

I. White-topped gill nets (the top 7 feet of the net is white mesh that is considerably easier to see in the water column).

II. White-topped gill nets with pingers that emit an annoying beep similar to those made by watches with alarms.

III. Unmodified gill nets with pingers.

IV. Unmodified gill nets (standard method).

The table below lists the quantity of salmon caught per boat per day using each of these techniques.

	Catch of salmon (100 kg/boat-day)			
Day	**I**	**II**	**III**	**IV**
1	23	21	24	25
2	16	14	19	17
3	19	17	23	21
4	11	12	15	16
5	41	30	40	42
6	32	20	37	35
7	9	11	19	17
8	10	14	19	16

Are there significant differences in the quantity of salmon caught per boat per day using these techniques?

SOLUTION. The hypotheses are

$$H_0: \mu_{\text{I}} = \mu_{\text{II}} = \mu_{\text{III}} = \mu_{\text{IV}}$$

H_a: At least one pair of the μ_i's is not equal.

Read and list the data and put the result in `data.Ex09.1`.

```
> data.Ex09.1 <- read.table("http://waveland.com/Glover-Mitchell/Example09-1.txt",
+ header = TRUE)
> tail(data.Ex09.1, n = 5)

##    Day Technique Catch
## 28   G        IV    17
## 29   H         I    10
## 30   H        II    14
## 31   H       III    19
## 32   H        IV    16
```

All 32 data values appear to be there. There are three columns: the blocks (`Day`), the treatment (`Technique`), and the response measurement (`Catch`).

Carry out the ANOVA using `aov( )` and put the result in the table `aov.Ex09.1`. The analysis focuses on whether the `Catch` varies by the factors of fishing `Technique` and `Day`. The arguments in `aov( )` are the header names from the corresponding columns in the data table `data.Ex09.1`. Display the results with the `summary( )` command. Note how the model equation in the `aov( )` function expresses `Catch` as a combination of `Technique + Day`.

`Catch` now depends on both `Technique` and `Day`. This is expressed in the `aov( )` function as `Catch` ∼ `Technique + Day`.

```
> aov.Ex09.1 <- aov(Catch ~ Technique + Day, data = data.Ex09.1)
> summary(aov.Ex09.1)

##             Df Sum Sq Mean Sq F value  Pr(>F)
## Technique    3    259      86    10.2 0.00023 ***
## Day          7   2267     324    38.4 1.4e-10 ***
## Residuals   21    177       8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The F test for the difference in means for the fishing `Techniques` is significant at the $\alpha = 0.001$ level. To determine which means are different, carry out multiple comparisons of the means using the `TukeyHSD( )` function. The data come from the ANOVA results `aov.Ex09.1`. To separate the means by `Technique`, include `"Technique"` as an argument of `TukeyHSD( )`. Setting the argument `ordered = TRUE` ensures that in each comparison the smaller mean is subtracted from the larger one making the difference non-negative, which simplifies the analysis.

```
> TukeyHSD(aov.Ex09.1, "Technique", ordered = TRUE) # Note the quotes

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##     factor levels have been ordered
##
## Fit: aov(formula = Catch ~ Technique + Day, data = data.Ex09.1)
##
## $Technique
##         diff       lwr      upr    p adj
## I-II   2.750 -1.297872  6.79787 0.260763
## IV-II  6.250  2.202128 10.29787 0.001656
## III-II 7.125  3.077128 11.17287 0.000405
## IV-I   3.500 -0.547872  7.54787 0.105854
## III-I  4.375  0.327128  8.42287 0.031096
## III-IV 0.875 -3.172872  4.92287 0.930127
```

Three of the differences have P values that are less than $\alpha = 0.05$. The conclusion is that `Technique` II is significantly different from both III and IV and that `Technique` I is significantly different from III. The results table also provides a 95% confidence interval for the difference in each pair of means. As usual, only when the confidence interval does not contain 0 is the difference between the means significant. We can summarize these results using superscript notation. Any means with the same superscript are not significantly different.

II	I	IV	III
17.38^{a}	$20.12^{a,b}$	$23.62^{b,c}$	24.50^{c}

The results indicate that either method to prevent by-catch alone (white-topped gill nets without pingers or pingers alone) does not significantly reduce the salmon catch. This indicates that fishermen could use either white-topped gill nets or pingers (but not both) to reduce "by-catch" and while not significantly lowering their catch of salmon.

Factorial Design Two-Way ANOVA

EXAMPLE 9.2. In an attempt to find the most effective methods for training companion dogs for the physically challenged, an experiment was conducted to compare three different training regimes in combination with three different reward systems. All the animals in the study were Labrador retrievers and were 6 to 8 months old at the start of the experiment. Individual dogs were assigned randomly to a combination of training regime and reward system. At the end of a 10-week training period the dogs were given a standardized test to measure their ability to function as companion dogs for the visually impaired. The results of the test are given below:

	Training regime		
Reward	**I**	**II**	**III**
Praise	45	51	52
	69	50	18
	53	62	25
	51	68	32
Tangible	54	53	51
	72	63	59
	69	67	47
	66	70	42
Praise and tangible	91	69	66
	87	73	68
	89	77	70
	91	74	64

The behavioral scientist running this study would like to know which training regime, if any, is best and which reward system, if any, is best. Also, she is looking for *interactions* between training regimes and reward systems. These interactions can be either positive (synergy) or negative (interference).

SOLUTION. Read the data file `http://waveland.com/Glover-Mitchell/Example09.2.txt` and list some of the data to ensure accuracy.

```
> data.Ex09.2 <- read.table("http://waveland.com/Glover-Mitchell/Example09-2.txt",
+ header = TRUE)
> tail(data.Ex09.2)

##               Reward Training Score
## 31 PraiseAndTangible       II    77
## 32 PraiseAndTangible       II    74
## 33 PraiseAndTangible      III    66
## 34 PraiseAndTangible      III    68
## 35 PraiseAndTangible      III    70
## 36 PraiseAndTangible      III    64
```

All 36 data values appear to be present. There are three columns: the two different factors (`Reward` and `Training`) and the response measurement (`Score`).

Carry out the ANOVA using `aov( )` and put the result in the table `aov.Ex09.2`. Note the syntax: To indicate that `Score` possibly depends on an interaction between the `Reward` system and the `Training` method, we use an asterisk ∗ instead of + in the `aov( )` command: `Score ∼ Reward ∗ Training`.

```
> aov.Ex09.2 <- aov(Score ~ Reward * Training, data = data.Ex09.2)
> summary(aov.Ex09.2)

##                 Df Sum Sq Mean Sq F value  Pr(>F)
## Reward           2   4968    2484   38.14 1.4e-08 ***
## Training         2   2671    1335   20.50 3.8e-06 ***
## Reward:Training  4    583     146    2.24   0.091 .
## Residuals       27   1759      65
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

First investigate interactions. The hypotheses are

$$H_0\colon (\alpha\beta)_{ij} = 0 \text{ for all } i, j$$
$$H_a\colon (\alpha\beta)_{ij} \neq 0 \text{ for some } i, j.$$

The test statistic is

$$F_{\text{A}\times\text{B}} = F_{\text{Reward } \times \text{ Training}}.$$

This is found in the third row of the ANOVA table labelled `Reward:Training`. $F_{\text{A}\times\text{B}} = 2.24$ with a corresponding P value of 0.091. This exceeds 0.05, so *we don't have evidence for interaction here*. We accept H_0.

Now test the main effects for `Training` regimes. The hypotheses are

$$H_0\colon \mu_{\text{I}} = \mu_{\text{II}} = \mu_{\text{III}}$$
$$H_a\colon \text{At least one pair of } \mu\text{'s is not equal.}$$

From the second row of the ANOVA table, we see that the test statistic is

$$F_{\text{B}} = F_{\text{Training}} = 20.50$$

with a tiny corresponding P value of 0.0000038. We have evidence that *at least some `Training` regimes are significantly different*.

Finally, test the main effects for `Reward` systems. The hypotheses are

$$H_0\colon \mu_{\text{P}} = \mu_{\text{T}} = \mu_{\text{P+T}}$$
$$H_a\colon \text{At least one pair of } \mu\text{'s is not equal.}$$

From the first row of the ANOVA table the test statistic is

$$F_{\text{A}} = F_{\text{Reward}} = 38.14$$

with a corresponding P value of almost 0. We have evidence that *at least some `Reward` systems are significantly different*.

Mean separation techniques. Separate the main effects for `Training` regimes using Tukey's test. List the mean scores by reward system and make a box plot of these data.

```
> tapply(data.Ex09.2$Score, data.Ex09.2$Training, mean)

##     I    II   III
## 69.75 64.75 49.50
```

```
> boxplot(data.Ex09.2$Score~data.Ex09.2$Training)
```

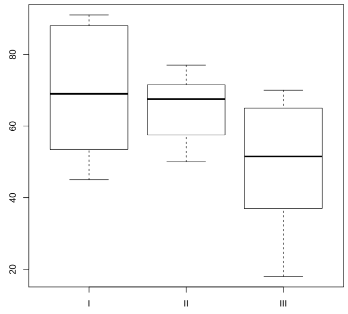


Figure 9.11: The box plot of `Score` by `Training` regime.

The box plot indicates that the results of method III may be different than the other two methods. Confirm this with Tukey's test. Remember to include the `"Training"` argument, and for convenience set `ordered = TRUE`.

```
> TukeyHSD(aov.Ex09.2, "Training", ordered = TRUE)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##     factor levels have been ordered
##
## Fit: aov(formula = Score ~ Reward * Training, data = data.Ex09.2)
##
## $Training
##         diff     lwr     upr    p adj
## II-III 15.25  7.0811 23.4189 0.000237
## I-III  20.25 12.0811 28.4189 0.000004
## I-II    5.00 -3.1689 13.1689 0.298744
```

Using the P values, `Training` regimes I and II are *not significantly different*, but they are *superior* to regime III.

III	**II**	**I**
49.5^a	64.8^b	69.8^b

Carrying out a similar analysis for the `Reward` systems, determine the means:

```
> tapply(data.Ex09.2$Score, data.Ex09.2$Reward, mean)

##            Praise PraiseAndTangible          Tangible
##           48.0000           76.5833           59.4167
```

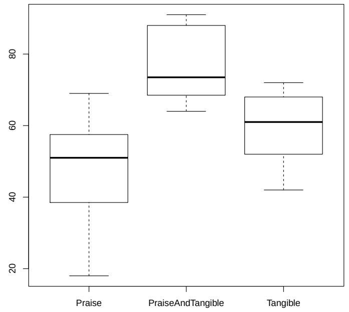


Figure 9.12: The box plot of `Score` by `Reward` system. All three means may be significantly different.

The box plot indicates that there may be differences between means. Confirm this by using

```
> TukeyHSD(aov.Ex09.2, "Reward", ordered = TRUE)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##     factor levels have been ordered
##
## Fit: aov(formula = Score ~ Reward * Training, data = data.Ex09.2)
##
## $Reward
##                               diff      lwr     upr    p adj
## Tangible-Praise            11.4167  3.24777 19.5856 0.004919
## PraiseAndTangible-Praise   28.5833 20.41444 36.7522 0.000000
## PraiseAndTangible-Tangible 17.1667  8.99777 25.3356 0.000050
```

Tukey's test indicates that all three `Reward` systems are significantly different from each other as all the P values are much smaller than 0.05. The very large F statistic ($F_A = 38.14$) was an early indication that differences here were dramatic. This is summarized as

P	**T**	**P+T**
48.0^a	59.4^b	76.6^c

Finally, we conclude that while praise plus tangible rewards increases the test scores the most, either training regime I or II is better than III and no interaction between `Training` regime and `Reward` system is indicated.

PROBLEM 9.16 (Additional Exercise). As part of a much larger study on neonatal health in America, the birth weights of 36 newborn full-term infants were recorded in kilograms. The principal factors of interest were the maternal age and the smoking status of the mother. Analyze the data below appropriately, assuming normality of the birth weights.

	Smoking Status of Mother		
Age of Mother	**Non-smoker**	**Moderate Smoker**	**Heavy Smoker**
Young < 25 years	3.63	3.35	3.23
	3.45	3.30	3.05
	3.40	3.31	3.18
	3.51	3.20	3.40
	3.60	3.25	3.19
	3.55	3.43	3.15
	$\overline{X}_{11.} = 3.523$	$\overline{X}_{21.} = 3.307$	$\overline{X}_{31.} = 3.200$
Older > 25 years	3.61	3.08	3.05
	3.49	3.10	3.00
	3.58	3.37	3.18
	3.67	3.28	3.20
	3.40	3.19	2.99
	3.18	3.31	3.25
	$\overline{X}_{12.} = 3.488$	$\overline{X}_{22.} = 3.222$	$\overline{X}_{32.} = 3.112$
$\sum_i \sum_j \sum_k X_{ijk}^2 = 395.336$		$\sum_i \sum_j \sum_k X_{ijk} = T_{...} = 119.110$	

SOLUTION. Read the data file `http://waveland.com/Glover-Mitchell/Problem09.16.txt` and list some of the data to ensure accuracy.

```
> data.Prob09.16 <- read.table("http://waveland.com/Glover-Mitchell/Problem09-16.txt",
+ header = TRUE)
> tail(data.Prob09.16, n = 3)

##      Age SmokingStatus Weight
## 34 Older         Heavy   3.20
## 35 Older         Heavy   2.99
## 36 Older         Heavy   3.25
```

There are three columns: the two different factors (`Age` and `SmokingStatus`) and the response measurement (`Weight`). Carry out the ANOVA using `aov( )` and put the result in the table `aov.Prob09.16`. Note the syntax: To indicate that `Weight` possibly depends on an interaction `Age` and `SmokingStatus`, use

```
> aov.Prob09.16 <- aov(Weight ~ Age * SmokingStatus, data = data.Prob09.16)
> summary(aov.Prob09.16)

##                   Df Sum Sq Mean Sq F value  Pr(>F)
## Age                1  0.043   0.043    3.04   0.091 .
## SmokingStatus      2  0.771   0.385   27.02 1.9e-07 ***
## Age:SmokingStatus  2  0.005   0.003    0.19   0.830
## Residuals         30  0.428   0.014
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Test for interaction. Test H_0: $(\alpha\beta)_{ij} = 0$ for all i, j versus H_0: $(\alpha\beta)_{ij} \neq 0$ for some i, j. The test statistic is

$$F_{\text{A}\times\text{B}} = F_{\text{Age} \times \text{SmokingStatus}}.$$

This is found in the third row of the ANOVA table labelled `Age:SmokingStatus`. $F_{\text{A}\times\text{B}} = 0.19$ with a corresponding P value of 0.830. This exceeds 0.05, so *we don't have evidence for interaction here.*

Test the `SmokingStatus`. H_0: $\mu_{NS} = \mu_{MS} = \mu_{HS}$ versus H_a: At least one pair of μ's is different. Since $F_{\text{SmokingStatus}} = 27.02$ has a tiny P value, reject H_0. Mean birth weights are significantly different for at least two of the levels of smoking behavior.

Test the `Age` of the mothers. H_0: $\mu_Y = \mu_O$ versus $H_a : \mu_Y \neq \mu_O$. $F_{\text{Age}} = 3.04$ has a P value $>$ 0.05, accept H_0. Mean birth weights are not significantly different for younger and older mothers. Mean separation is not required.

Separate main effects for `SmokingStatus` using Tukey's test. List the mean scores by `SmokingStatus` and make a box plot of these data.

```
> tapply(data.Prob09.16$Weight, data.Prob09.16$SmokingStatus, mean)

##      Heavy   Moderate Non-smoker
##    3.15583    3.26417    3.50583
```

```
> boxplot(data.Prob09.16$Weight ~ data.Prob09.16$SmokingStatus)
```

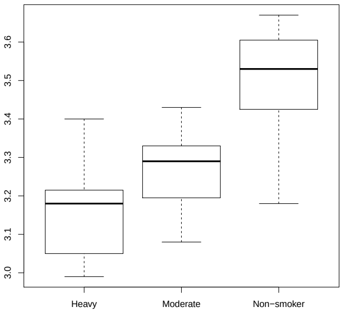


Figure 9.13: The box plot of `Weight` by `SmokingStatus`.

The box plot indicates that the non-smokers may be different than either of the smokers. Confirm this with Tukey's test. Remember to include the `"SmokingStatus"` argument, and for convenience set `ordered = TRUE`.

```
> TukeyHSD(aov.Prob09.16, "SmokingStatus", ordered = TRUE)

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##     factor levels have been ordered
##
## Fit: aov(formula = Weight ~ Age * SmokingStatus, data = data.Prob09.16)
##
## $SmokingStatus
##                        diff        lwr      upr    p adj
## Moderate-Heavy     0.108333 -0.0118396 0.228506 0.083596
## Non-smoker-Heavy    0.350000  0.2298271 0.470173 0.000000
## Non-smoker-Moderate 0.241667  0.1214938 0.361840 0.000076
```

The results show that non-smokers differ from either moderate or heavy smokers, while the difference between moderate and heavy smokers just fails to be significant (the confidence interval barely contains 0). This is a slightly different answer than given in the back of the text. The Tukey tests are more conservative than the Bonferroni-Holm tests.

The Friedman k-Sample Test: Matched Data

In R, the `friedman.test( )` function makes carrying out the Friedman test relatively easy.

`friedman.test(x ~ Treatment | Block, data = dataFrame)`

Details: `x` is the measurement or response data, `Treatment` and `Block` give the corresponding treatment levels (groups) and blocks for the data value.

EXAMPLE 9.3. A preliminary study at Lady Elliot Island on the Great Barrier Reef was conducted to investigate how different species of encrusting sponge compete for and partition the space available on the underside of small boulders in the intertidal zone. Eight boulders were examined and the percent cover by each of three different species of sponge (determined by color) was estimated by using a grid. Is there a difference in the space colonized by the different species of sponges?

Boulder	Peach	Black	Cream
1	30	35	8
2	17	10	5
3	33	18	12
4	18	34	2
5	25	20	25
6	50	15	11
7	40	20	6
8	19	5	14

SOLUTION. Read the data file `http://waveland.com/Glover-Mitchell/Example09.3.txt` and list the data.

```
> data.Ex09.3 <- read.table("http://waveland.com/Glover-Mitchell/Example09-3.txt",
+ header = TRUE)
> head(data.Ex09.3, n = 2)

##   Boulder Color Space
## 1       1 Peach    30
## 2       2 Peach    17

> tail(data.Ex09.3, n = 2)

##    Boulder Color Space
## 23       7 Cream     6
## 24       8 Cream    14
```

There are three columns: the percentage of `Space` occupied, the `Color` of the sponge, and `Boulder` number that acts as the block. Carry out the Friedman test using

```
> friedman.test(Space ~ Color | Boulder, data = data.Ex09.3)

##
##  Friedman rank sum test
##
## data:  Space and Color and Boulder
## Friedman chi-squared = 7.8065, df = 2, p-value = 0.02018
```

The result is significant; the P value is smaller than $\alpha = 0.05$. The median space occupied by sponges differs significantly by `Color`. Note: The test statistic differs slightly from the one in the text because a small adjustment has been made for the tied ranks.

Paired Comparisons

When the Friedman test indicates that we are able to reject H_0, we can use a method of paired comparisons to pinpoint where the differences lie. If we have k populations (treatments), then there are $\binom{k}{2} = \frac{k(k-1)}{2}$ possible pairs of treatments that can be compared that lead to $\binom{k}{2}$ tests of the form:

- H_0: The effects of the ith and jth treatments are the same.
- H_a: The effects of the ith and jth treatments are different.

Though R does not provide a function to carry out paired comparisons as described in the text, we provide `pairwise.friedman.test( )` to carry out this task. Use the `source( )` function to download it.

`pairwise.friedman.test(x, treatment, block)`

Details: `x` is the measurement or response data, `treatment` gives the treatment levels, and `block` gives the experimental blocks. The test is two-sided. The P values are compared directly to α.

In this example, `Space` is the response measured, each `Boulder` is a block, and each `Color` is a treatment (group).

```
> source("http://waveland.com/Glover-Mitchell/pairwise.friedman.txt")
> pairwise.friedman.test(data.Ex09.3$Space, data.Ex09.3$Color, data.Ex09.3$Boulder)

## Paired comparisons for Friedman test
##
## Data: data.Ex09.3$Space and data.Ex09.3$Color
##
## p values (compare directly to alpha)
##
##          Black     Cream
## Cream 0.507394        NA
## Peach 0.507394 0.0178786
```

The output of `pairwise.friedman.test( )` is presented in tabular form. The entry in the column `Black` and row `Cream` is 0.507. This is the P value for the the test of the difference in median `Space` occupied, and is not significant at the $\alpha = 0.05$ level. The table indicates that only sponges of colors `Peach` and `Cream` differ significantly in area they occupy on boulders.

EXAMPLE 9.4. The nature of the light reaching the rainforest floor is quite different from that in the canopy and the understory. One difference is wavelength. A study was conducted to examine the adaptations to light wavelength of 10 different fern species found on the rainforest floor. Each block consisted of 4 mature ferns of the same species. Each fern was grown in the dark for two weeks. One plant of each species was assigned to each of the 4 light treatments. Each plant was exposed to a single dose of light (wavelength measured in nm), returned to the dark, and 24 hours later the increase in the cross-sectional area of the fern tip was measured (in μm^2). Was there a significant difference in growth among the light treatments? Analyze with a Friedman test.

Species	400 nm	500 nm	600 nm	700 nm
A	960	800	950	910
B	780	750	700	630
C	1130	1040	1050	1000
D	670	740	630	610
E	790	770	700	720
F	960	720	810	820
G	1110	1000	1040	980
H	930	970	910	860
I	920	850	840	820
J	880	860	920	830

SOLUTION. Use the Friedman test at the $\alpha = 0.05$ level. Read the data file `http://waveland.com/Glover-Mitchell/Example09.4.txt` and list the data.

```
> data.Ex09.4 <- read.table("http://waveland.com/Glover-Mitchell/Example09-4.txt",
+ header = TRUE)
> tail(data.Ex09.4, n = 3)

##    Species Wavelength Area
## 38       H      700nm  860
## 39       I      700nm  820
## 40       J      700nm  830
```

The response measurement is the leaf tip `Area`, the `Wavelength` is the treatment, and each `Species` acts as a block. Analyze using the `friedman.test( )`.

```
> friedman.test(Area ~ Wavelength | Species, data = data.Ex09.4)

##
##  Friedman rank sum test
##
## data:  Area and Wavelength and Species
## Friedman chi-squared = 15.96, df = 3, p-value = 0.001156
```

The result is significant since the P value is much smaller than $\alpha = 0.05$. Paired comparisons should be carried out using the `pairwise.friedman.test( )` as in Example 9.3.

```
> pairwise.friedman.test(data.Ex09.4$Area, data.Ex09.4$Wavelength,
+ data.Ex09.4$Species)

## Paired comparisons for Friedman test
##
## Data: data.Ex09.4$Area and data.Ex09.4$Wavelength
##
## p values (compare directly to alpha)
##
##             400nm    500nm    600nm
## 500nm 0.226001533       NA       NA
## 600nm 0.146060302 1.000000       NA
## 700nm 0.000407074 0.340481 0.499587
```

Only the shortest and longest wavelengths produce significantly different growth.

10. Linear Regression and Correlation

Introduction

In this chapter we present analyses to determine the *strength of relationship* between two variables. In the case of linear regression, we will examine the amount of variability in one variable (Y, the dependent variable) that is explained by changes in another variable (X, the independent variable). Correlation analysis is used when both variables are experimental and measured with error.

Linear Regression

To carry out a linear regression analysis, make use of `R`'s linear model function `lm( )`. Remember in carrying out such an analysis the assumption is that the dependent variable Y is a linear function of X and, thus, there exist constants a and b so that

$$Y = a + bX.$$

Here b is the slope and a is the Y-intercept. (You may have written the equation of a line as $y = mx + b$, where m is the slope and b is the intercept. The idea is the same, but in the context of regression different letters are used.)

```
lm(y ~ x, data = dataFrame)
```

Details: `y` is the measurement or response data while `x` is the independent variable typically under the experimenter's control, and `data = dataFrame` specifies the data frame in which `x` and `y` reside. Note the order of the input variables, `y` is entered first and `x` second. Reversing them changes the result.

It is also helpful to make scatterplots of the data in a regression. This can be quickly accomplished with `R`'s `plot( )` command.

```
plot(x, y, main = "Title", xlab = "x-label", ylab = "y-label")
```

```
plot(dataFrame, main = "Title", xlab = "x-label", ylab = "y-label")
```

Details: In the first case `x` is the independent variable and `y` is the measurement or response data. Note the order of the input variables, `x` is entered first and `y` second. This is the reverse of the order for `lm( )`. In the second case, `dataFrame` is a data frame with the independent variable `x` in the first column and the corresponding dependent or response variable `y` in the second column. The remaining arguments are optional, but useful. `main = "Title"` places the "Title" at the top of the plot and `xlab = "x-label"` and `ylab = "y-label"` are are used to label the horizontal and vertical axes. There are other optional arguments that are discussed later.

EXAMPLE 10.1. An introductory biology student wishes to determine the relationship between temperature and heart rate in the common leopard frog, *Rana pipiens*. He manipulates the temperature in 2° increments ranging from 2 to 18°C and records the heart rate (beats per minute) at each interval. His data are presented below. Complete a regression analysis.

	X	Y
Recording number	Temperature (°Celsius)	Heart rate (bpm)
1	2	5
2	4	11
3	6	11
4	8	14
5	10	22
6	12	23
7	14	32
8	16	29
9	18	32

SOLUTION. Start by reading the data and listing it.

```
> data.Ex10.1 <- read.table("http://waveland.com/Glover-Mitchell/Example10-1.txt",
+ header = TRUE)
> tail(data.Ex10.1, n = 3)

##   Temperature HrtRt
## 7          14    32
## 8          16    29
## 9          18    32
```

Note the form of the data: Each row consists of a temperature and the corresponding heart rate. To determine whether there may be a linear relation between these two variables, begin by making a scatterplot of the data.

```
> plot(data.Ex10.1$Temperature, data.Ex10.1$HrtRt, main = "Heart Rate
vs Temperature", xlab = "Temperature (C)", ylab = "BPM")
```

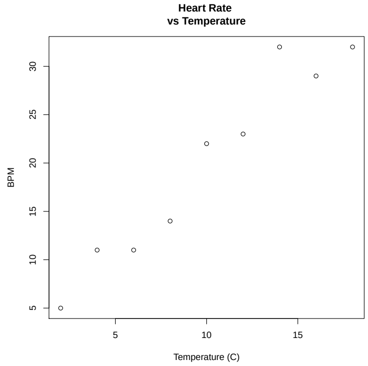


Figure 10.14: A scatterplot of `data.Ex10.1$HrtRt` versus `data.Ex10.1$Temperature`.

As noted above, the following command also would have plotted the data.

```
> plot(data.Ex10.1, main = "Heart Rate vs Temperature",
+ xlab = "Temperature (C)", ylab = "BPM")
```

Next, carry out the preliminary calculations for the regression analysis using the `lm( )` function. The response variable is `HrtRt` and the independent variable is `Temperature`.

```
> lm.Ex10.1 <- lm(HrtRt ~ Temperature, data = data.Ex10.1)
> lm.Ex10.1       # print the results

##
## Call:
## lm(formula = HrtRt ~ Temperature, data = data.Ex10.1)
##
## Coefficients:
## (Intercept)  Temperature
##        2.14         1.78
```

The results state that the intercept of the least squares regression line is $a = 2.14$ and the slope is $b = 1.78$. Thus, the equation of the regression line is

$$Y = 2.14 + 1.78X \text{ or, more precisely, } \hat{Y}_i = 2.14 + 1.78X_i.$$

In the text, the equation of the regression line was given as

$$\hat{Y}_i = 19.9 + 1.78(X_i - 10.0).$$

But this is the same as

$$\hat{Y}_i = 19.9 + 1.78X_i - 17.8 = 2.1 + 1.78X_i,$$

which is the equation that R yields up to rounding error.

To plot the regression line use an additional function with the `plot( )` command.

```
abline(a, b, col = "color")
```

Details: `a` is the intercept of the line and `b` is the slope, hence the name `abline( )`. The optional argument `col = "color"` causes the line to be drawn in the selected color, with the default being `"black"`. The command should immediately follow a `plot( )` command and the line will be included in the plot.

In the context of this regression question, `a` and `b` can be specified by using `lm.Ex10.1`.

```
> plot(data.Ex10.1, main = "Heart Rate vs Temperature",
+ xlab = "Temperature (C)", ylab = "BPM")
> abline(lm.Ex10.1, col = "red")  # lm.Ex10.1 consists of intercept a and slope b
```

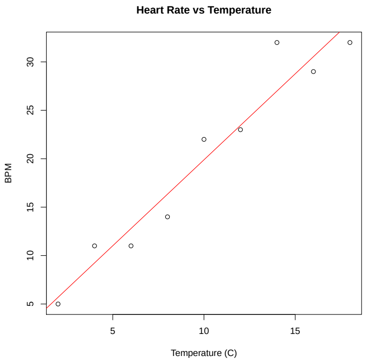


Figure 10.15: A scatterplot of `data.Ex10.1$HrtRt` versus `data.Ex10.1$Temperature` with regression line.

After the regression equation is calculated, test whether or not it explains a significant portion of the variability in the Y's. The hypotheses are H_0: $\beta = 0$ and H_a: $\beta \neq 0$. The test is carried out using ANOVA with the `aov( )` command as in Chapter 8.

```
> aov.Ex10.1 <- aov(HrtRt ~ Temperature, data = data.Ex10.1)
> summary(aov.Ex10.1)

##             Df Sum Sq Mean Sq F value   Pr(>F)
## Temperature  1    756     756     109 0.000016 ***
## Residuals    7     49       7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The P value is tiny, so H_0 is rejected: A significant portion of the variability in heart rate is explained by linear regression on temperature.

EXAMPLE 10.3. For Example 10.1, find the following:

- 95% confidence interval for β.
- 95% confidence interval for $\mu_{Y|9^\circ}$, the mean heart rate at 9°C.
- 95% confidence interval for $\mu_{Y|17^\circ}$, the mean heart rate at 17°C.

SOLUTION. To find a 95% confidence interval for β, use the `confint( )` function.

```
confint(linear.model, level = 0.95)
```

Details: `linear.model` contains the results of a linear regression model and the optional argument `level` is the desired confidence level. The default is `0.95`.

In this example, the linear regression results are contained in `lm.Ex10.1`. Since the default confidence level is 0.95, simply use

```
> confint(lm.Ex10.1)

##                 2.5 %  97.5 %
## (Intercept) -2.39401 6.67179
## Temperature  1.37224 2.17776
```

While our estimate of the slope is 1.78, we are 95% confident that the true slope β is between 1.372 and 2.178. As expected from the ANOVA table, 0 is not included in this interval. For a 0.99 confidence interval use

```
> confint(lm.Ex10.1, level = 0.99)

##                 0.5 %  99.5 %
## (Intercept) -4.56950 8.84728
## Temperature  1.17894 2.37106
```

To find a 95% confidence interval for $\mu_{Y|9^\circ}$ requires another new command. First create a data frame containing the `Temperature` for which you want the confidence interval.

```
> new <- data.frame(Temperature = 9)
> new

##   Temperature
## 1           9
```

Now use the `predict( )` function to determine the confidence interval.

`predict(linear.model, newData, interval = "confidence", level = 0.95)`

Details: `linear.model` contains the results of a linear regression model, `newData` is a data frame containing the values for which you want the confidence level. If `newData` is missing, the `x` values from `linear.model` are used. The optional argument `interval = "confidence"` indicates that the output should contain a confidence interval as well as the predicted mean, and the optional argument `level` is the desired confidence level. The default is `0.95`.

The linear regression results are contained in `lm.Ex10.1`, so use

```
> predict(lm.Ex10.1, new, interval = "confidence")        # note the quotes!

##       fit     lwr     upr
## 1 18.1139 15.9954 20.2324
```

Using the regression equation, the predicted mean heart rate at 9° is $\widehat{\mu}_{Y|9^\circ} = 18.11$. The 95% confidence interval is $[16.00, 20.23]$.

To find a 95% confidence interval for $\mu_{Y|17^\circ}$, we could simply repeat the process above. However, it is possible to determine confidence intervals for multiple temperatures in a single command by including all of the temperatures in the `new` data frame. For example, to find the confidence intervals for $\mu_{Y|9^\circ}$, $\mu_{Y|17^\circ}$, and $\mu_{Y|21^\circ}$, first create the data frame with the three temperatures.

```
> new <- data.frame(Temperature = c(9, 17, 21))
> new

##   Temperature
## 1           9
## 2          17
## 3          21
```

Then use `predict( )` exactly as above.

```
> predict(lm.Ex10.1, new, interval = "confidence")

##       fit     lwr     upr
## 1 18.1139 15.9954 20.2324
## 2 32.3139 28.8104 35.8174
## 3 39.4139 34.5196 44.3081
```

This produces the predicted means (in the `fit` column) and the lower and upper bounds for 95% confidence intervals (the `lwr` and `upr` columns) for each temperature. Note the increase in confidence interval width as X gets farther from $\overline{X}$.

Many statistics programs include options to graph confidence bands and R is no exception. The `lines( )` command can be used to draw the band boundaries.

`lines(x, y, col = "color", lty = "lineType")`

Details: `x` is the independent variable and `y` is the dependent variable; `col = "color"` specifies the color; and `lty = "lineType"` specifies the line type such as `"solid"`, `"dashed"`, `"dotted"`, `"dotdash"`, `"longdash"`, or `"twodash"`. There are additional optional arguments that are not relevant at this point. The command should immediately follow a `plot( )` command and it will automatically be included in the plot.

In this example, the independent variable x is `data.Ex10.1$Temperature`, which is used by default since no `newData` are specified. The dependent variable comes from creating the y-coordinates of the confidence intervals for the mean heart rate at the temperatures used. So first we must obtain these confidence intervals using the `predict( )` function with the default `level = 0.95`.

```
> # put the result into a data frame
> fitted <- as.data.frame(predict(lm.Ex10.1, interval = "confidence"))
> fitted

##        fit      lwr      upr
## 1  5.68889  1.85386  9.52392
## 2  9.23889  6.05056 12.42722
## 3 12.78889 10.15808 15.41970
## 4 16.33889 14.10851 18.56927
## 5 19.88889 17.80905 21.96873
## 6 23.43889 21.20851 25.66927
## 7 26.98889 24.35808 29.61970
## 8 30.53889 27.35056 33.72722
## 9 34.08889 30.25386 37.92392
```

The `lwr` and `upr` columns will be the y-coordinates for the boundary curves. The x-coordinates of the boundary curves are the original temperatures in `data.Ex10.1$Temperature`, which is used by default since no `newData` are specified. To create the final graph, plot the original data and then the regression line as above. Finally add the boundary curves of the confidence bands with `lines( )` commands.

```
> # the original data
> plot(data.Ex10.1, main = "Heart Rate versus Temperature",
+ xlab = "Temperature (C)", ylab = "BPM")
> # the least squares regression line
> abline(lm.Ex10.1, col = "red")
> # lower boundary curve
> lines(data.Ex10.1$Temperature, fitted$lwr, col = "blue", lty = "longdash")
> # upper boundary curve
> lines(data.Ex10.1$Temperature, fitted$upr, col = "blue", lty = "longdash")
```

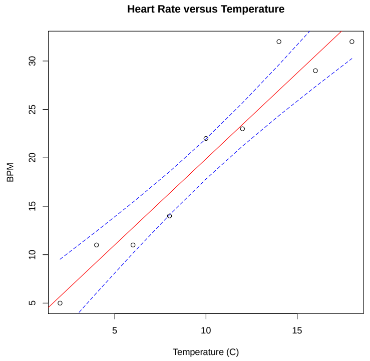


Figure 10.16: A scatterplot of `data.Ex10.1$HrtRt` versus `data.Ex10.1$Temperature` with regression line and 95% confidence interval boundaries.

Simple Linear Correlation Analysis: Pearson Correlation Coefficient

Correlation analysis is used to measure the intensity of association observed between any pair of variables. A widely used index of the association of two quantitative variables is the **Pearson product-moment correlation coefficient**.

EXAMPLE 10.2. A malacologist interested in the morphology of West Indian chitons, *Chiton olivaceus*, measured the length (anterior-posterior) and width of the eight overlapping plates composing the shell of 10 of these animals. Her data (in cm) are presented below:

Animal	Length	Width
1	10.7	5.8
2	11.0	6.0
3	9.5	5.0
4	11.1	6.0
5	10.3	5.3
6	10.7	5.8
7	9.9	5.2
8	10.6	5.7
9	10.0	5.3
10	12.0	6.3

Analyze these data as a correlation problem.

SOLUTION. Start by reading the data and listing it.

```
> data.Ex10.2 <- read.table("http://waveland.com/Glover-Mitchell/Example10-2.txt",
+ header = TRUE)
> data.Ex10.2

##    Length Width
## 1    10.7   5.8
## 2    11.0   6.0
## 3     9.5   5.0
## 4    11.1   6.0
## 5    10.3   5.3
## 6    10.7   5.8
## 7     9.9   5.2
## 8    10.6   5.7
## 9    10.0   5.3
## 10   12.0   6.3
```

To determine whether there may be a linear relation between these two variables, begin by making a scatterplot of the data.

```
> plot(data.Ex10.2$Length, data.Ex10.2$Width, main = "Scatterplot Width
versus Length", xlab = "Length (cm)", ylab = "Width (cm)")
```

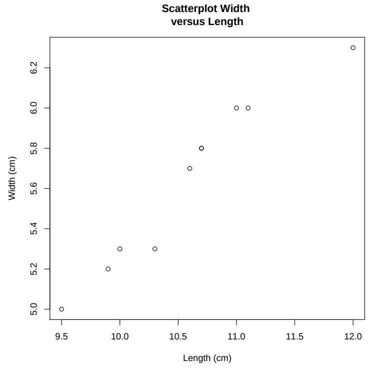


Figure 10.17: A scatterplot of `data.Ex10.2$Width` versus `data.Ex10.2$Length`.

To carry out the correlation analysis use

`cor.test(x, y, method = "testMethod", conf.level = 0.95)`

Details: `x` and `y` are paired data vectors of the same length; `method = "testMethod"` specifies the correlation method to be used. The default method is `"pearson"` and the other options are the non-parametric methods `"kendall"` and `"spearman"`. As usual `conf.level` specifies the confidence level at which the test is to be run with the default being `0.95`, which is equivalent to $\alpha = 0.05$. There are other options that will be detailed when the non-parametric methods are discussed.

In this example the data are contained in `data.Ex10.2$Length` and `data.Ex10.2$Width`. Use `method = "pearson"` and the default `conf.level = 0.95`. To determine the correlation coefficient and test the hypotheses H_0: $\rho = 0$ and H_a: $\rho \neq 0$ with $\alpha = 0.05$ use

```
> cor.test(data.Ex10.2$Length, data.Ex10.2$Width, method = "pearson",
+ conf.level = 0.95)

##
##  Pearson's product-moment correlation
##
## data:  data.Ex10.2$Length and data.Ex10.2$Width
## t = 11.136, df = 8, p-value = 3.781e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.871327 0.992922
## sample estimates:
##      cor
## 0.969226
```

The correlation coefficient (sample estimate) is $r = 0.9692$. The very small P value of 0.0000038 means that H_0 should be rejected: There is a *strong linear correlation* between length and width of chiton shells. The confidence interval for ρ is $[0.871, 0.993]$. The interval does not contain 0, confirming that the null hypothesis should be rejected.

Correlation Analysis Based on Ranks

Kendall's Measure of Correlation τ

The Kendall correlation coefficient depends on a direct comparison of the n observations (X_i, Y_i) with each other.

EXAMPLE 10.5. There have been several studies of the relative age effect on various types of achievement (e.g., academic, sport, and social). Briefly, relative age effects may occur whenever there are minimum age requirements for participation in an activity. For example, for entrance into kindergarten in Geneva, New York, where the authors reside, a child must be 5 by December 1 of the current school year. Consequently, on December 1, children in the same kindergarten class can range in age from a minimum of 5 years, 0 days (those born on December 1) to 5 years, 364 days (those born on December 2). Similarly, to participate on athletic teams (soccer, little league, etc.) children must attain a minimum age by a certain date, again leading to a 1-year age difference in initial cohorts of participants. Are these differences in ages associated with differences in performance, that is, are there relative age effects?

A study by DeMeis and Stearns examined relative age effects on academic and social performance in Geneva. The data in the following table come from one part of their study, which examined the number of students in grades K through 4 evaluated for the district's Gifted and Talented Student Program.

The first and second columns give the month after the cut-off date in which the student was born. The third column lists the number of students in each month cut-off category in grades K through 4 evaluated for the district's Gifted and Talented Student Program.

Birth month	Month after cut-off	Students evaluated
December	1	53
January	2	47
February	3	32
March	4	42
April	5	35
May	6	32
June	7	37
July	8	38
August	9	27
September	10	24
October	11	29
November	12	27

The table shows that, in general, the older students (those with birth dates in the first few months after the cut-off date) tend to be overrepresented and younger students underrepresented in those evaluated for the district's Gifted and Talented Student Program. For example, the month of December (the first month after the cut-off date) had the most referrals. Is there a correlation between these two measurements? Determine Kendall's correlation coefficient τ and determine whether it is significantly different from 0 at the $\alpha = 0.05$ level. (Based on data reported in: DeMeis, J. and E. Stearns. 1992. Relationship of school entrance age to academic and social performance. *The Journal of Educational Research*, **86:** 20–27.)

SOLUTION. Start by reading the data and listing it.

```
> data.Ex10.5 <- read.table("http://waveland.com/Glover-Mitchell/Example10-5.txt",
+ header = TRUE)
> data.Ex10.5

##        Month AfterCutOff Students
## 1   December           1       53
## 2    January           2       47
## 3   February           3       32
## 4      March           4       42
## 5      April           5       35
## 6        May           6       32
## 7       June           7       37
## 8       July           8       38
## 9     August           9       27
## 10 September          10       24
## 11   October          11       29
## 12  November          12       27
```

To carry out a correlation analysis using Kendall's measure of correlation, again use `cor.test( )` but with `method = "kendall"`. In this example the data are contained in `data.Ex10.5$AfterCutOff` and `data.Ex10.5$Students`. To determine the correlation coefficient and test the hypotheses H_0: $\tau = 0$ and H_a: $\tau \neq 0$ with $\alpha = 0.05$ use

```
> cor.test(data.Ex10.5$AfterCutOff, data.Ex10.5$Students, method = "kendall",
+ conf.level = 0.95)

## Warning in cor.test.default(data.Ex10.5$AfterCutOff, data.Ex10.5$Students, :
Cannot compute exact p-value with ties

##
##  Kendall's rank correlation tau
##
## data:  data.Ex10.5$AfterCutOff and data.Ex10.5$Students
## z = -2.7559, p-value = 0.005853
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## -0.615457
```

The correlation coefficient (sample estimate) is $\tau = -0.6155$. It appears that there is a moderately strong negative correlation between month after cut-off date and referrals for gifted evaluation. The P value of 0.0059 means that H_0 should be rejected.

Why does the warning appear in the output? Look back at the original data and notice the ties. The pairs of months February and May and August and November had the same number of students. The presence of ties makes it impossible to compute an exact P value. If you know in advance that there will be ties in the data, this warning can be avoided by using the additional argument: `exact = FALSE`, so that the full command could have been

```
> cor.test(data.Ex10.5$AfterCutOff, data.Ex10.5$Students, method = "kendall",
+ exact = FALSE, conf.level = 0.95)
```

Spearman's Coefficient r_s

Another common measure of correlation is **Spearman's rank correlation coefficient r_s**.

EXAMPLE 10.8. A study by Musch and Hay examined relative age effects in soccer in various northern and southern hemisphere countries. For each country, a sample consisting of all players in the highest professional soccer league was investigated. Their data for Germany are given below. A cut-off date of August 1 applies in Germany. Since participation cut-off dates vary by country, foreign players were excluded from their analysis. For each country, the distribution of professional players' birthdays was computed by month. These birthday distributions were then compared with that of the general population of that country.

The first column is the month after the cut-off date in which the player was born. The second column is the number of professional soccer players born in the respective months of the competition year. The third column is the number of soccer players that would be expected on the basis of official birth statistics assuming that the birth distribution of soccer professionals is the same as that of the general population, and that no relative age effect exists. The fourth column is the difference between this expected and the observed number of players.

Month	Actual players	Expected players	Difference
1	37	28.27	8.73
2	33	27.38	5.62
3	40	26.26	13.74
4	25	27.60	−2.60
5	29	29.16	−0.16
6	33	30.05	2.95
7	28	31.38	−3.38
8	25	31.83	−6.83
9	25	31.16	−6.16
10	23	30.71	−7.71
11	30	30.93	−0.93
12	27	30.27	−3.27

The table shows that, in general, the older players in the 1-year cohorts tend to be over-represented and younger players underrepresented in the total number of professional soccer players in Germany. Compute and interpret r_s for these data where X is the month and Y is the difference between actual and expected numbers of professional players. (Based on data reported by: Musch, J. and R. Hay. 1999. The relative age effect in soccer: Cross-cultural evidence for a systematic discrimination against children born late in the competition year. *Sociology of Sport Journal*, **16:** 54–64.)

SOLUTION. Start by reading the data and listing it.

```
> data.Ex10.8 <- read.table("http://waveland.com/Glover-Mitchell/Example10-8.txt",
+ header = TRUE)
> head(data.Ex10.8, n = 3)

##   Month Actual Expected
## 1     1     37    28.27
## 2     2     33    27.38
## 3     3     40    26.26
```

To be able to carry out the test, first calculate the difference between the actual number of players and the expected number of players for each month and put the result into a new column of `data.Ex10.8`.

```
> data.Ex10.8["Difference"] <- data.Ex10.8$Actual - data.Ex10.8$Expected
> data.Ex10.8

##    Month Actual Expected Difference
## 1      1     37    28.27       8.73
## 2      2     33    27.38       5.62
## 3      3     40    26.26      13.74
## 4      4     25    27.60      -2.60
## 5      5     29    29.16      -0.16
## 6      6     33    30.05       2.95
## 7      7     28    31.38      -3.38
## 8      8     25    31.83      -6.83
## 9      9     25    31.16      -6.16
## 10    10     23    30.71      -7.71
## 11    11     30    30.93      -0.93
## 12    12     27    30.27      -3.27
```

To carry out the correlation analysis using Spearman's method, use `cor.test( )` and set `method = "spearman"`. In this example the data are contained in `data.Ex10.8$Month` and `data.Ex10.8$Difference`. To determine the correlation coefficient and test the hypotheses H_0: $\rho_s = 0$ and H_a: $\rho_s \neq 0$ with $\alpha = 0.05$ use

```
> cor.test(data.Ex10.8$Month, data.Ex10.8$Difference, method = "spearman",
+ conf.level = 0.95)

##
##  Spearman's rank correlation rho
##
## data:  data.Ex10.8$Month and data.Ex10.8$Difference
## S = 494, p-value = 0.01
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## -0.727273
```

The correlation coefficient (sample estimate) is $r_s = -0.7273$. The P value for the test is 0.01, so reject H_0 and accept H_a. The negative Spearman rank correlation ($r_s = -0.7273$) is significant and indicates that there is an excess in the number of "goliath" players (those born early in the competition year) and a lack of players born late in the competition year among professional soccer players in Germany.

11. Goodness of Fit Tests for Categorical Data

This final chapter introduces techniques to analyze *categorical* or *count* data.

The Binomial Test

The binomial test applies to data that can be classified into two mutually exclusive categories. The test presumes that there are estimates for the proportions of the population falling into each of the categories prior to conducting the test. The purpose of the test is to obtain evidence that either supports or refutes the supposition that these hypothesized proportions are the actual proportions. The binomial test is carried out in R by using the `binom.test( )` function.

```
binom.test(x, n, p = proportion, alternative = "two.sided", conf.level = 0.95)
```

> Details: `x` is the number of successes; `n` specifies the number of trials; `p = proportion` specifies the hypothesized probability of success, with default `p = 0.5`; `alternative` indicates the alternative hypothesis where the default is `"two.sided"` and the other options are `"greater"` and `"less"`; and `conf.level` is the confidence level for the returned confidence interval. The default is `0.95`, which is equivalent to $\alpha = 0.05$.

EXAMPLE 11.2. The severe drought of 1987 in the U.S. affected the growth rate of established trees. It is thought that the majority of the trees in the affected areas each have a 1987 growth ring that is less than half the size of the tree's other growth rings. A sample of 20 trees is collected and 15 have this characteristic. Do these data support the claim?

SOLUTION. Use the binomial test with $\alpha = 0.05$, that is, with the default `conf.level = 0.95`. Let p denote the proportion of trees with a 1987 growth ring that is less than half their usual size. The alternative hypothesis is that "the majority of the trees" have this property, that is, H_a: $p > 0.5$. The null hypothesis is H_0: $p \leq 0.5$. The number of successes is `x = 15`, the number of trials is `n = 20`, and the hypothesized probability of success is the default `p = 0.5`. This is a right-tailed test, so `alternative = "greater"`. There are no data to read for the problem. To carry out the test use

```
> binom.test(x = 15, n = 20, p = 0.5, alternative = "greater", conf.level = 0.95)
```

or, taking advantage of the default settings, use

```
> binom.test(x = 15, n = 20, alternative = "greater")

##
##  Exact binomial test
##
## data:  15 and 20
## number of successes = 15, number of trials = 20, p-value = 0.02069
## alternative hypothesis: true probability of success is greater than 0.5
## 95 percent confidence interval:
##  0.544418 1.000000
## sample estimates:
## probability of success
##                   0.75
```

Since the P value is less than $\alpha = 0.05$, reject H_0. There is evidence that the majority of trees have growth rings for 1987 less than half their usual size. A 95 percent right-tailed confidence interval for the proportion p of trees with a 1987 growth ring that is less than half their usual size is $[0.5444, 1.0000]$ and does not contain 0.5. We are 95 percent certain that the true proportion is at least 0.5444.

EXAMPLE 11.3. Pardalotes are small (8–12 cm) birds that feed mostly in the outer canopy, high in eucalyptus trees. However, they nest in holes dug in earth banks. ("Pardalote" comes from the Greek word for "spotted.") There are two different races of striated pardalote, *Pardalotus striatus*, in southeastern Queensland. Suppose that historical records indicate that race A comprised 70% of the population. A small census at Mt. Coot-tha locates 18 pardalotes: 11 of race A and 7 of race B. Do these figures indicate any difference from the historical pattern?

SOLUTION. A two-sided binomial test with $\alpha = 0.05$ is appropriate. The hypothesized proportion of race A is `p = 0.7`. The null hypothesis is H_0: $p = 0.7$ while the alternative is H_a: $p \neq 0.7$. The test statistic or number of successes is `x = 11` and the number of trials is `n = 18`. To carry out the test use

```
> binom.test(x = 11, n = 18, p = 0.7)

##
##  Exact binomial test
##
## data:  11 and 18
## number of successes = 11, number of trials = 18, p-value = 0.4429
## alternative hypothesis: true probability of success is not equal to 0.7
## 95 percent confidence interval:
##  0.357451 0.827014
## sample estimates:
## probability of success
##               0.611111
```

The P value is 0.4429, which is much greater than $\alpha = 0.05$. There is not sufficient evidence to support a claim of a change in the population proportions of the two races. The 95 percent confidence interval for the proportion of race A pardalotes is $[0.357, 0.827]$ and so contains the historical proportion of 0.7.

EXAMPLE 11.4. In 2011, according to the U.S. Census Bureau 30.4% of adults residing in the United States had received at least a bachelor's degree. A high-tech firm is considering relocating its headquarters to a city where it is believed that there is a higher than average proportion of college graduates. In a random sample of $n = 480$ individuals in this city, 169 claim to have a bachelor's degree. Is there evidence that citizens of this city are more highly-educated than the population as a whole?

SOLUTION. A right-tailed binomial test with $\alpha = 0.05$ is appropriate with null and alternative hypotheses H_0: $p \leq 0.304$ and H_a: $p > 0.304$, respectively.

```
> binom.test(x = 169, n = 480, p = 0.304, alternative = "greater")

##
##  Exact binomial test
```

```
##
## data:  169 and 480
## number of successes = 169, number of trials = 480, p-value =
## 0.01331
## alternative hypothesis: true probability of success is greater than 0.304
## 95 percent confidence interval:
##  0.315934 1.000000
## sample estimates:
## probability of success
##               0.352083
```

The P value for the test is 0.013. Since this value is less than $\alpha = 0.05$, reject H_0. Equivalently, a 95 percent confidence interval $[0.316, 1.000]$ does not contain the hypothesized proportion $p = 0.304$ of college graduates. There is evidence that citizens of this city are more highly-educated than the U.S. population as a whole.

EXAMPLE 11.5. Assume that the hypotheses are the same as in Example 11.3, but that the sample size is $n = 180$ birds: 110 of race A and 70 of race B. Would these figures indicate any difference from the historical pattern?

SOLUTION. A two-sided binomial test with $\alpha = 0.05$ is appropriate and the null and alternative hypotheses remain the same as in Example 11.3. The test statistic or number of successes is `x = 110`, the number of trials is `n = 180`. To carry out the test use

```
> binom.test(x = 110, n = 180, p = 0.7)

##
##  Exact binomial test
##
## data:  110 and 180
## number of successes = 110, number of trials = 180, p-value =
## 0.01149
## alternative hypothesis: true probability of success is not equal to 0.7
## 95 percent confidence interval:
##  0.535762 0.682736
## sample estimates:
## probability of success
##               0.611111
```

The P value is 0.0115, which is smaller than $\alpha = 0.05$. The 95 percent confidence interval for the proportion of race A pardalotes is is $[0.536, 0.683]$ and does not contain the historical proportion of 0.7. Reject H_0 this time. There is evidence that there has been a change in the historical pattern. The benefit of a larger sample size can be seen in the much narrower confidence interval for the proportion p of race A pardalotes. More work pays off.

Comparing Two Population Proportions

There are many situations where we may want to compare the proportions of two groups that have some specified characteristic. The proportions test is carried out in R by using the `prop.test( )` function.

`prop.test(x, n, alternative = "two.sided", conf.level = 0.95)`

> Details: `x` is a two-dimensional vector specifying the successes for each group; `n` is a two-dimensional vector specifying the number of trials for each group; `alternative` indicates the alternative hypothesis where the default is `"two.sided"` and the other options are `"greater"` and `"less"`; and `conf.level` is the confidence level for the returned confidence interval. The default is `0.95`, which is equivalent to $\alpha = 0.05$.

However, `prop.test( )` uses a different method (chi-square) to compute a P value than the method illustrated in the text (Z-statistic). So we have provided the function `z.prop.test( )` that uses the latter method. The two functions differ only slightly in the the P value computed.

`z.prop.test(x, n, alternative = "two.sided", conf.level = 0.95)`

Details: The arguments are identical to those for `prop.test( )`.

EXAMPLE 11.6. A stent is a small wire mesh tube, often inserted in an artery, that acts as a scaffold to provide support to keep the artery open. Stents are a common therapy used to keep open totally occluded arteries in coronary patients. They may be implanted even several days after a heart attack under the assumption that any procedure that increases blood flow will lead to an improvement in survival rate.

A study was carried out to determine whether the time when stents are implanted in patients' arteries changes the effectiveness of the treatment. In the study 2166 stable patients who had had heart attacks in the previous 3 to 28 days and had a total occlusion of an artery related to this attack were randomly assigned to two groups. Group 1 consisted of 1082 patients who received stents and optimal medical therapy, while Group 2 consisted of 1084 patients who received medical therapy alone with no stents. The results were that 171 of the patients in Group 1 and 179 patients in Group 2 had died by the end of four years.

Researchers had expected to find a reduction in death rate for patients receiving stents under the conditions described. Was there any evidence for this?

SOLUTION. A left-tailed, two-sample proportions test at the $\alpha = 0.05$ level is appropriate. The hypotheses are H_0: $p_1 \geq p_2$ and H_a: $p_1 < p_2$. Create the vectors holding the number of successes and the number of trials.

```
> x <- c(171, 179)                # successes (deaths) for Group 1 and  Group 2
> n <- c(1082, 1084)              # trials for Group 1 and  Group 2
> source("http://waveland.com/Glover-Mitchell/z.prop.txt") # Download source file

## Downloaded: z.prop.test( ).

> z.prop.test(x, n, alternative = "less")

##
##  2-sample Z-test for equality of proportions
##
## data:  x out of n
## z-statistic = -0.4481, p-value = 0.327
## alternative hypothesis: less
## 95 percent confidence interval:
##  -1.0000000  0.0189272
## sample estimates:
##   prop 1   prop 2
## 0.158041 0.165129
```

The resulting P value is 0.327 and is much larger than $\alpha = 0.05$. Do not reject H_0; there is no evidence for a significant reduction in death rate with the implantation of stents in patients 3 to 28 days after heart attack.

For completeness, here's the analysis using `prop.test`. The results are quite similar.

```
> prop.test(x, n, alternative = "less")

##
##  2-sample test for equality of proportions with continuity
##  correction
##
```

```
## data:  x out of n
## X-squared = 0.1519, df = 1, p-value = 0.3484
## alternative hypothesis: less
## 95 percent confidence interval:
##  -1.0000000  0.0198505
## sample estimates:
##   prop 1   prop 2
## 0.158041 0.165129
```

EXAMPLE 11.7. Find a 95% confidence interval for the difference in death rates in Example 11.6.

SOLUTION. To find a two-sided 95% confidence interval for the difference in death rates, use `z.prop.test( )` with the defaults `alternative = "two.sided"` and `conf.level = 0.95` and with `x` and `n` as in Example 11.6.

```
> z.prop.test(x, n)

##
##  2-sample Z-test for equality of proportions
##
## data:  x out of n
## z-statistic = -0.4481, p-value = 0.654
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.0380880  0.0239111
## sample estimates:
##   prop 1   prop 2
## 0.158041 0.165129
```

The confidence interval is $[-0.038, 0.024]$. Since the confidence interval contains 0, there is no significant difference in the two death rates.

The Chi-Square Test for Goodness of Fit

The first chi-square test that we consider is a test of differences between distributions. Its purpose is to compare the observed (empirical) frequencies of a discrete, ordinal, or categorical data set with those of some theoretically expected discrete distribution such as the binomial or Poisson. The hypothesis test takes only one form:

- H_0: The observed frequency distribution is the same as the hypothesized frequency distribution.
- H_a: The observed and hypothesized frequency distributions are different.

The test is carried out with the function

`chisq.test(x = observed, p = expected)`

Details: `x = observed` is the vector containing the observed counts of each category; `p = expected` is the vector specifying the expected proportions for each category. (One can use the expected values instead of the proportions.) The counts and corresponding expected proportions (values) must be listed in the same order. There are additional arguments that may be used with this function that will be detailed later.

The Extrinsic Model

The first example we consider is an *extrinsic* model. In an extrinsic model all population parameters required for the analysis are assumed before the data are collected. No parameters (e.g., the mean or variance) need to be estimated from the data.

EXAMPLE 11.8. Four-o'clocks, *Mirabilis jalapa*, are plants native to tropical America. Their name comes from the fact that their flowers tend to open in the late afternoon. Individual four-o'clock plants can have red, white, or pink flowers. Flower color in this species is thought to be controlled by a single gene locus with two alleles expressing incomplete dominance, so that heterozygotes are pink-flowered, while homozygotes for one allele are white-flowered and homozygotes for the other allele are red-flowered. According to Mendelian genetic principles, self-pollination of pink-flowered plants should produce progeny that have red, pink, and white flowers in a 1:2:1 ratio. A horticulturalist self-pollinates several pink-flowered plants and produces 240 progeny with 55 that are red-flowered, 132 that are pink-flowered, and 53 that are white-flowered. Are these data reasonably consistent with the Mendelian model of a single gene locus with incomplete dominance?

SOLUTION. The hypotheses are

- H_0: The data are consistent with a Mendelian model (red, pink, and white flowers occur in the ratio 1:2:1).
- H_a: The data are inconsistent with a Mendelian model.

The three colors are the categories. In order to calculate expected frequencies, no parameters need to be estimated because the Mendelian ratios (25% red, 50% pink, and 25% white) were established prior to the data collection, making this an *extrinsic* test. To carry out the test, first create the vector `observed` holding the observed counts for the different colors and then create the vector `expected`. Notice below that fractions were used instead of decimal proportions. In general using fractions is more accurate than using decimals that may need to be rounded off, though in this case using `expected <- c(0.25, 0.5, 0.25)` would have produced the same result.

```
> observed <- c(55, 132, 53)
> expected <- c(1/4, 2/4, 1/4)
> chisq.test(x = observed, p = expected)

##
##  Chi-squared test for given probabilities
##
## data:  observed
## X-squared = 2.4333, df = 2, p-value = 0.2962
```

The resulting P value is 0.2962 and is greater than $\alpha = 0.05$. H_0 is accepted. There is support for the Mendelian genetic model.

The Intrinsic Model

The next example we consider is an *intrinsic* model. The intrinsic model requires an estimation of some population parameter(s) from the data collected.

EXAMPLE 11.9. The Poisson distribution is useful for describing rare, random events such as severe storms. To review the characteristics of the Poisson distribution see Chapter 3. In the 98-year period from 1900 to 1997, there were 159 U.S. landfalling hurricanes. Does the number of landfalling hurricanes/year (see the table below) follow a Poisson distribution? (Based on data reported in: Bove, M. et al. 1998. Effect of El Niño on U.S. landfalling hurricanes, revisited. *Bulletin of the American Meteorological Society*, **79**: 2477–2482.)

Hurricanes/year	0	1	2	3	4	5	6
Frequency	18	34	24	16	3	1	2

SOLUTION. The hypotheses are

- H_0: The annual number of U.S. landfalling hurricanes follows a Poisson distribution.
- H_a: The annual number of U.S. landfalling hurricanes is inconsistent with a Poisson distribution.

To carry out an intrinsic chi-square test requires several preliminary calculations. To use the Poisson density formula we need to specify the mean number of hurricanes/year. In this example, this parameter must be estimated from the data, and this is what makes this an *intrinsic* model. From the information given, the estimate for the mean μ is

```
> mu.est <- round(159/98, digits = 3)
> mu.est

## [1] 1.622
```

As in Chapter 3, use the Poisson probability density function `dpois(x, mu)` to determine the probability of `x` hurricanes per year. For example, to determine the probability of 2 hurricanes per year, use

```
> dpois(2, mu.est)

## [1] 0.259804
```

To calculate the probabilities for 0 through 5 hurricanes/year, use the vector `c(0:5)` as `x` in `dpois( )`.

```
> dpois(c(0:5), mu.est)

## [1] 0.1975033 0.3203503 0.2598041 0.1404674 0.0569595 0.0184777
```

The last "observed" category in the given data is 6 hurricanes/year. However, theoretically there is a small probability that more than 6 hurricanes might be observed in a year, so we use "6 or more hurricanes/year" as the last category.

$$P(x \geq 6 \text{ hurricanes/year}) = 1 - P(x < 6 \text{ hurricanes/year}) = 1 - P(x \leq 5 \text{ hurricanes/year})$$

In R use

```
> ppois(5, mu.est, lower.tail = FALSE)    # P(x >= 6)

## [1] 0.00643757
```

Put all of these probabilities into a single variable called `expected.proportions` by using the `c( )` operation.

```
> expected.proportions <- c(dpois(c(0:5), mu.est), ppois(5, mu.est, lower.tail = FALSE))
> expected.proportions

## [1] 0.19750330 0.32035035 0.25980413 0.14046743 0.05695954 0.01847768
## [7] 0.00643757
```

The χ^2 test requires that all expected *values* be at least 5. To determine the expected values, multiply the `expected.proportions` for x hurricanes/year by the number of years, 98.

```
> expected.values <- 98*expected.proportions
> expected.values

## [1] 19.355323 31.394334 25.460805 13.765809  5.582035  1.810812  0.630882
```

The expected frequency of observing 0 hurricanes/year is 19.35, 1 hurricane/year is 31.39, and so on, up to 6 or more hurricanes/year, which is 0.63. The expected frequencies for 5 and for 6 or more hurricanes/year are both less than 5 so do not satisfy the assumptions of the test. To get around this difficulty, adjacent categories should be combined until the expected number is at least 5. In this case, it requires collapsing the last three categories into a single category “≥ 4,” or, in terms of R, `ppois(3, mu.est, lower.tail = FALSE)`.

```
> # the probabilities of 0 to 3 hurricanes and 4 or more hurricanes combined
> expected <- c(dpois(c(0:3), mu.est), ppois(3, mu.est, lower.tail = FALSE))
> expected

## [1] 0.1975033 0.3203503 0.2598041 0.1404674 0.0818748
```

Now we are ready to carry out an intrinsic chi-square test. R does not have a built in function to do this, but we have created the `intrinsic.chisq.test( )` to fill this lacuna.

`intrinsic.chisq.test(x = observed, p = expected, est.params)`

Details: As with `chisq.test( )`, `x = observed` is a vector containing the observed counts of each category and `p = expected` is a vector specifying the expected proportions for each category. The counts and the corresponding expected proportions must be listed in the same order. Finally `est.params` is the number of parameters that were estimated from the data to determine the `expected` proportions. This must be a non-negative integer and the default value is `est.params = 1`. This is used to determine the degrees of freedom in the analysis.

In this example one parameter, μ, was estimated from the data, so `est.params = 1`. From the original data, the observed number of years with no hurricanes was 18, with 1 hurricane was 34, with 2 hurricanes was 24, with 3 hurricanes was 16, and with 4 or more hurricanes was 6. Put these values in a variable called `observed`.

```
> observed <- c(18, 34, 24, 16, 6)
```

The corresponding probabilities were stored in `expected`. So after we download the required function the entire analysis is carried out with a single command.

```
> source("http://waveland.com/Glover-Mitchell/intrinsic.chisq.txt")

## Downloaded: intrinsic.chisq.test( ).

> intrinsic.chisq.test(x = observed, p = expected, est.params = 1)

##
##  Chi-squared test for given probabilities
##
## data:  x = observed and p = expected
## X-squared = 1.268, df = 3, p-value = 0.7367
```

The P value is 0.7367, which is much greater than $\alpha = 0.05$. We cannot reject the claim that the annual number of U.S. landfalling hurricanes is described by a Poisson process with $\mu = 159/98 \approx 1.622$. Accept H_0.

The Chi-Square Test for $r \times k$ Contingency Tables

We now examine a second type of chi-square test that is used to test whether the distribution of a categorical variable is the same in two or more populations.

Contingency Tables

We are interested in how categorical variables are distributed among two or more populations. Let r be the number of categories and k the number of populations or treatments. We can form a table or matrix with r rows, one for each category, and k columns, one for each population. The entry O_{ij} in row i and column j represents the number of observations of category i in population j. Such an arrangement of data is called an $r \times k$ **contingency table**. The question at hand is whether there is a relationship or dependency between the row and column variables. In this context we use the `chisq.test( )` but with different arguments.

`chisq.test(x, correct = TRUE)`

Details: `x` is the contingency table containing the observed counts of each category and the default `correct = TRUE` specifies that a continuity correction is used when calculating the P value for a 2×2 contingency table.

EXAMPLE 11.10. In a study of hermit crab behavior at Point Lookout, North Stradbroke Island, a random sample of three types of gastropod shells was collected. Each shell was then scored as being either occupied by a hermit crab or empty. Shells with living gastropods were not sampled. Do hermit crabs prefer a certain shell type? Or do hermit crabs occupy shells in the same proportions as they occur as empty shells? In other words, is the shell species independent of whether it is occupied?

Species	**Occupied**	**Empty**	**Total**
Austrocochlea	47	42	89
Bembicium	10	41	51
Cirithiidae	125	49	174
Total	182	132	314

SOLUTION. The hypotheses are

- H_0: The status (occupied or not) is independent of the shell species.
- H_a: The status is not independent of the shell species.

The most complicated aspect of the analysis is entering the data. The data must be put into a table (or matrix) before it can be analyzed. Enter the data, *excluding the totals*, row by row (or column by column, see comments below). Then bind the rows together using the `rbind( )` command (or bind the columns with `cbind( )`) to form the data table. Finally, add the column names (or row names).

```
> Austrocochlea <- c(47, 42)      # or Occupied <- c(47, 10, 42)
> Bembicium <- c(10, 41)          # and Empty <- c(42, 41, 49)
> Cirithiidae <- c(125, 49)
> Example11.10.table <- rbind(Austrocochlea, Bembicium, Cirithiidae)
> # or Example11.10.table <- cbind(Occupied, Empty)
> colnames(Example11.10.table) <- c("Occupied", "Empty")
> # rownames(Example11.10.table) <- c("Austrocochlea", "Bembicium", "Cirithiidae")
> Example11.10.table

##               Occupied Empty
## Austrocochlea       47    42
## Bembicium           10    41
## Cirithiidae        125    49
```

The analysis is now carried out using

```
> chisq.test(Example11.10.table)

##
##  Pearson's Chi-squared test
##
## data:  Example11.10.table
## X-squared = 45.5116, df = 2, p-value = 1.31e-10
```

The tiny P value indicates that H_0 should be rejected. There is reason to believe that shell species and occupancy are not independent. That is, hermit crabs are "selective" in the species of shell that they occupy.

A couple of other features of R are useful to mention at this point. To add the marginal totals to `Example11.10.table` use

```
> addmargins(Example11.10.table)

##               Occupied Empty Sum
## Austrocochlea       47    42  89
## Bembicium           10    41  51
## Cirithiidae        125    49 174
## Sum                182   132 314
```

To see the expected values used in the test, use

```
> chisq.test(Example11.10.table)$expected

##               Occupied   Empty
## Austrocochlea  51.5860 37.4140
## Bembicium      29.5605 21.4395
## Cirithiidae   100.8535 73.1465
```

EXAMPLE 11.11. A study was conducted in the reef flat of Lady Elliot Island to examine the distribution of the animals associated with different species of coral. Three species of coral were selected and appropriate coral heads were sampled. The number of snails of species A and B associated with each were recorded. Is the distribution of snail species the same for all three coral species? Use the data below to test the hypothesis at the $\alpha = 0.05$ level. Clearly state the hypotheses and interpret the results.

	Coral			
	Pocillopora eydouxi	***Acropora* sp.**	***Acropora aspera***	**Total**
Species A	6	2	14	22
Species B	7	21	1	29
Total	13	23	15	51

SOLUTION. The hypotheses are

- H_0: The distribution of snail species A and B is the same in all three types of coral heads.
- H_a: The distribution of snail species A and B is not the same in all three types of coral heads.

As in the previous example, form the data table a row at a time and carry out the test.

```
> SpeciesA <- c(6, 2, 14)
> SpeciesB <- c(7, 21, 1)
> Example11.11.table <- rbind(SpeciesA, SpeciesB)
> colnames(Example11.11.table) <- c("Pocillopora eydouxi", "Acropora sp",
+ "Acropora aspera")
> Example11.11.table

##          Pocillopora eydouxi Acropora sp Acropora aspera
## SpeciesA                   6           2              14
## SpeciesB                   7          21               1

> chisq.test(Example11.11.table)$expected # check all expected values >= 5

##          Pocillopora eydouxi Acropora sp Acropora aspera
## SpeciesA             5.60784     9.92157         6.47059
## SpeciesB             7.39216    13.07843         8.52941

> chisq.test(Example11.11.table)

##
##  Pearson's Chi-squared test
##
## data:  Example11.11.table
## X-squared = 26.5792, df = 2, p-value = 1.692e-06
```

The small *P* value indicates that null hypothesis should be rejected. Based on these data, the distribution of snail species A and B varies with coral species.

Finally, observe that the expected values were also printed. It is worth recalling the rule of thumb that each *expected* frequency should be at least 5. Note that Example 11.11 satisfies this criterion even though some of the *observed* frequencies fall below 5. The `chisq.test( )` will provide a warning whenever an expected value is smaller than 5.

2 × 2 *Contingency Tables*

A special case of the $r \times k$ contingency table is the 2×2 table. Because of the small number of cells (4), a correction factor is usually employed.

EXAMPLE 11.12. A study of kidney damage during organ retrieval for transplantation was conducted in the United Kingdom using data from the UK National Transplant Database for the 1992 to 1996 period. In many cases of organ donation, when the kidneys are retrieved the liver is retrieved as well in a single surgical procedure. When both types of organs were retrieved, the researchers categorized the surgical team, based on the operating surgeon's specialty, as either a renal retrieval team or a liver retrieval team. Their data are given below. Was the rate of reported damage to kidneys independent of the type of surgical team? Analyze the data with a 2×2 contingency table. *Note:* 94% of damaged organs were still transplanted. (Based on data from: Wigmore, S. et al. 1999. Kidney damage during organ retrieval: Data from the UK National Transplant Database. *The Lancet*, **354**: 1143–1146.)

Team	**Damaged kidneys**	**Undamaged kidneys**	**Total**
Renal retrieval	454	1692	2146
Liver retrieval	415	2054	2469
Total	869	3746	4615

SOLUTION. The hypotheses are

- H_0: Surgical team and condition of kidneys are independent.
- H_a: An association between type of surgical team and condition of kidneys exists (one team experiences more success than the other in these circumstances).

As in the previous two examples, create the data table and carry out the test. Here we illustrate constructing the table column by column.

```
> Damaged <- c(454, 415)
> Undamaged <- c(1692, 2054)
> Example11.12.table <- cbind(Damaged, Undamaged) # use cbind( ) for columns
> rownames(Example11.12.table) <- c("Renal retrieval", "Liver retrieval")
> addmargins(Example11.12.table)          # optional: print the margin totals

##                 Damaged Undamaged  Sum
## Renal retrieval     454      1692 2146
## Liver retrieval     415      2054 2469
## Sum                 869      3746 4615

> chisq.test(Example11.12.table)$expected # check all expected values >= 5

##                 Damaged Undamaged
## Renal retrieval  404.09   1741.91
## Liver retrieval  464.91   2004.09

> chisq.test(Example11.12.table)

##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  Example11.12.table
## X-squared = 13.9127, df = 1, p-value = 0.0001915
```

Note: With 2 × 2 contingency tables, a continuity correction is automatically applied. The P value associated with a test statistic of 13.91 is less than 0.0002; reject H_0. The kidney damage rate for renal surgical teams (21.2%) is significantly higher than the kidney damage rate for liver surgical teams (16.8%), a rather unexpected result.

Partitioning the Chi-Square Test

The final example in this section demonstrates the use of various chi-square analyses to pinpoint discrepancies from expectation.

EXAMPLE 11.13. A genetics student wished to repeat one of Gregor Mendel's classic experiments with garden peas, *Pisum sativum*. She decided to study two characteristics: stem length and seed pod color. From her research she knows that a single gene locus with two alleles controls stem length. AA or Aa produces tall plants (about 1 m) while aa produces short plants (approximately 0.5 m). Also a single locus with two alleles controls seed pod color with BB and Bb producing green seed pods and bb producing yellow seed pods. In other words, both loci exhibit complete dominance. From Mendel's published work these two gene loci are assumed to be independently assorting. The student crosses together plants that are tall with green seed pods and are known to be heterozygous at both loci:

tall, green pods (AaBb) × tall, green pods (AaBb)

↓

Experimentally produced offspring:

178 tall, green pods (A_B_)
30 tall, yellow pods (A_bb)
62 short, green pods (aaB_)
10 short, yellow pods (aabb)

If these genes behave according to Mendel's laws, she expects the offspring to be in a 9:3:3:1 ratio. Test this hypothesis.

SOLUTION. The hypotheses are

- H_0: The results are in a 9:3:3:1 phenotypic ratio.

- H_a: The results deviate significantly from a 9:3:3:1 ratio.

This requires an extrinsic chi-square test; the expected proportions, namely $\frac{9}{16}$, $\frac{3}{16}$, $\frac{3}{16}$, and $\frac{1}{16}$, do not require any parameters estimated from the data.

```
> observed <- c(178, 30, 62, 10)
> expected <- c(9/16, 3/16, 3/16, 1/16)
> chisq.test(x = observed, p = expected)

##
##  Chi-squared test for given probabilities
##
## data:  observed
## X-squared = 17.2444, df = 3, p-value = 0.0006295
```

Since $\chi^2 = 17.24$ has a P value less than 0.001, reject H_0. The data deviate significantly from a 9:3:3:1 ratio.

The student would like to know exactly why the data set failed to meet the expectation that she had so carefully researched. Thinking about this situation, she realized that the 9:3:3:1 ratio was predicated on three assumptions.

1. The gene locus for plant height produced offspring in a 3 A$_-$:1 aa ratio.
2. The gene locus for seed pod color produced offspring in a 3 B$_-$:1 bb ratio.
3. The 2 gene loci, A and B, are independent of each other.

Test each of these assumptions separately and independently by the following methods:

1. First test

   - H_0: Offspring have a 3 tall :1 short ratio.
   - H_a: Offspring deviate significantly from the expected ratio of 3 tall:1 short.

   This requires another extrinsic chi-square test. A 3:1 ratio uses proportions $\frac{3}{4}$ and $\frac{1}{4}$.

   ```
   > observedTs <- c(208, 72)                # Ts = Tall short
   > expectedTs <- c(3/4, 1/4)
   > chisq.test(x = observedTs, p = expectedTs)

   ##
   ##  Chi-squared test for given probabilities
   ##
   ## data:  observedTs
   ## X-squared = 0.0762, df = 1, p-value = 0.7825
   ```

   Because the P value of the test is much larger than α, $(0.7825 \gg 0.05)$ H_0 is accepted. Plant heights in offspring are in a 3:1 ratio.

2. Next test

   - H_0: Offspring have a 3 green seed pod:1 yellow seed pod ratio.
   - H_a: Offspring deviate significantly from the expected ratio of 3 green:1 yellow.

   The set up is similar to the Tall-short analysis.

   ```
   > observedGy <- c(240, 40)                # Gy = Green yellow
   > expectedGy <- c(3/4, 1/4)
   > chisq.test(x = observedGy, p = expectedGy)

   ##
   ##  Chi-squared test for given probabilities
   ##
   ## data:  observedGy
   ## X-squared = 17.1429, df = 1, p-value = 0.00003467
   ```

This time the P value of the test is much smaller than $\alpha = 0.05$, so H_0 is rejected. The phenotypic ratio for the seed pod color is significantly different from 3:1.

3. Finally test

- H_0: Gene locus A is independent of gene locus B.
- H_a: The loci are not independent.

We test this H_0 with a 2×2 contingency table:

```
> A_ <- c(178, 30)
> aa <- c(62, 10)
> Example11.13.table <- rbind(A_, aa)
> colnames(Example11.13.table) <- c("B_ ", "bb")
> addmargins(Example11.13.table) # print the margin totals, too

##      B_  bb Sum
## A_   178 30 208
## aa    62 10  72
## Sum 240 40 280

> chisq.test(Example11.13.table)$expected # check all expected values >= 5

##         B_       bb
## A_ 178.2857 29.7143
## aa  61.7143 10.2857

> chisq.test(Example11.13.table)

##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  Example11.13.table
## X-squared = 0, df = 1, p-value = 1
```

The calculated value of χ^2 is nearly 0 and the corresponding P value is effectively 1. Accept H_0 that the two loci are behaving independently.

Looking at the three single degree of freedom chi-square analyses, one can see that the discrepancy in the overall χ^2 testing 9:3:3:1 is due to a distortion in the green to yellow seed pod ratio. Notice the distorted ratio for the B locus has no effect on the 2×2 contingency test. Also the single degree of freedom chi-squares are a decomposition of the overall chi-square and sum to that chi-square value with some small rounding differences. The analysis done here is analogous to the partitioning of the sums of squares in analysis of variance. This type of decomposition can be used on chi-square analyses as long as each test is independent of the other tests involved in the decomposition.

The Kolmogorov-Smirnov Test

As with the previous goodness of fit tests, the purpose of this test is to determine whether a random sample from some unknown distribution is, in fact, actually from some specified distribution. While the χ^2 test is specifically designed for use with discrete or categorical data, the Kolmogorov-Smirnov test is used to test whether a random sample comes from a population with a particular *continuous* distribution. The Kolmogorov-Smirnov test is carried out in R by using the `ks.test( )` function.

```
ks.test(x, y, ..., alternative = "two.sided")
```

Details: `x` is a numeric vector of data values; `y` specifies the distribution in question and the ... indicate any parameters required to specify the distribution (e.g., mean and standard deviation); `alternative` indicates the alternative hypothesis where the default is `"two.sided"` and the other options are `"greater"` and `"less"`.

EXAMPLE 11.14. The heights of 18- to 22-year-old females in the U.S. are normally distributed with $\mu = 163$ cm and $\sigma = 8$ cm. As part of a data analysis exercise on the Kolmogorov-Smirnov test, 10 female students were selected at random in one of our statistics classes and their heights measured. Do the data below follow the same distribution as the national data or are the heights of women at our college distributed differently?

149 157 159 160 160 163 168 170 178 185

SOLUTION. The hypotheses are:

- H_0: $S(x) = F(x)$ for all x. The empirical (actual) cumulative distribution $S(x)$ and hypothesized cumulative distribution $F(x)$ are identical.
- H_a: $S(x) \neq F(x)$ for at least one value of x. The actual and hypothesized cumulative distributions differ significantly for at least one point.

To carry out the test create a vector containing the observed heights. The question is whether this sample comes from a population in which heights are normally distributed with a mean of 163 cm and a standard deviation of 8 cm. This distribution is specified in R using `pnorm, mean = 163, sd = 8`. The data may be downloaded from `http://waveland.com/Glover-Mitchell/Example11-14.txt`. Thus, to carry out the test use

```
> data.Ex11.14 <- read.table("http://waveland.com/Glover-Mitchell/Example11-14.txt",
+ header = TRUE)
> head(data.Ex11.14, n = 2)        # list data, determine variable name

##   Height
## 1    149
## 2    157

> ks.test(data.Ex11.14$Height, y = pnorm, mean = 163, sd = 8)

## Warning in ks.test(data.Ex11.14$Height, y = pnorm, mean = 163, sd = 8):  ties
should not be present for the Kolmogorov-Smirnov test

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  data.Ex11.14$Height
## D = 0.1696, p-value = 0.9359
## alternative hypothesis: two-sided
```

The presence of ties always generates a warning, since ties are not expected to occur in continuous distributions. Perhaps the heights should have been measured to the nearest millimeter. Note: If the data were not provided, you could easily create your own data set and do the analysis as follows.

```
> Height <- c(149, 157, 159, 160, 160, 163, 168, 170, 178, 185)
> ks.test(Height, pnorm, mean = 163, sd = 8)

## Warning in ks.test(Height, pnorm, mean = 163, sd = 8):  ties should not be
present for the Kolmogorov-Smirnov test

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Height
## D = 0.1696, p-value = 0.9359
## alternative hypothesis: two-sided
```

The very large P value indicates that H_0 is cannot be rejected. In further analysis it would be appropriate to assume that female student heights in our class are from a normally distributed population with a mean of 163 cm and a standard deviation of 8 cm.

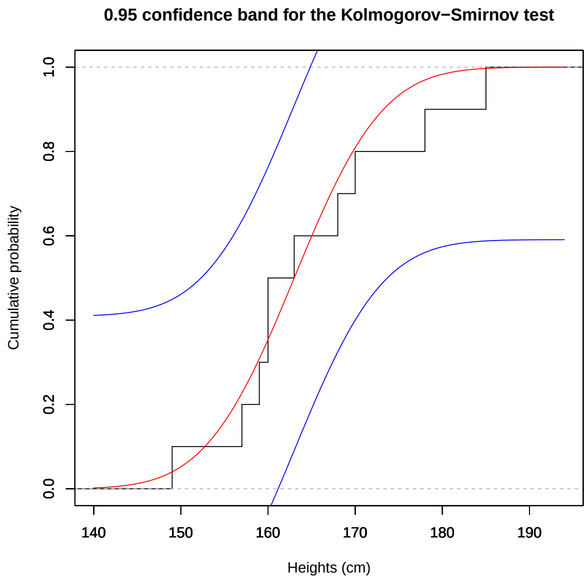
Here's a function we've supplied that illustrates what the Kolmogorov-Smirnov test is measuring.

ks.plot(x, xlab = "x-label", y, ...)

Details: `x` is a numeric vector of data values; `xlab = "x-label"` specifies the label for the x-axis that describes the data; and `y` specifies the distribution in question and the `...` indicate any parameters required to specify the distribution (e.g., mean and standard deviation).

To download this function use the `source( )` command as follows.

```
> source("http://waveland.com/Glover-Mitchell/ks-lillie.plot.txt")        # download source file

## Downloaded: ks.plot( )and lillie.plot( ).

> ks.plot(data.Ex11.14$Height, xlab = "Heights (cm)", y = pnorm, mean = 163, sd = 8)
```



The step function in the plot shows the empirical cumulative distribution (the data). The central red S-curve is the hypothesized distribution: the cumulative normal distribution with $\mu = 163$ and $s = 8$. The Kolmogorov-Smirnov test statistic is the maximum vertical distance between the two curves. The smaller this distance, the closer the curves are and the more likely that the sample comes from a normal distribution with the specified mean and standard deviation. The two blue boundary S-curves form the 0.95 confidence band about about the hypothesized distribution. Only if the empirical cumulative distribution (the step function) remains within this band is the null hypothesis retained: The empirical (actual) cumulative distribution $S(x)$ and hypothesized cumulative distribution $F(x)$ are identical.

EXAMPLE 11.15. We have suggested that a binomial distribution in which $np > 5$ and $n(1 - p) > 5$ is well approximated by a normal distribution with $\mu = np$ and $\sigma^2 = np(1 - p)$. We used R to simulate 20 independent experiments each using a binomial random variable with $n = 1000$ trials and $p = 0.5$. The successes in each of the trials are listed below.

480	492	505	483	513	485	474	524	505	509
484	514	498	493	507	494	487	508	499	501

Does a Kolmogorov-Smirnov test indicate that these data could reasonably have come from the normal distribution $F(x)$ with $\mu = np = 1000(0.5) = 500$ and $\sigma^2 = np(1 - p) = 1000(0.5)(0.5) = 250$?

SOLUTION. Notice that even though the sample comes from a *discrete* binomial distribution, we are asking whether we may act as if the sample were drawn from a *continuous* normal distribution. Thus, a Kolmogorov-Smirnov test is appropriate. The hypotheses are H_0: $S(x) = F(x)$ for all x and H_a: $S(x) \neq F(x)$ for at least one value of x. We will carry out the test with $\alpha = 0.05$. The question is whether this sample comes from a population in which heights are normally distributed with a mean of 500 and a standard deviation of $\sqrt{250}$. The data may be found at `http://waveland.com/Glover-Mitchell/Example11-15.txt`.

```
> data.Ex11.15 <- read.table("http://waveland.com/Glover-Mitchell/Example11-15.txt",
+ header = TRUE)
> head(data.Ex11.15, n = 3)        # list data, determine variable name

##   Successes
## 1       480
## 2       492
## 3       506

> ks.test(data.Ex11.15$Successes, y = pnorm, mean = 500, sd = sqrt(250))

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  data.Ex11.15$Successes
## D = 0.138, p-value = 0.7924
## alternative hypothesis: two-sided
```

The P value is 0.79 which exceeds $\alpha = 0.05$. Consequently accept H_0. We conclude that we can reasonably act as if the data came from a normal distribution with $\mu = 500$ and $\sigma^2 = 250$, as was suggested.

EXAMPLE 11.16. The cerulean warbler, *Setophaga cerulea*, is a small songbird most often seen in forest treetops. It nests and forages higher in the canopy than most other warblers. Their breeding habitat is confined to the eastern North American deciduous forests. It winters in the boreal forests of South America. It is under consideration for listing under the Endangered Species Act. As part of a study of the warbler's breeding behavior, the nest heights of 20 pairs of cerulean warblers were measured (in feet). Could this sample of nest heights have been drawn from a uniformly distributed population with a range of 10 to 60 feet?

14.8 17.1 24.9 26.2 26.2 29.2 39.4 42.7 45.9 55.8

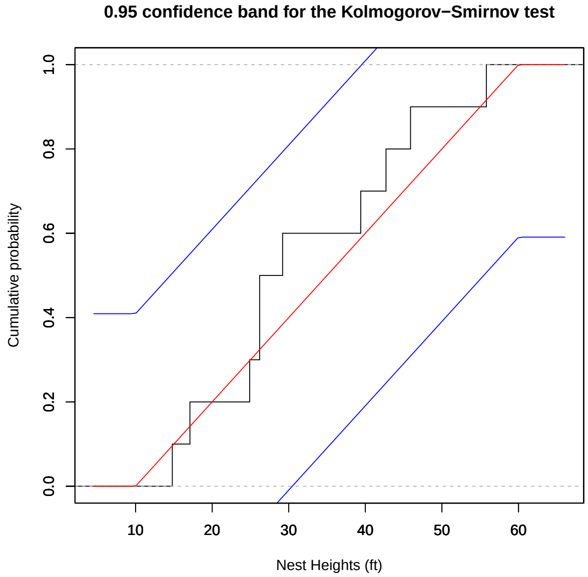
SOLUTION. In the uniform distribution on $[10, 60]$, every number is equally likely to occur. Thus, it is completely specified by the minimum and maximum values of the range, namely, 10 and 60. In R this is specified as: `punif, min = 10, max = 60`. The data can be read from `http://waveland.com/Glover-Mitchell/Example11-16.txt`.

```
> data.Ex11.16 <- read.table("http://waveland.com/Glover-Mitchell/Example11-16.txt",
+ header = TRUE)
> head(data.Ex11.16, n = 2)        # list data, determine variable name

##   NestHt
## 1   14.8
## 2   17.1

> ks.test(data.Ex11.16$NestHt, y = punif, min = 10, max = 60)

## Warning in ks.test(data.Ex11.16$NestHt, y = punif, min = 10, max = 60):  ties
should not be present for the Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  data.Ex11.16$NestHt
## D = 0.216, p-value = 0.7392
## alternative hypothesis: two-sided
```

The P value is 0.7392 and is greater than $\alpha = 0.05$; accept H_0. We may assume that the nest heights are uniformly distributed on the interval $[10, 60]$. Here's the plot of the what the test measures. The uniform distribution is the red line. Note that the empirical cumulative distribution (black step function) stays within the 0.95 confidence band about the hypothesized distribution.

```
> ks.plot(data.Ex11.16$NestHt, xlab = "Nest Heights (ft)", y = punif, min = 10, max = 60)
```



PROBLEM 11.0 (Additional Exercise). Many states now have state lotteries. Typical of these games is the New Jersey Pick-4 lottery. Participants may choose any of the ten-thousand four-digit numbers from 0000 to 9999. Winning numbers are chosen twice a day, at midday and in the evening. Assuming that the lottery is fair, each number has an equal probability $p = 0.0001$ of being selected. This means that the cumulative distribution of winning numbers should be approximately the continuous uniform distribution. The data in `http://waveland.com/Glover-Mitchell/Problem11-NJLottery.txt` are the 365 winning numbers from from midday on January 1 to midday on July 1, 2004.

(*a*) Why did we say "the cumulative distribution of winning numbers should be approximately the continuous uniform distribution."

(*b*) Carefully state the null and alternative hypotheses for a Kolmogorov-Smirnov test to determine whether the lottery is fair.

(*c*) What is the outcome of the test?

(*d*) Create a graphical plot of the test.

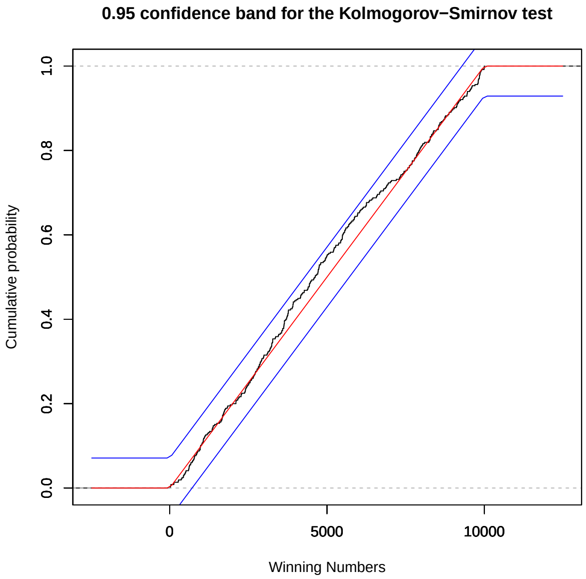
SOLUTION. We said "approximate" because the true distribution of the winning numbers is actually discrete. The winning numbers can only be integers. *But* because there are so many categories (10000 possible numbers) the distribution is effectively continuous.

The hypotheses are:

- H_0: The distribution of winning numbers is uniform.
- H_a: The distribution of winning numbers is not uniform.

Read the data to determine the structure of the file and see whether all data are present.

```
> data.Prob.NJ <- read.table("http://waveland.com/Glover-Mitchell/Problem11-NJLottery.txt",
+ header = TRUE)
> head(data.Prob.NJ)       # list data, determine variable name

##   Pick4
## 1  3917
## 2  7350
## 3  6224
## 4  2904
## 5  3276
## 6  2735

> tail(data.Prob.NJ)

##     Pick4
## 360  3514
## 361  5455
## 362  7771
## 363   435
## 364  9582
## 365   790

> ks.test(data.Prob.NJ$Pick4, y = punif, min = 0, max = 9999)

## Warning in ks.test(data.Prob.NJ$Pick4, y = punif, min = 0, max = 9999):  ties
should not be present for the Kolmogorov-Smirnov test

##
##  One-sample Kolmogorov-Smirnov test
##
## data:  data.Prob.NJ$Pick4
## D = 0.0564, p-value = 0.1962
## alternative hypothesis: two-sided

> ks.plot(data.Prob.NJ$Pick4, xlab = "Winning Numbers", y = punif, min = 0, max = 9999)
```



Notice how as the sample size has increased, the cv-corridor has narrowed compared to the previous example (the vertical scales in both diagrams are the same). The empirical distribution remains within the corridor and the P value of the test is 0.1962, which is greater than $\alpha = 0.05$. We cannot reject the null hypothesis.

The Lilliefors Test

We finish our discussion of goodness of fit tests with a description of the Lilliefors test, which can be viewed as a variant of the Kolmogorov-Smirnov test. Its purpose is to determine whether a sample could have come from a population with a normal distribution *without having specified the mean and variance in advance*. This distinguishes it from the Kolmogorov-Smirnov test.

`lillie.test(x)`

Details: `x` is a numeric vector of data values; the test is two-sided.

EXAMPLE 11.17. In Problem 2 of Chapter 6 we listed data that were part of a benthic community survey of Lady Elliot Island. Sixteen sea stars, *Linckia laevigata*, were collected and their longest arms were measured to the nearest tenth of a centimeter.

10.3	11.0	10.5	10.0	11.3	14.5	13.0	12.1
12.1	9.4	11.3	12.0	11.5	9.3	10.1	7.6

The problem said to assume normality for the arm lengths. Is this assumption warranted?

SOLUTION. Let $\alpha = 0.05$ be the significance level of the test. The data may be found at `http://waveland.com/Glover-Mitchell/Example11-17.txt`. The hypotheses are:

- H_0: The distribution of *Linckia laevigata* is normal.
- H_a: The distribution of *Linckia laevigata* is not normal.

```
> data.Ex11.17 <- read.table("http://waveland.com/Glover-Mitchell/Example11-17.txt",
+ header = TRUE)
> data.Ex11.17    # list data, determine variable name

##    Length
## 1    10.3
## 2    11.0
## 3    10.5
## 4    10.0
## 5    11.3
## 6    14.5
## 7    13.0
## 8    12.1
## 9    12.1
## 10    9.4
## 11   11.3
## 12   12.0
## 13   11.5
## 14    9.3
## 15   10.1
## 16    7.6

> lillie.test(data.Ex11.17$Length)

## Error in eval(expr, envir, enclos):  could not find function "lillie.test"
```

What happened? Typically the Lilliefors test is not included in the basic packages that are automatically installed in R. You must download the R package called `nortest` that contains the code for `lillie.test` and other functions. Your instructor or TA can help you do this. If you are working on a university network, you may not have permission to do so, so your instructor may have to download it for you (or may already have done so). On your personal computer, you can try using the command `install.packages("nortest")`. Depending on the operating system you are using, when in R you may have a menu item called `Packages & Data` that can be used.

Once the `nortest` package is installed on your computer (or network), you must still load the package for each R session with the command `library(nortest)`, which will make the package available to use. Let's try it.

```
> library(nortest)
> lillie.test(data.Ex11.17$Length)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  data.Ex11.17$Length
## D = 0.1244, p-value = 0.7307
```

The P value is 0.7307 and is greater than $\alpha = 0.05$; accept H_0. Continue to assume the arm lengths of *Linckia laevigata* are normally distributed.

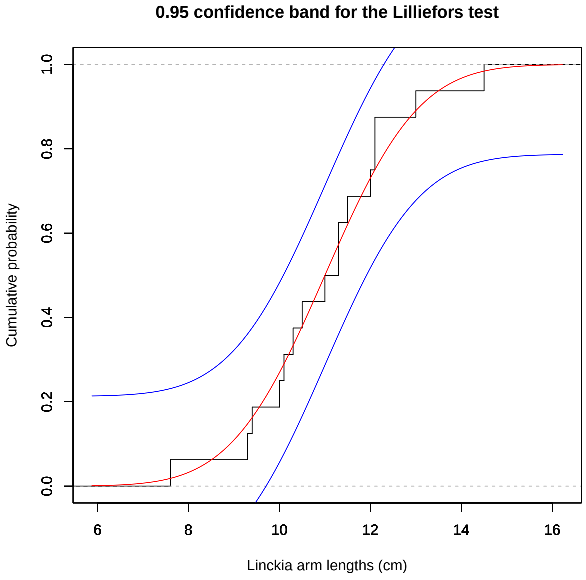
We can illustrate what the Lilliefors test is measuring with

`lillie.plot(x, xlab = "x-label")`

Details: `x` is a numeric vector of data values and `xlab = "x-label"` specifies the label for the x-axis that describes the data.

This function is downloaded with `ks.plot`. If you have not already downloaded it, use the `source( )` command as follows.

```
> source("http://waveland.com/Glover-Mitchell/ks-lillie.plot.txt")        # download source file

## Downloaded: ks.plot( )and lillie.plot( ).

> lillie.plot(data.Ex11.17$Length, xlab = "Linckia arm lengths (cm)")
```



The step function in the plot shows the empirical cumulative distribution. The central red S-curve that is plotted is the hypothesized distribution, the cumulative normal distribution with sample mean $\overline{X}$ and sample standard deviation s. The Lilliefors test statistic is the maximum vertical distance between the two curves. The smaller this number, the closer the curves are and the more likely that the sample comes from a normal distribution with the sample mean and standard deviation. The two blue boundary S-curves form the 0.95 confidence band about about the hypothesized distribution. Only if the empirical cumulative distribution (the step function) remains within this corridor is the null hypothesis retained: The empirical (actual) cumulative distribution $S(x)$ and hypothesized cumulative distribution $F(x)$ are identical.

Two Additional Examples

EXAMPLE 11.18. The table below provides annual rainfall data (in mm) for 118 consecutive years at Cowarie Station, South Australia from 1882 to 1999. (Cowarie is a 4,000 square kilometer cattle station in north east South Australia.) The data have been rearranged in increasing order and are located at `http://waveland.com/Glover-Mitchell/Example11-18.txt`.

13	34	48	74	87	108	124	145	185	264
15	35	50	75	91	108	126	146	191	268
16	35	52	76	92	109	129	147	195	283
18	37	54	77	93	112	130	152	204	290
19	38	56	78	93	113	130	157	206	294
20	38	57	78	95	116	132	159	212	301
23	39	57	78	96	116	132	159	217	324
24	41	60	79	96	117	133	167	219	340
27	42	62	80	99	121	137	170	228	465
32	43	63	80	102	121	139	172	229	484
34	44	71	81	104	121	141	173	230	
34	44	73	85	107	122	143	183	248	

(*a*) Using an appropriate test, determine whether rainfall is normally distributed in this part of Australia. Illustrate the result with an appropriate graph.

(*b*) Carryout a similar analysis for the annual rainfall data for Brisbane, Queensland (on the east coast of Australia) mentioned in Chapter 0 of this guide and located at `http://waveland.com/Glover-Mitchell/Example00-2.txt`.

(*c*) Make side-by-side box plots of the Cowarie Station and Brisbane data.

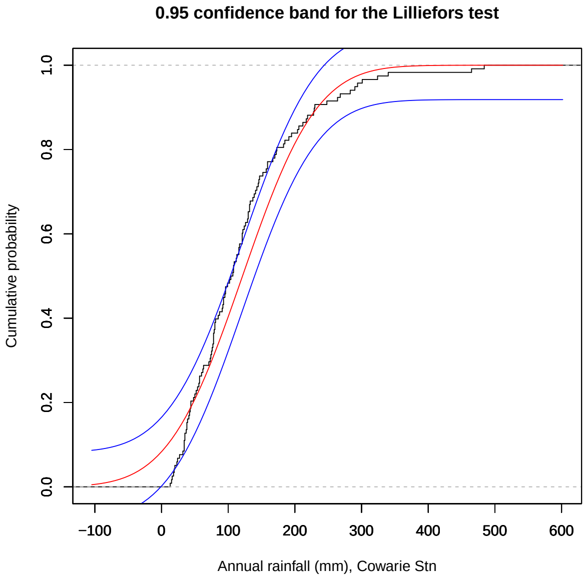
SOLUTION. Use `lillie.test( )` and `lillie.plot( )` for both analyses. The hypotheses for each test are:

- H_0: The distribution of rainfall is normal.
- H_a: The distribution of rainfall is not normal.

Begin with the Cowarie Station analysis.

```
> data.Ex11.Cowarie <- read.table("http://waveland.com/Glover-Mitchell/Example11-Cowarie.txt",
+ header = TRUE)
> head(data.Ex11.Cowarie, n = 3)

##   Rainfall
## 1       13
## 2       34
## 3       48

> tail(data.Ex11.Cowarie, n = 3)

##     Rainfall
## 116      143
## 117      183
## 118      248

> lillie.test(data.Ex11.Cowarie$Rainfall)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  data.Ex11.Cowarie$Rainfall
## D = 0.1256, p-value = 0.00009602

> lillie.plot(data.Ex11.Cowarie$Rainfall, xlab = "Annual rainfall (mm), Cowarie Stn")
```



The tiny P value (about 0.0001) indicates that the null hypothesis should be rejected. Rainfall does not follow a normal distribution at Cowarie Station. Equivalently, the graph of the rainfall data at Cowarie Station does not remain in the 0.95 confidence band.

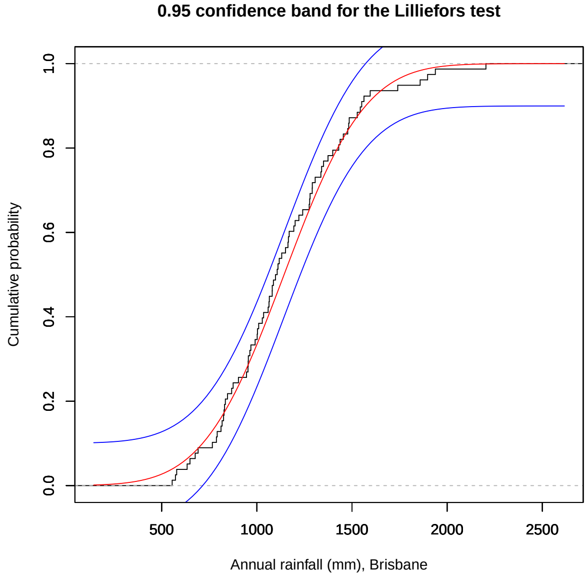
The Brisbane data are analyzed in a similar fashion. First read in the data from `http://waveland.com/Glover-Mitchell/Example00-2.txt` and print a few rows.

```
> data.Ex00.2 <- read.table("http://waveland.com/Glover-Mitchell/Example00-2.txt",
+ header = TRUE)
> head(data.Ex00.2, n = 4)        # list some data

##   Year   Jan   Feb   Mar   Apr  May   Jun   Jul  Aug  Sep  Oct   Nov   Dec
## 1 1921 129.9  16.9 194.7 203.0 19.8 197.9 167.7  4.3 44.8 28.4  57.2 226.4
## 2 1922  67.0 230.1  26.9   8.4 54.0  39.6 118.1  3.4 66.6 37.2  51.9 114.4
## 3 1923  46.7  21.4  87.1 192.4  9.1  73.9  64.5 24.4 36.1 10.9  47.1  62.5
## 4 1924  79.6 174.0 105.2  75.0 42.7 113.3 162.7 33.8 38.8 47.4 169.5  45.0
##   Annual
## 1 1291.0
## 2  817.6
## 3  676.1
## 4 1087.0
```

Each row represents a year with rainfall totals for each of the twelve months followed by the annual rainfall. The annual rainfall data resides in the column `data.Ex00.2$Annual`. Carry out the Lilliefors test on the `Annual` data and plot the confidence band.

```
> head(data.Ex00.2$Annual)        # list some data

## [1] 1291.0  817.6  676.1 1087.0 1373.9  787.5

> tail(data.Ex00.2$Annual)

## [1]  633.7  811.7 1111.1 1485.4  903.2  955.3
```

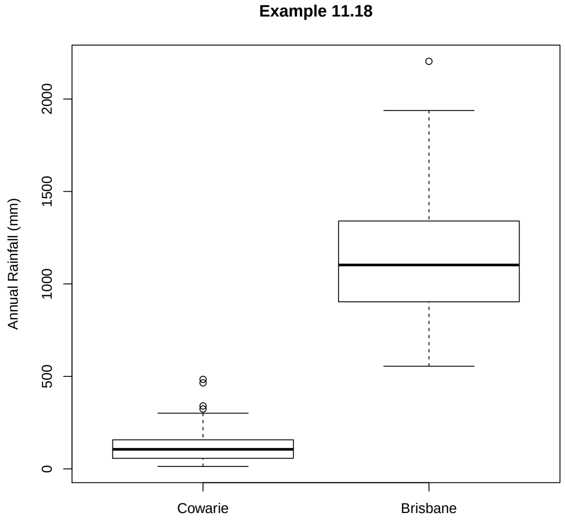
```
> lillie.test(data.Ex00.2$Annual)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  data.Ex00.2$Annual
## D = 0.0706, p-value = 0.4393

> lillie.plot(data.Ex00.2$Annual, xlab = "Annual rainfall (mm), Brisbane")
```



The P value for the test is 0.4393, so the rainfall in Brisbane appears to follow a normal distribution. The graph of the rainfall remains within the 0.95 confidence band.

Next create the box plots.

```
> boxplot(data.Ex11.Cowarie$Rainfall, data.Ex00.2$Annual, main = "Example 11.18",
+ names = c("Cowarie", "Brisbane"), ylab = "Annual Rainfall (mm)")
```



EXAMPLE 11.19. The data set `http://waveland.com/Glover-Mitchell/Problem01-OldFaithful.txt` lists the time interval (in minutes) between eruptions of Old Faithful for the entire

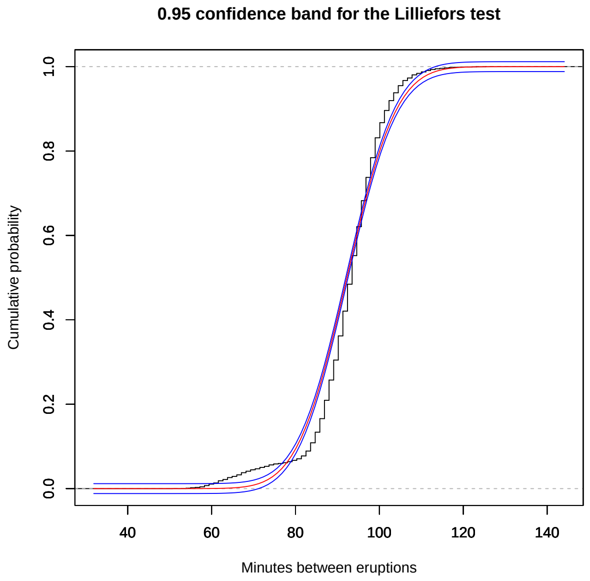
year of 2010. In Problem 1 of Chapter 1 of this guide, you created a histogram of these data. Now determine whether the eruption intervals follow a normal distribution. Create a graphical plot of the analysis.

SOLUTION. Use `lillie.test( )` and `lillie.plot( )` for the analysis and plot. The hypotheses for the test are:

- H_0: The distribution of eruption times is normal.

- H_a: The distribution of eruption times is not normal.

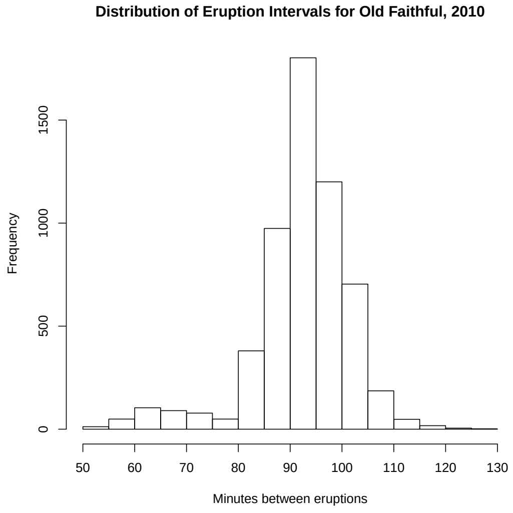
```
> data.Ex11.OF <- read.table("http://waveland.com/Glover-Mitchell/Problem01-OldFaithful.txt",
+ header = TRUE)
> head(data.Ex11.OF)      # list data

##   Interval
## 1     93.5
## 2     67.1
## 3    102.3
## 4     93.5
## 5     92.4
## 6     95.7

> tail(data.Ex11.OF)

##      Interval
## 5694     92.4
## 5695     94.6
## 5696     90.2
## 5697     88.0
## 5698     95.7
## 5699     99.0

> lillie.test(data.Ex11.OF$Interval)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  data.Ex11.OF$Interval
## D = 0.1165, p-value < 2.2e-16

> lillie.plot(data.Ex11.OF$Interval, xlab = "Minutes between eruptions")
```



The tiny P value for the test indicates that the time intervals between eruptions of Old Faithful are not distributed normally. There are a couple of things to notice in the plot. First, the confidence band is quite narrow because of the large sample size ($n = 5699$). Second, notice how the empirical distribution first rises above the confidence band starting at about 65 minutes, then dips below the band at about 80 minutes and exceeds the band again at about 95 minutes.

A histogram of the eruption interval data shows that it is bimodal. There is a small mode at about 65 minutes and a much larger mode at about 95 minutes.

```
> hist(data.Ex11.OF$Interval, breaks = seq(50, 130, by = 5),
+ main = "Distribution of Eruption Intervals for Old Faithful, 2010",
+ xlab = "Minutes between eruptions", xaxt = 'n')
> axis(side = 1, at = seq(50, 130, by = 10), labels = seq(50, 130, by = 10))
```

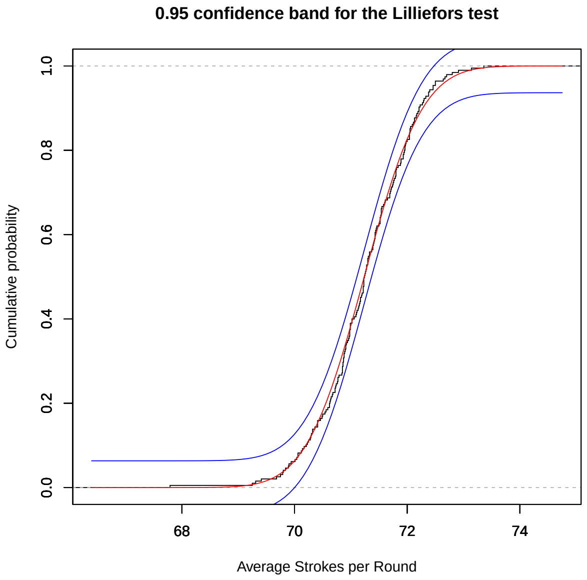


EXAMPLE 11.20. The file `http://waveland.com/Glover-Mitchell/Example11-PGA.txt` contains the scoring averages for the top 195 players on PGA Tour for the year 2000 (source: `www.pgatour.com/stats/2000/r_120.html`). Determine with an appropriate test whether these averages are described by a normal distribution. Create a graphical plot of the results of this test.

SOLUTION. Use `lillie.test( )` and `lillie.plot( )` for the analysis and plot. The hypotheses for the test are:

- H_0: The distribution of scoring averages is normal.
- H_a: The distribution of scoring averages is not normal.

```
> data.Ex11.PGA <- read.table("http://waveland.com/Glover-Mitchell/Example11-PGA.txt",
+ header = TRUE)
> head(data.Ex11.PGA, n = 10)     # list data

##             Player Average
## 1      Tiger_Woods   67.79
## 2   Phil_Mickelson   69.25
## 3        Ernie_Els   69.31
## 4      David_Duval   69.41
## 5     Paul_Azinger   69.68
## 6       Nick_Price   69.75
## 7     Stewart_Cink   69.79
## 8     Steve_Flesch   69.80
## 9       Tom_Lehman   69.84
## 10  Loren_Roberts   69.89

> lillie.test(data.Ex11.PGA$Average)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  data.Ex11.PGA$Average
## D = 0.0451, p-value = 0.4328

> lillie.plot(data.Ex11.PGA$Average, xlab = "Average Strokes per Round")
```



```
> mean(data.Ex11.PGA$Average)     # find the mean and standard deviation

## [1] 71.2375

> sd(data.Ex11.PGA$Average)

## [1] 0.809796
```

The P value for the test is 0.4328; the scoring averages follow a normal distribution. The graph of the averages remains within the 0.95 confidence band and quite closely tracks the normal distribution with mean 71.24 and standard deviation 0.81.

Problem

1. (*a*) Example 1.8 gives snowfall measurements (in inches) for 50 consecutive years (1951–2000) in Syracuse, NY. Determine whether snowfall in Syracuse can be modeled by a normal distribution. Illustrate the result of your analysis with an appropriate plot.

(*b*) The data set `http://waveland.com/Glover-Mitchell/Problem00-3.txt` lists the monthly snowfall for Buffalo, NY for each each snow season from 1940–41 to 2013–14. Determine whether snowfall in Buffalo can be modeled by a normal distribution. Illustrate the result of your analysis with an appropriate plot.

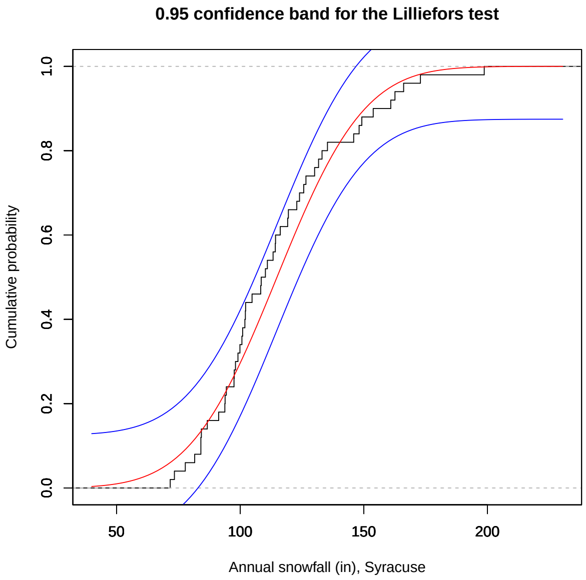
(*c*) Make side-by-side box plots of the annual snowfalls for the two cities. Test whether the mean snowfalls for the two cities are different. Your choice of test should be determined by the earlier parts of this question.

Answer

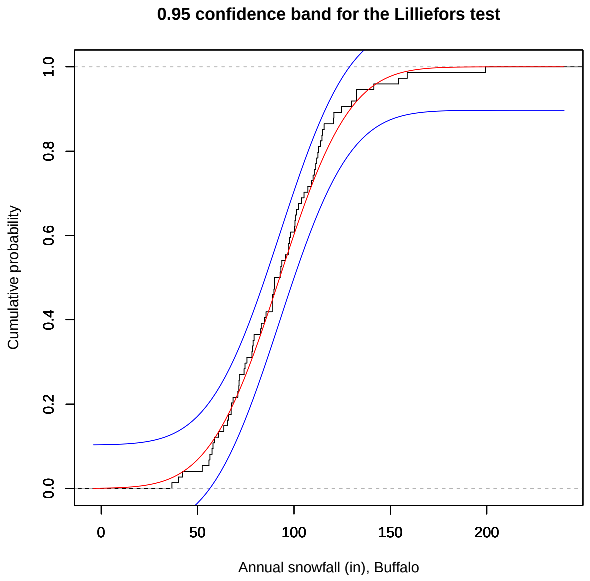
The hypotheses for each test are:

- H_0: The distribution of snowfall is normal.
- H_a: The distribution of snowfall is not normal.

```
> # Use the Lilliefors test for the Syracuse snowfall.
> lillie.test(data.Ex01.8$Snowfall)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  data.Ex01.8$Snowfall
## D = 0.1159, p-value = 0.09092

> lillie.plot(data.Ex01.8$Snowfall, xlab = "Annual snowfall (in), Syracuse")
```



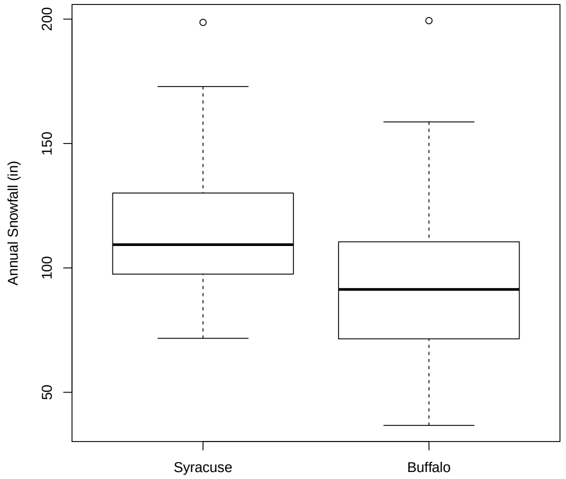
```
> # Analyze the Buffalo data in the same way.
> data.Pr00.3 <- read.table("http://waveland.com/Glover-Mitchell/Problem00-3.txt",
+ header = TRUE)
> lillie.test(data.Pr00.3$ANN)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  data.Pr00.3$ANN
## D = 0.0754, p-value = 0.3741

> lillie.plot(data.Pr00.3$ANN, xlab = "Annual snowfall (in), Buffalo")
```



```
> lillie.test(data.Ex11.17$Length)

##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  data.Ex11.17$Length
## D = 0.1244, p-value = 0.7307
```

Both analyses and plots indicate that the snowfalls in Syracuse and Buffalo follow normal distributions. Now create side-by-side box plots.

```
> boxplot(data.Ex01.8$Snowfall, data.Pr00.3$ANN,
+ names = c("Syracuse", "Buffalo"), ylab = "Annual Snowfall (in)")
```



Since both distributions were normal carry out a two-sample t test. Determine whether the variances are equal first.

```
> var.test(data.Ex01.8$Snowfall, data.Pr00.3$ANN)

##
##  F test to compare two variances
##
## data:  data.Ex01.8$Snowfall and data.Pr00.3$ANN
## F = 0.9445, num df = 49, denom df = 73, p-value = 0.8414
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.570715 1.605178
## sample estimates:
## ratio of variances
##           0.944501

> # The variances may be assumed to be equal.
> # Carry out the appropriate t test.
> #
> t.test(data.Ex01.8$Snowfall, data.Pr00.3$ANN, var.equal = TRUE)

##
##  Two Sample t-test
##
## data:  data.Ex01.8$Snowfall and data.Pr00.3$ANN
## t = 4.3067, df = 122, p-value = 0.00003371
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  12.0537 32.5615
## sample estimates:
## mean of x mean of y
##  114.8860   92.5784
```

The mean annual snowfalls in Syracuse and Buffalo differ significantly.

Index of Examples

Index of Terms