Statistical Modeling using RStudio and RMarkdown
Course Notes for MTH 220 - Data Analysis

Chester Ismay

November 21, 2014

Contents

Chapter 1

Introduction

1.1 Beginning to Use RStudio

We will be using `RStudio` throughout this course. It is an integrated development environment (IDE) for `R`, a freely available statistical computing language useful for many things including data analysis, statistical modeling, and producing beautiful graphics. `RStudio` and `R` are available for free for PC, Mac, and Linux.

For our purposes, we will be using the `RStudio` Server edition, which is available at the following link. You can log-in to the site using the same logon that you use to sign into your Ripon email using Google Apps. This cloud computing resource allows you to not have to worry about installing a version on your own computer, working with a potential different version in the labs on campus, and many other problems that go along with working on analyses in different locations. All of your files and data will be stored on the cloud provided by the folks at http://www.rstudio.org. They have also provided many useful videos there if you'd like a further introduction to using it.

1.1.1 Initial screenshot

After logging in to your `RStudio` Server account, you should see a screen that looks something similar to Figure 1.1. This figure displays three panels: **Console** in the left pane, **Environment** and **History** in the top right pane, and **Files**, **Plots**, etc. in the bottom right pane.

We also will be creating script files that will go into a fourth pane splitting with the **Console** pane. These script files will contain all of the commands that we would like to run in `R` in addition to our documentation of what the commands do and our discussion of the resulting plots and statistical analysis. We will see more about this when we work with `RMarkdown` in a later section of this chapter.

To display this fourth pane, go through the following sequence of steps: `File`, `New File`, `RScript`. You will now see the four different panes in something similar to Figure 1.2. We will type commands that we would like to execute in `R` here in the Script pane that is reminiscent of a Word processor like Microsoft Word.

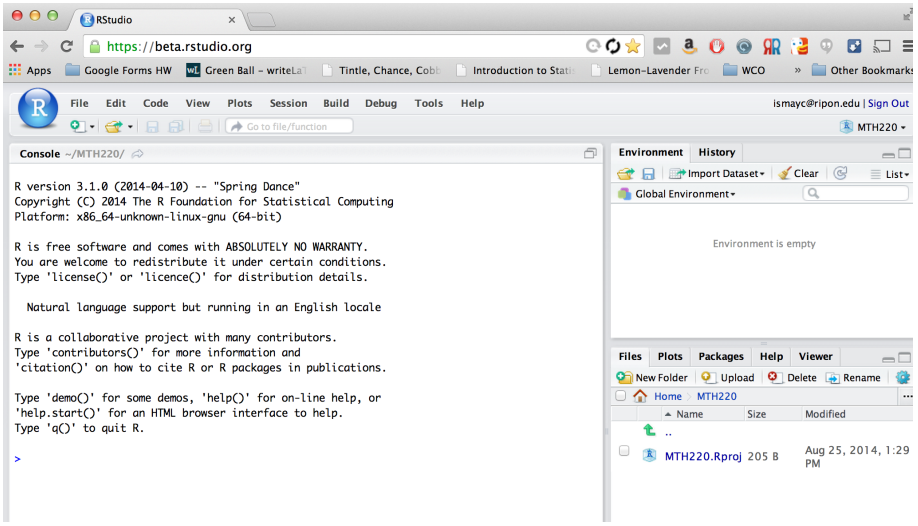


Figure 1.1: `RStudio` Initial Screen

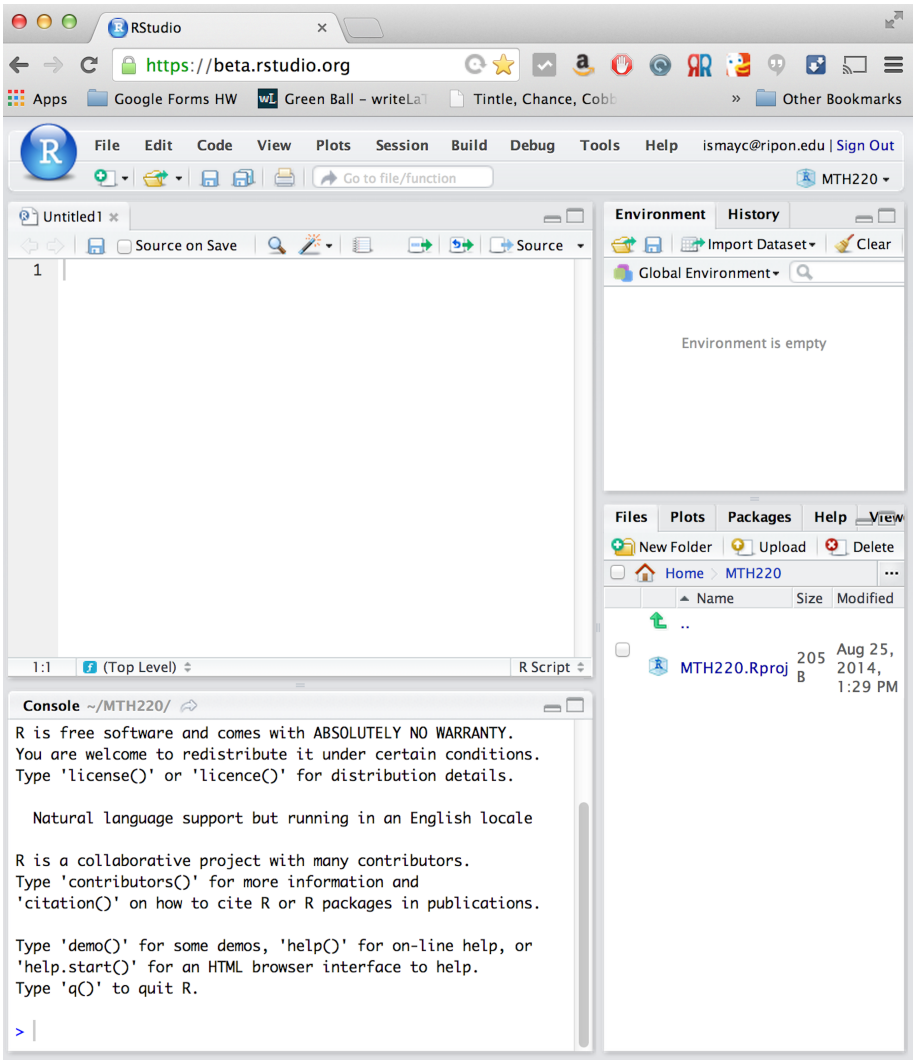


Figure 1.2: `RStudio` Screen with Text Editor

1.1.2 Basic Computations in R

To get a feeling for how to use R, we will begin with a basic arithmetic problem. Type $97 - 74$ in your Script pane. Then click the Run button in the upper right of the screen. This subtraction will be computed in the Console pane as shown below.

```
> 97 - 74
[1] 23
```

Note here that R commands in these course notes will be colored red and the (non-plot) output will be colored blue. You will learn later on how to create your own documents using RMarkdown in RStudio that will show you the R code as well as the R output. We will be doing this all throughout the course for many reasons that will be discussed later in this chapter.

You can also store computations in a variable, which you can use at a later time. In addition, you can use the # key to provide comments as to what a command or section of code does. The text immediately following the # key is ignored by R when it runs. I've done that for each of the following lines.

```
> root <- sqrt(4*9) #Store the result
> root #Display the result stored in root
[1] 6
```

It's important to note the use of the <- operator here. It puts the value of `sqrt(4*9)` into `root`. This operation makes it clear exactly what is assigned to what. We can then do calculations on this `root` variable.

```
> root^3
[1] 216
> root*7.2
[1] 43.2
```

1.2 Tips on Using R

1.2.1 Calling functions

We saw above that we could call functions in R such as the square root function. The *arguments* to the function are placed inside round parentheses. Functions can also have multiple arguments. In that case, we can specify what each of the different arguments are. The logarithm is one such example.

```
> log(16, base=2)
[1] 4
```

There also are functions that have no parameters. We must still specify the empty () next to the function name though so that R is able to recognize that it is a function and not a variable name such as `root`. It's often good practice to not name variables and functions the same thing to help reduce confusion and improve the readability of your code.

It's often useful if you can't remember the syntax of a particular function to use the ? before the function name. Try this out by typing and executing `?log`. This will be sent into the **Help** tab in the bottom right panel of `RStudio` and you should look over it there. We'll also be investigating what information the other tabs provide as the course progresses. Luckily, many of them are self-explanatory.

1.2.2 Case-sensitive

Another important feature of `R` is that it is case-sensitive. Therefore, `fit` and `Fit` correspond to two different things as do `compStatForRegression` and `compstatforregression` or any variation on capital and small letters for this phrase. BE CAREFUL WITH THIS!

1.2.3 Finish your parentheses and brackets

You may forget to close an opening parenthesis or bracket { and this can lead to a lot of headache. `R` will often let you know that it is still expecting more with a + sign in the **Console**. If you see this, check to make sure that you matched up all opening parentheses and brackets with their closers,) and/or }. If you are stuck and aren't sure what to do hit the ESC key on your keyboard to exit the current computation.

1.3 Loading Different `R` Packages

`R` is improved on a continual basis often by the implementation of different packages. These packages perform specific tasks and are often made up of functions that can be called easily in `R`. This makes doing many different types of statistical analyses simple by calling functions defined in the different packages with specified parameters/arguments. Make note of when we use different packages throughout this course packet and also how we install them if they aren't built in.

1.4 Getting Data into `R`

With this being a Statistics course, it's obvious that we need to figure out ways to work with datasets. We'd like to be able to plot them, summarize them, and do further analysis. The first step after collecting data is to figure out a way to input the data into `R`. Before we discuss ways to input our own data, we will discuss how packages also often store datasets and mention the different data sets that `R` has built-in to its standard package. We'll hold off on inputting our own data until we've actually collected some data. We'll be there soon!

Nicely enough, `R` is now distributed with the `datasets` package. (Statisticians aren't known for their creativity...) One interesting dataset in this package is the `ToothGrowth` dataset which is described in its corresponding help file as:

> The response is the length of odontoblasts (teeth) in each of 10 guinea pigs at each of three dose levels of Vitamin C (0.5, 1, and 2 mg) with each of two delivery methods (orange juice or ascorbic acid).

Remember that this help file is given by entering the `?ToothGrowth` command. You can also enter `help(ToothGrowth)` if you'd like to use the standard function notation.

It's often helpful to get a peak at the first few rows of a dataset in addition to the different variables (columns) specified in the help file. This is done using the `head` function.

```
> head(ToothGrowth)
   len supp dose
1  4.2   VC  0.5
2 11.5   VC  0.5
3  7.3   VC  0.5
4  5.8   VC  0.5
5  6.4   VC  0.5
6 10.0   VC  0.5
```

We can also get an idea of the size of the data by using the `dim` function.

```
> dim(ToothGrowth)
[1] 60  3
```

Note: Most data is stored as a *data frame* structure in `R`. There are other types of structures as well such as `array`, `string`, `matrix`, etc. We may investigate these more throughout the course.

The `head` function above doesn't tell us a lot since the data is sorted. We will now look at a dataset in the `mosaicData` package called `KidsFeet`. We must first load the `mosaicData` package by using the `library` function after using `install.packages("mosaicData")`.

Note: The MOSAIC package will be a very important one throughout this course. The developers of the package have created an easy template to handle many different types of analyses. We'll see the template a little later on and I'll be sure to point out its importance again there. The `mosaicData` package includes the datasets that used to be bundled with the MOSAIC package.

```
> library(mosaic)
> head(KidsFeet)
    name birthmonth birthyear length width sex biggerfoot domhand
1  David          5        88   24.4   8.4   B          L       R
2   Lars         10        87   25.4   8.8   B          L       L
3   Zach         12        87   24.5   9.7   B          R       R
4   Josh          1        88   25.2   9.8   B          L       R
5   Lang          2        88   25.1   8.9   B          L       R
6 Scotty          3        88   25.7   9.7   B          R       R
```

We can see here that we can easily identify what each of the variables given in this table represent. It's of utmost importance that you define your variables in this manner. It should not be hard to decipher what each column of data means. We can also clearly see that some of the variables are numeric/quantitative here and others are categorical/quantitative. This course will focus on building models involving both types of variables as explanatory and response variables.

Remember the key term *observational unit* which corresponds to what particular object a variable is measuring. For the `KidsFeet` example, all of the measurements are taken on children in a fourth grade classroom in Ann Arbor, MI in October 1997. Thus, these specific children are the observational units for this data set.

Chapter 2

Introductory Statistics Review

2.1 Plotting the Distribution of One Variable

When we work with data we often want to be able to see what the data looks like. We want to identity what sorts of patterns exist. This is defined nicely in Tintle et al.'s *Introduction to Statistical Investigations* textbook.

Definition 2.1. *The **distribution** of a variable describes the pattern of value/category outcomes.*

We will begin by discussing the distribution for one variable. For a quantitative variable, the distribution can be thought of as showing which values the variable takes on and also how frequently each of these realizations occurs. For a categorical variable, it shows the different levels/categories and a count of how many times each of these levels occurred.

We can create a nice template for producing just about any kind of statistical plot we could think of:

`plotName( ~ variable, data = dataName)`.

We have three different pieces of information to give `R` in order to create a visual representation of the distribution we are working with:

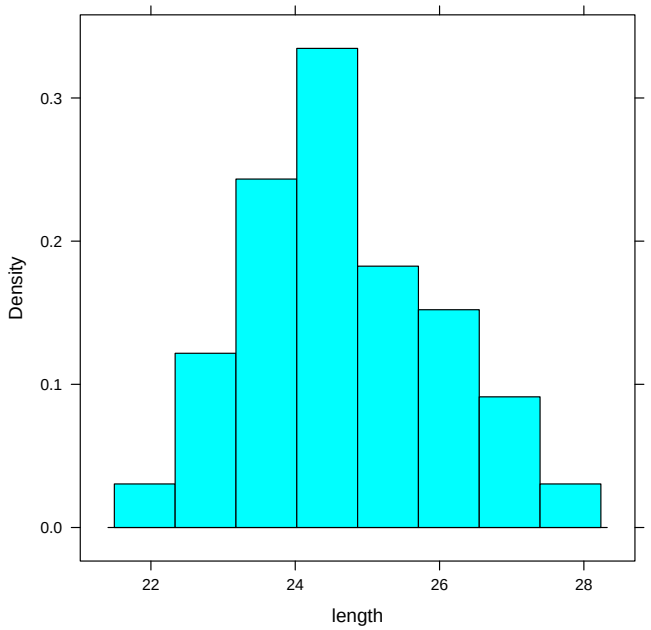
1. `plotName` - the kind of plot (`histogram()`, `bargraph()`, `bwplot()`)
2. `variable` - the name of the variable in our dataset
3. `dataName` - the name of the data frame the variable is in

2.1.1 Histograms for quantitative variables

Recall that histograms are the most common way to give a visual representation of the distribution of a quantitative variable. Histograms depend on the argument either being the *number of bins*, denoted by the argument `n`, or the *width of the bin*, denoted by `width`.

Using the MOSAIC package we can specify either of these as arguments to the `histogram` function. We will investigate the `length` variable, which denotes the length of the longer foot in centimeters, in the `KidsFeet` dataset.

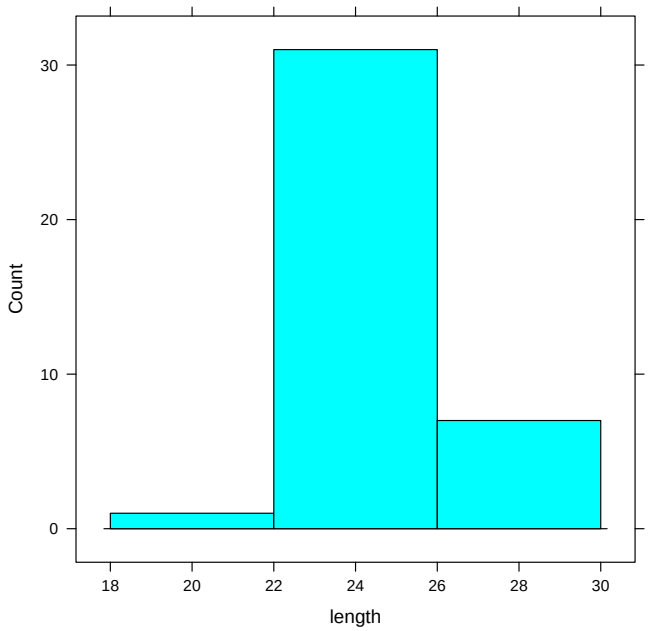
```
> histogram( ~ length, data=KidsFeet, n=8)
```



We can also specify the desired width of the bins if we use the `histogram` function defined in the MOSAIC package. We specify the width to be 5, for example. Think about the differences between the following histogram and the one with 8 as the number of bins.

Note that for the first histogram, the vertical axis is "Density." Histograms are often displayed with "Frequency" as the vertical axis. This can be achieved by adding the parameter `type` and setting it equal to `'count'`.

```
> histogram( ~ length, data=KidsFeet, width=4, type='count')
```



2.1.2 Distributional shape

Looking at the two histograms, we can see that the choice of bin size can have a big effect on describing how a given variable looks. As a general rule, you should make more than one histogram for each analysis to be able to assess the spread, center, and shape of the distribution.

As a review from introductory statistics, there are many common terms used to describe shape:

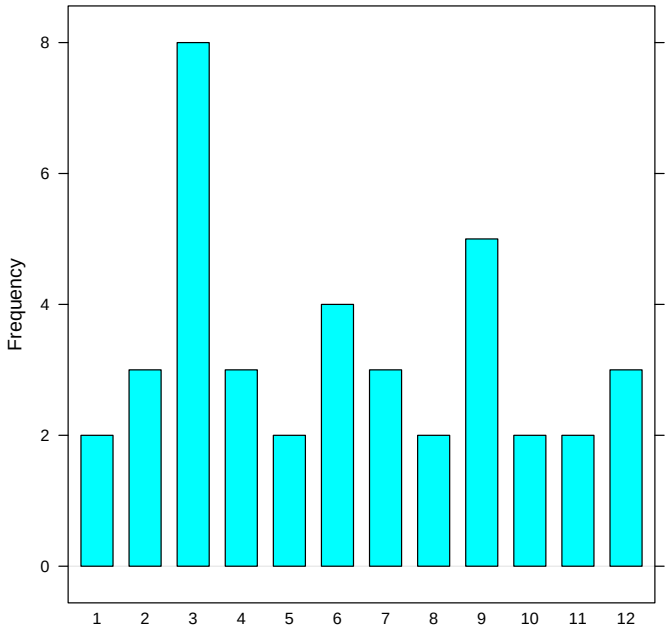
- *symmetric* - The left and right sides mirror each other.
- *skewed* - The distribution has a long tail in one direction or the other. We say that the distribution is "skewed left" if the tail appears on the left and "skewed right" if the tail appears on the right.
- *uniform* - The distribution is flat. In other words, the heights of all the bars in the histogram are close to the same.
- *unimodal/bimodal/trimodal* - One bump/two bumps/three bumps in the distribution.
- *outlier* - An observed data value that does not fit with the overall pattern in the data. It usually is far away from the others.

We will discuss box-and-whisker plots in the next chapter. If you'd like to work with them some before then, the command is `bwplot`. Remember to use `?bwplot` for more information.

2.1.3 Bar graphs for categorical variables

A related plot to the histogram is the bar graph, which is used to plot categorical variables. Notice below how similar the commands are for creating histograms and bar charts. Remember to use bar charts for categorical variables and histograms for quantitative variables. They cannot be interchanged. We will now show a bar graph corresponding to the month of birth for the 39 students in the `KidsFeet` dataset.

```
> bargraph( ~birthmonth, data=KidsFeet)
```



2.2 Looking for Relationships Between Two Variables

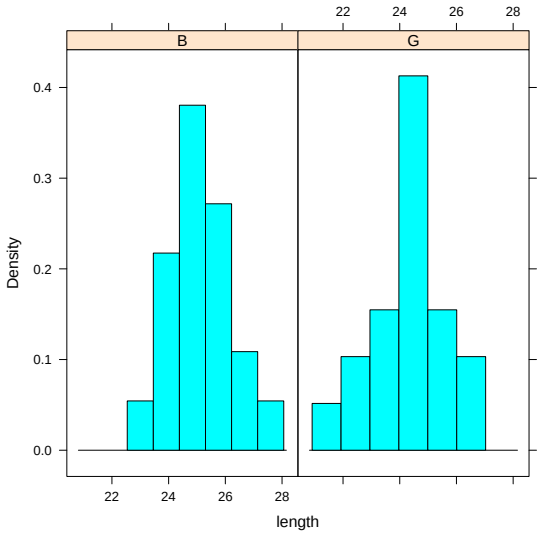
2.2.1 Conditional plots

We can extend the general template we gave before of `plotName( ~ variable, data = dataName)` slightly to look at different groups based on a condition. The new syntax is then

`plotName( ~ variable | condition, data = dataName)`.

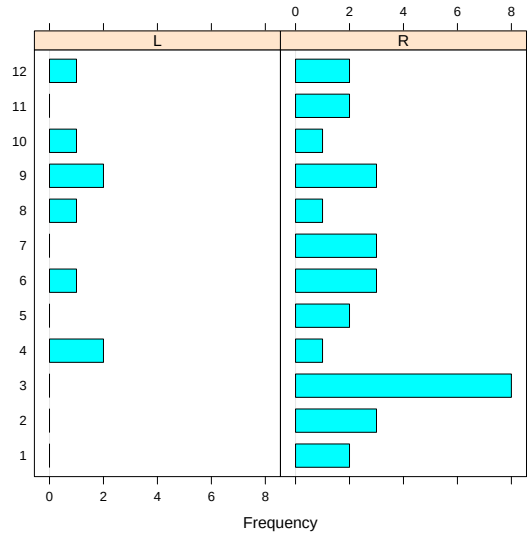
As an example, we can look at the distribution of `length` for boys versus girls.

```
> histogram( ~length | sex, data=KidsFeet)
```



The same conditioning can be applied to bar graphs such as those using `birthmonth` from above. I've also added the horizontal tweak here. There are almost always extra options that one can add when calling functions defined in either the standard `R` environment or in packages. I recommend you try lots of different options, especially with plots, to get your desired output.

```
> bargraph( ~birthmonth | domhand, data=KidsFeet, horizontal=TRUE)
```



In both the quantitative and qualitative cases, this is often a useful tool in exploratory data analysis when you are trying to get an idea of what the cause of skew may be in your distribution. You can also use this to better understand how one subgroup compares on a variable to another (or more than one) subgroup.

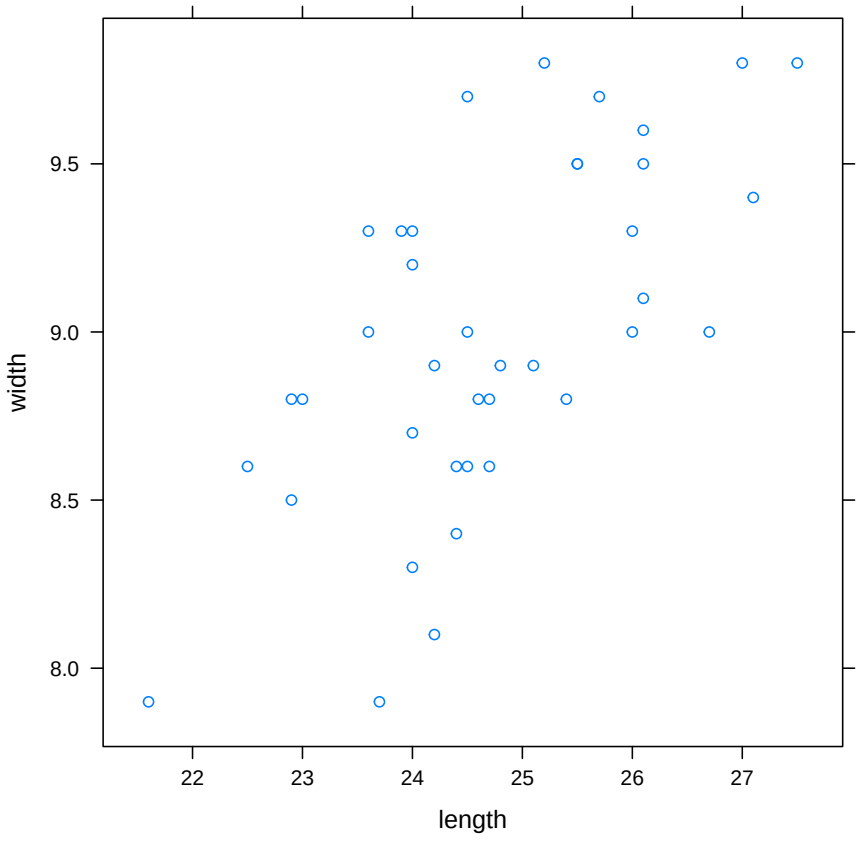
2.2.2 Scatterplots

The conditional plots provide a useful tool for looking at the distribution split on levels of a certain categorical variable. Most often we want to look at the relationships between two quantitative variables and usually this is done with a *scatterplot*. In `R`, you can create scatterplots using the `xyplot()` function. We can look at a plot of the `width` and `length` variables from our `KidsFeet` dataset.

It's important to always pause before you actually type in your code to perform an action and think about what your guesses are as to what the plot will look like. This is a perfect example. How do we expect shoe width and shoe length to be related? As one goes up, do we expect the other to decrease? To increase? To stay the same?

These are important questions that you should always be thinking about as you conduct statistical analyses. If you make an initial guess, you may be amazed by the findings in the produced plot or you might have good reasons to back up why the plot looks the way that it does. Either way, always think about what you'd expect BEFORE you have the computer do the busy work.

```
> xyplot(width ~ length, data=KidsFeet)
```



Did you think that there would be a general positive linear relationship between length and width? If so, that's what is given here in the plot. To better get an idea of the relationship between `length` and `width` of the longer feet of these children, we can also have `R` calculate the correlation function. Remember from Stat 1 what the range of possible values of correlation is.

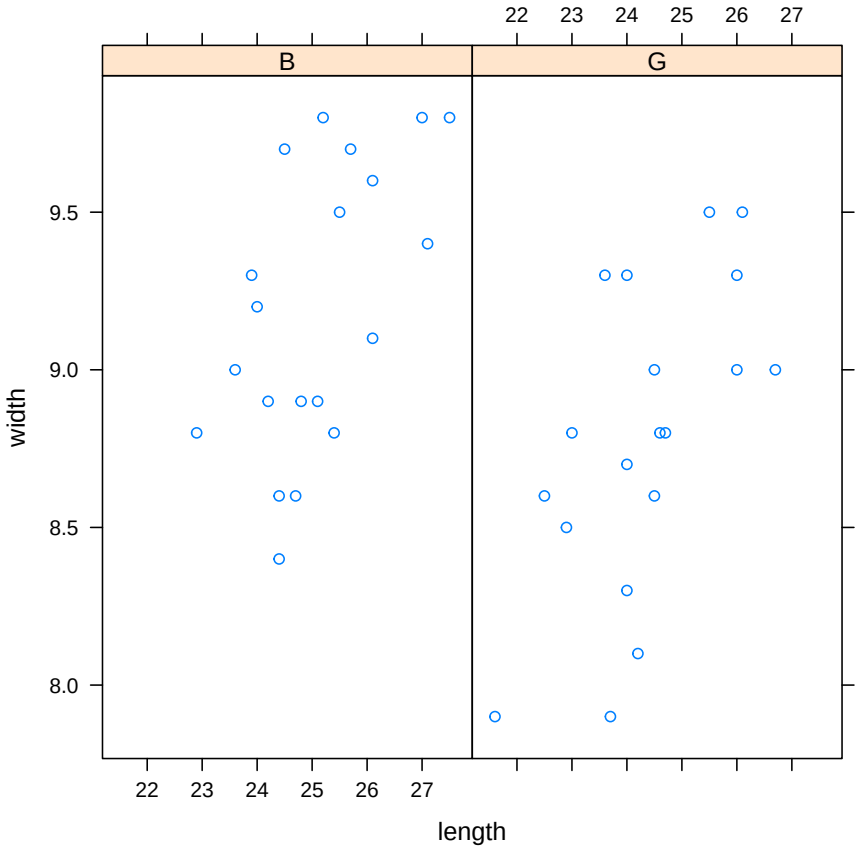
```
> cor(width ~ length, data=KidsFeet)
[1] 0.6410961
```

We see we have a value of about 65% which corresponds to a moderate positive linear relationship. Note in the `cor()` function call that the template remains the same as that with `xyplot()`. If you are carefully following along, you may have also noticed that we now have two variables on either side of the 'tilde' sign. We can read that as `width` is modeled by `length` or `width` depends on `length`.

To further analyze the scatterplot, we can do the same sort of conditioning that we did before. We will now look at the relationship between two variables split on the different choices of a third variable. Confused yet?

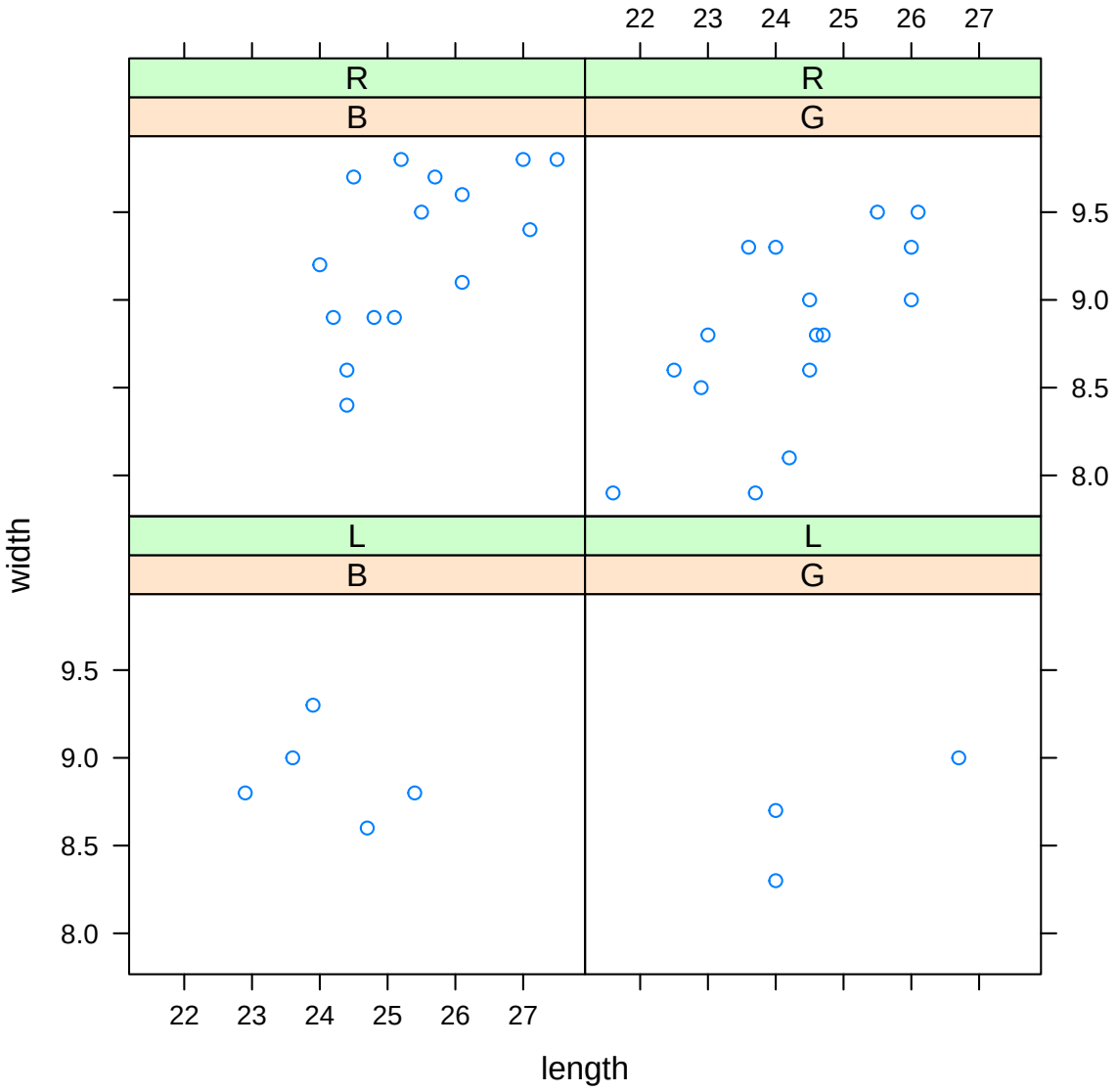
Let's look at an example using this same `KidsFeet` dataset with `sex` being the condition. Do we expect to see similar scatterplots for the two sexes? Do we believe that males and females behave differently in terms of the lengths and widths of their feet? It's important to get in the habit of asking yourself these types of questions always before you do the analysis.

```
> xyplot(width ~ length | sex, data=KidsFeet)
```



Extending one more time, suppose we wanted to look at how `width` and `height` differ over the categories of `sex` AND `domhand`. We would expect to now see ... four scatterplots corresponding to each of the four possible combinations of boy/girl and left/right. Our dataframe only has 39 observations so slicing and dicing like this isn't necessarily the best thing to do but it should give you some idea of just how simple it is to use `R` to explore a given dataset.

```
> xyplot(width ~ length | sex + domhand, data=KidsFeet)
```



2.3 Reproducible Research

You've likely all seen these types of things before. It's a review after all, isn't it? I'm sure a question you are asking right now is "How do I export this plot and code into Word so that I can work with it by adding comments?" Sure, there are ways to do that. They are easily found in `RStudio` and a simple web search will lead you to directions on how to do it. I'm not going to tell you how to do

that here because I'm trying to break you of a habit. This habit stretches far and wide across lots of the scientific community. The problem is that all too often research is NOT reproducible!

Have you ever tried to go back to a project you did years ago and wondered how you produced the plots there? What point-and-click procedures did you do in Excel to get that beautiful bar graph of **disposable income** versus **Kardashian sister**? It's often hard to remember these sorts of things and they are hardly ever documented carefully.

This also becomes a HUGE problem in the workforce. When someone creates a project report and then five years later after that person has taken a position elsewhere, the company needs to produce a new report ... what are they to do!? Never fear, `R` is here!

In all seriousness, `RStudio` now provides a remarkably simple way to produce high-quality documents that include the code and output of results. This allows for the research to be reproducible! Think about the time that could be saved if everyone just learned how to program in `R`. To make things even better, `R` is free! (And no, I'm not paid by the makers of `R` to endorse their product. I really, really like it that much!)

2.3.1 R Markdown

In this course, we will be using the `RMarkdown` language to produce code, `R` output, and discussion. You will turn in all of your homework and projects using `RStudio` and specifically `RMarkdown`. Markdown is a simple mark-up language that allows users to merge plain text with formatting options such as section headers, numbered lists, bulleted lists, bold font, italic font, and many others. `RMarkdown` allows us to incorporate `R` input and output into our document easily. These files will then be exported to HTML by simply clicking the button to **Knit HTML** right above the text editor window where you type the Markdown text. Similarly, you can also export to PDF.

A new `RMarkdown` document is created by clicking on `File`, `New File`, `R Markdown`. By default a simple Markdown example is loaded into the text editor window. If you click on the **Knit HTML** button you can start to get an idea of how to create different text formats using `RMarkdown`. More details on different options in using `RMarkdown` is available at this link. I encourage you to play with this and get used to it.

One of the cool things about the `RStudio` implementation is the ability to add chunks of `R` code into the document. This can be done by clicking on the **Chunks** option near the top of the `RStudio` window near the green box with a white C in it. Inside this you will place your `R` input and `RMarkdown` will automatically put in your `R` output immediately following. You can see this in the default template with the `summary(cars)` and `plot(cars)` commands.

This is how we will be producing documents in this course. Another great thing about this style of research is that when you modify your comments or change one of your commands you don't have to worry about deleting the old output from your document and copying-and-pasting in your new plots. NO MORE COPY-PASTE! GOODBYE!

This sentiment in favor of reproducible research was eloquently laid out thirty years ago:

> Let us change our traditional attitude to the construction of programs: Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to humans what we want the computer to do.
>
> – Donald E. Knuth, *Literate Programming*, 1984

2.4 Summarizing Data

In addition to looking at what the distribution of a dataset looks like, it is often useful to attempt to summarize data using a few standard measures. Many of these will be familiar to you.

We will see that the same standard template that we used with plotting in the first two sections of this chapter can also be used when we'd like to make function calls for numerical summaries. We will begin with summarizing data in table form and then shift to quantitative data measures.

2.4.1 Categorical data tabulation

Earlier in this chapter we looked at a way to visually display a categorical variable using a bargraph. Sometimes it is useful to get an idea of the exact counts of the levels for a given variable. This is done using the `tally` command. (Notice the same template as we used before with the plots below.) We will show one example without the conditional statement and one with from the `KidsFeet` dataset.

```
> tally( ~birthmonth, data=KidsFeet)
 1  2  3  4  5  6  7  8  9 10 11 12
 2  3  8  3  2  4  3  2  5  2  2  3
```

Notice that this gives the counts of the different number of students corresponding to each of the 12 different months. This command is nice because it again gives exactly what we would expect "tally" to do on the variable `birthmonth` in the dataset `KidsFeet`.

We can also specify the `format` to show the values as proportions or as percentiles. (Remember that most functions have these extra options that we can add to tweak our output. Most often a Google search or use of the `help()` command can provide you with more information about these extra tweaks.)

```
> tally( ~birthmonth, data=KidsFeet, format="prop")
         1          2          3          4          5          6          7
0.05128205 0.07692308 0.20512821 0.07692308 0.05128205 0.10256410 0.07692308
         8          9         10         11         12
0.05128205 0.12820513 0.05128205 0.05128205 0.07692308

> tally( ~birthmonth, data=KidsFeet, format="perc")
        1         2         3         4         5         6         7         8
 5.128205  7.692308 20.512821  7.692308  5.128205 10.256410  7.692308  5.128205
        9        10        11        12
12.820513  5.128205  5.128205  7.692308
```

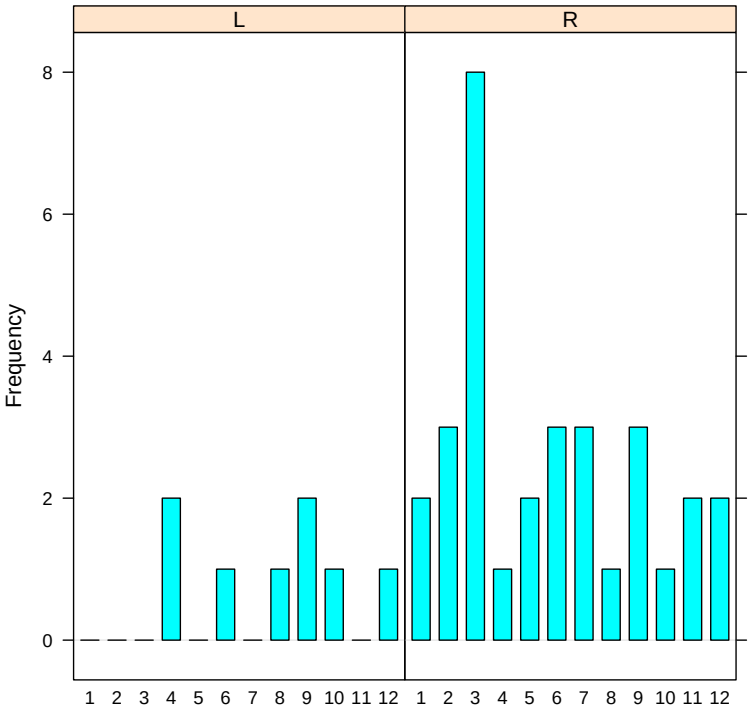
One of the great things about `R` is how easily customizable much of the output is to your liking. Many software packages struggle in giving you output such as this. You often need to convert it over to a different program in order to obtain the desired output. We now will look at the table comparing `birthmonth` for left-handed students to right-handed students. Pay careful attention to the differences here.

```
> tally( ~birthmonth | domhand, data=KidsFeet)
```

```
           domhand
birthmonth          L          R
         1  0.00000000 0.06451613
         2  0.00000000 0.09677419
         3  0.00000000 0.25806452
         4  0.25000000 0.03225806
         5  0.00000000 0.06451613
         6  0.12500000 0.09677419
         7  0.00000000 0.09677419
         8  0.12500000 0.03225806
         9  0.25000000 0.09677419
         10 0.12500000 0.03225806
         11 0.00000000 0.06451613
         12 0.12500000 0.06451613
```

Think about what this output is providing for us. By default, in the conditional view, it shows the values as decimals. A little thinking and observation provides some hints as to what these numbers mean. Let's review what the corresponding bargraph looked like.

```
> bargraph( ~ birthmonth | domhand, data=KidsFeet)
```



We can see that March corresponds to the largest value for right-handers in both the tabulated results and in the bargraph. Closer inspection of the left-handers gives us the direct relationship here. What do we get if we add down each of the two columns for lefties and righties? We get the value of 1. By default, the values are given as relative frequencies when we condition. This is often helpful in comparing two different groups along the same variable.

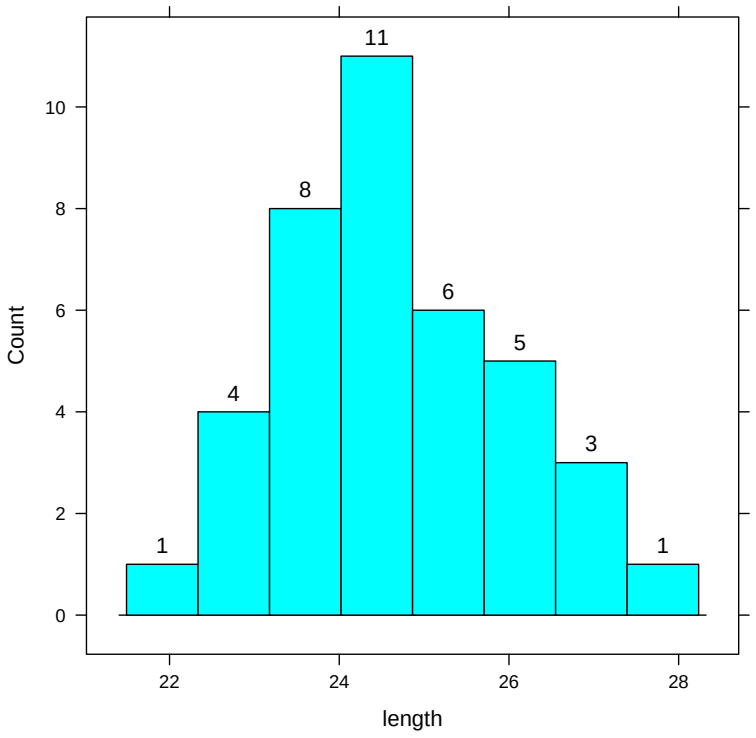
2.4.2 Numerical data tabulation

Similarly, we can create a table of values for numerical data. Oftentimes this will not be useful to use though since it will list out all possible instances of a quantitative variable and how many times each realization occurred. The `KidsFeet` data have a couple of quantitative variables in `length` and `width` but a listing of all possible lengths and/or weights from those 39 students and how many times they occurred probably isn't the kind of exploratory data analysis we'd like to do. Creating a histogram that bins the data together makes for much easier viewing.

The `tally` command does exist for numerical data though. Just be careful in using it instead of the histogram function. There's also a helpful add-in tweak to the `histogram` function in the MOSAIC package that makes the `tally` command almost entirely not useful for quantitative data.

Recall the first histogram example from earlier in this chapter on the `length` of the children's feet grouped into eight bins. We can now add the `label` argument and set it to TRUE to have the different counts for each of the bins appear above the bins if we use `type='count'`.

```
> histogram( ~ length, data=KidsFeet, n=8, type='count', label=TRUE)
```



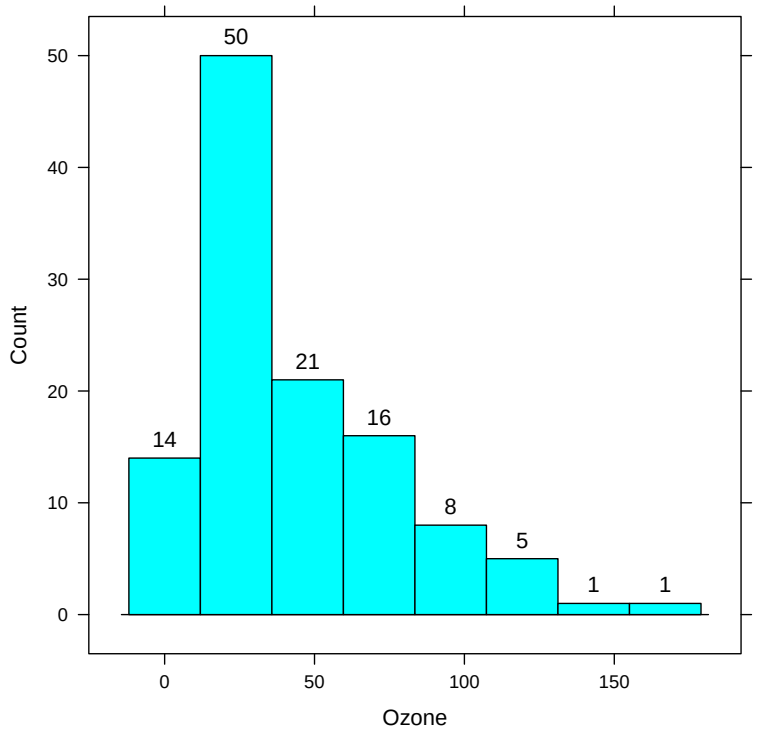
2.4.3 Measures of Center and Spread

2.4.3.1 Based on the mean

I would be remiss if I didn't discuss how to calculate the standard statistical measures that you've all learned to grow to love throughout your lifetimes. I'm amazed that we've gone this far without mentioning them. We've pretty much done everything there is to do with the `KidsFeet` dataset. I'll leave the numerical summary analysis as an exercise at the end of the chapter.

Let's work with daily air quality measurements in New York for May to September of 1973. These can be found in the `airquality` dataset in the `datasets` package. We will investigate the variable `Ozone`. To help motivate the numbers, I've included a histogram of the variable and then we will discuss the numeric summaries.

```
> histogram( ~ Ozone, data=airquality, type='count', label=TRUE)
```



After looking at the histogram, you should begin to ask yourself the following questions:

1. Is the dataset skewed to the right, to the left, or symmetric?
2. Will the median be much greater than the mean, much less than the mean, or about the same as the mean?
3. Do we expect the standard deviation to be large?
4. How do we interpret the standard deviation for this data set?

Let's discuss the results of these questions. The dataset is skewed to the right since it has a long tail in that direction. Recall that right-skewed distributions have means that are greater than medians since the mean is pulled in the direction of the skew. The standard deviation should be reasonably large (though what are we comparing it to?) since there is quite a bit of spread in the data. Much sure to look at the horizontal scale when you are observing the spread.

The standard deviation is often thought of as the "average deviation from the mean." The quotes are there because it isn't exactly true but it is often good enough for interpretation, which is what we are most often concerned with. Think about how we could use this definition for this data set.

Note: The standard deviation (like the mean) is strongly affected by outliers. We often should look at other variability measurements when the data is far from symmetric and highly skewed.

We will now give the standard measures of center and spread for this data set.

```
> mean( ~ Ozone, data=airquality)
[1] NA

> sd( ~ Ozone, data=airquality)
[1] NA
```

Not what you were expecting, huh? It turns out that some of the values in this data set are missing. R often stores missing data as NA. This can be seen when we output this variable to the screen. It's always a good idea to first take a look at the dataset before you do too many statistical analyses!

```
  [1]  41  36  12  18  NA  28  23  19   8  NA   7  16  11  14  18  14  34   6
 [19]  30  11   1  11   4  32  NA  NA  NA  23  45 115  37  NA  NA  NA  NA  NA
 [37]  NA  29  NA  71  39  NA  NA  23  NA  NA  21  37  20  12  13  NA  NA  NA
 [55]  NA  NA  NA  NA  NA  NA  NA 135  49  32  NA  64  40  77  97  97  85  NA
 [73]  10  27  NA   7  48  35  61  79  63  16  NA  NA  80 108  20  52  82  50
 [91]  64  59  39   9  16  78  35  66 122  89 110  NA  NA  44  28  65  NA  22
[109]  59  23  31  44  21   9  NA  45 168  73  NA  76 118  84  85  96  78  73
[127]  91  47  32  20  23  21  24  44  21  28   9  13  46  18  13  24  16  13
[145]  23  36   7  14  30  NA  14  18  20
```

We, thus, have to add the "remove N/A" tag to get our mean and standard deviation.

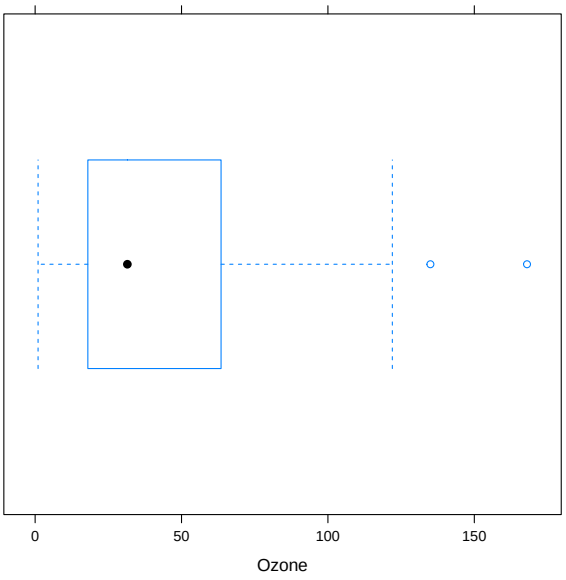
```
> mean( ~ Ozone, data=airquality, na.rm=TRUE)
[1] 42.12931

> sd( ~ Ozone, data=airquality, na.rm=TRUE)
[1] 32.98788
```

2.4.3.2 Based on the median

Often if the dataset is skewed it is better to use the median and its derivative (no, not the math term) measures to look at the center and spread of a data set. Remember that the median is **resistant** to outliers; it does not change much when a few small or large outliers are added to a dataset. The mean can change drastically with the addition of even one outlier.

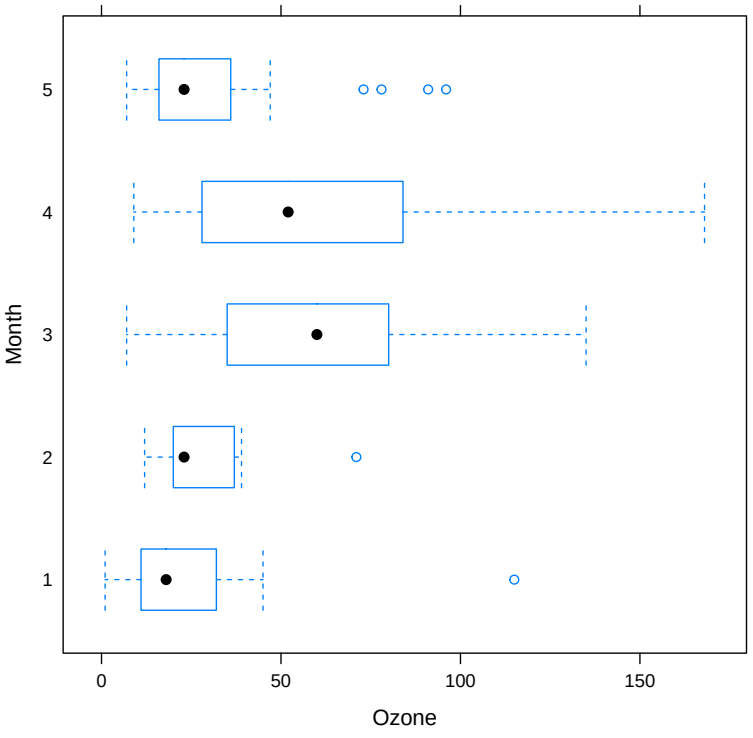
Recall that the five-number summary lists the minimum of a distribution, the 25th percentile Q_1, the 50th percentile (the median), the 75th percentile Q_3, and the maximum. This is often a much better way of seeing the variability in a skewed dataset. Also recall that we can plot this summary measure in what is known as a box-and-whisker plot or boxplot. We mentioned earlier in the chapter the command with which we can create this plot.

```
> bwplot( ~ Ozone, data=airquality, na.rm=TRUE)
```

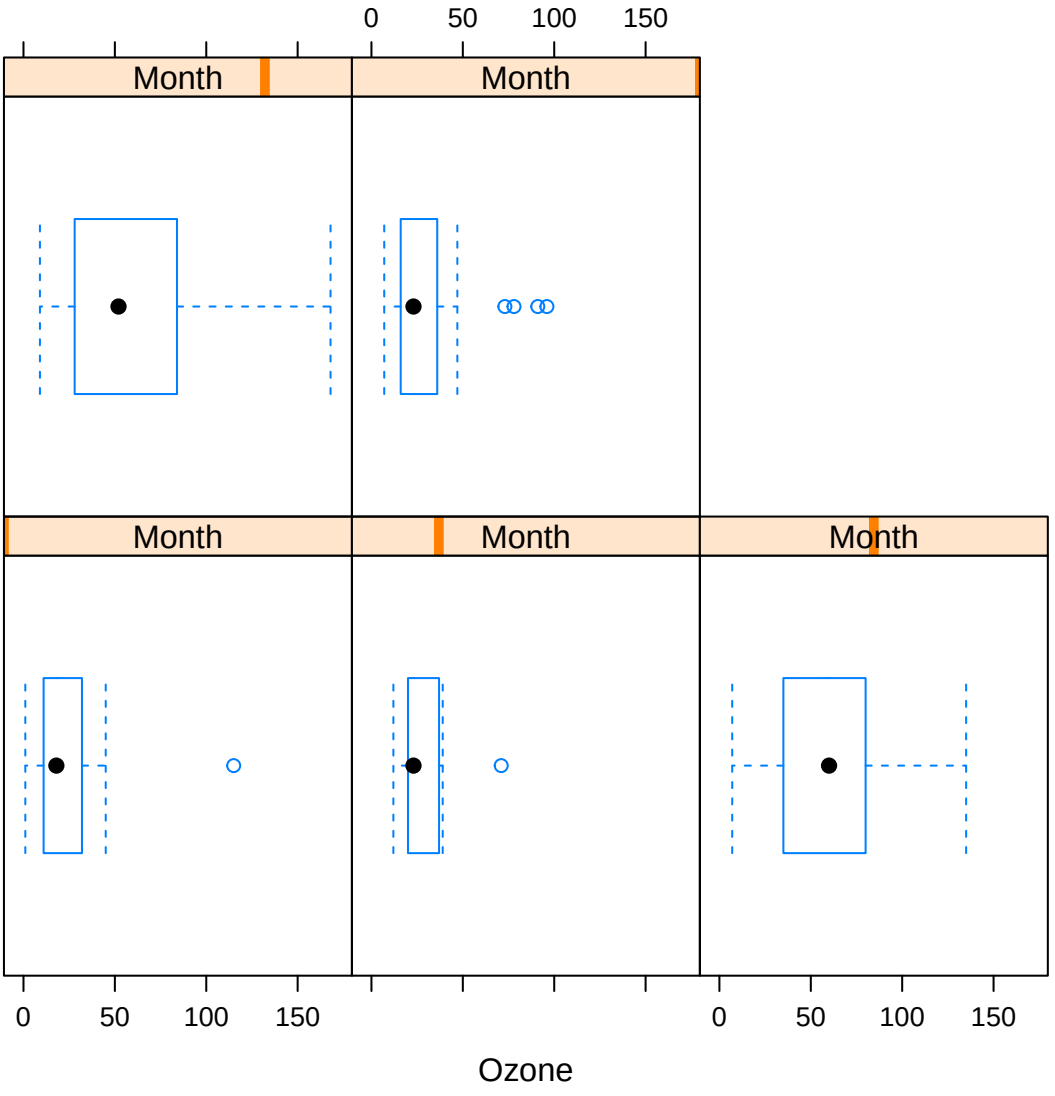


Note that the `na.rm=TRUE` option could be removed since plots by default ignore the missing data. It was left in here to accentuate that we have missing data. We can also create comparative boxplots by either using the conditioning or by inserting a second variable. (You can see that it's often better to use the `y ~ x` option as the plots are much easier to work with.) Think about what code you could run to get the boxplot to be a side-by-side boxplot instead of a comparative boxplot like this.

```
> bwplot(Month ~ Ozone, data=airquality, na.rm=TRUE)
```



```
> bwplot( ~ Ozone | Month, data=airquality, na.rm=TRUE)
```



2.4.3.3 Putting it all together

The Project MOSAIC authors have also included a nice function that provides many of the summary statistics all at once. Sometimes it is helpful to use this command first if you'd rather not look at all of the data at once in your dataset. I still urge you to at least take a peak at your dataset if you can though.

```
> favstats( ~ Ozone, data=airquality)
 min Q1 median    Q3 max     mean       sd   n missing
   1 18   31.5 63.25 168 42.12931 32.98788 116      37
```

We can see that this would have told us that there were 37 missing elements out of the total 116 measurements.

2.5 Exercises

Your solutions to these problems should be done in `RStudio` using an `RMarkdown` document and choosing the **Knit HTML** or **Knit PDF** option. Create a document that clearly labels each of the problems with their exercise number. Your document should include your `R` commands as well as the plots that were created by these commands. Make sure to also carefully answer the questions by discussing the output of the plots or the code wherever necessary.

2.1) Change the number of bins from 8 to 4, 16, and 32 in the histogram for the `length` variable in the `KidsFeet` dataset in the `mosaicData` package. Discuss how the histograms change depending on the number of bins. You should be clear in discussing shape and characteristics of each histogram.

2.2) The `alr3` package contains a data set with information about the eruptions of the Old Faithful geyser at Yellowstone National Park. (In order to get access to this package, you may need to install it. This is done by running the command `install.packages("alr3")`. After it is downloaded and installed, remember to use the `library(alr3)` command to get access to this package. You can find information about a lot of different packages at this link.)

- Create a scatterplot of the two variables in this data set.
- Discuss what each of the variables represent and also why you chose the explanatory and response variables as you did instead of the other way around.
- What insights do you have about Old Faithful eruptions from this scatterplot?

2.3) These questions refer to the `CPS85` data frame in the `mosaicData` package.

- Where does this data in the come from? (Remember that there is a simple command that can give you more information.)
- What are the observational units?
- How many OUs are there?

2.4) Carefully look through the help file for `CPS85` and decide on a numerical variable to further analyze. Make an appropriate plot and discuss what you see in the plot using appropriate statistical terminology.

2.5) Repeat the previous question but for a qualitative variable in `CPS85`.

2.6) Create and analyze a plot from the the `CPS85` data set that looks at the relationships between two or more variables.

2.7) Look over the details of the many different data sets in the `datasets` package that is built into `R`. This listing is available at this link. **Choose a different data set for each of the following parts.**

- Make a plot of a categorical variable in a data set and discuss what this plot tells you about the variable.
- Make a plot of a quantitative variable in a data set and discuss your findings.

- Make two conditional plots which look at the relationships of three variables in a data set. Carefully discuss your findings.
- Give a brief summary of why you chose the three data sets and how the plots helped you better understand these data sets, in general.

2.8) Use `R`'s help system to find out what the `i1` and `i2` variables represent in the `HELPrct` data frame.

- Create histograms for both `i1` and `i2` and comment on what you discover.
- What are the shapes of each of these distributions? Explain.
- Are there any outliers? How did you conclude if there were or were not any?
- Compare and contrast the distribution of `i1` among men and women.
- Compare and contrast the distribution of `i2` among the three groups of `substance`.

2.9) Calculate the mean, median, standard deviation, IQR, and five-number summary for the `length` and `width` variables in the `KidsFeet` dataset. Discuss any important findings from this summary information.

2.10) Repeat the previous exercise comparing males to females.

2.11) Create side-by-side boxplots of the variable `i1` (average number of drinks per day) for the different levels of `substance` in the `HELPrct` data frame. For each level, explain the skew of the distribution of the variable by comparing the mean to the median. (You shouldn't do any calculation for the mean here but analyze the skew and predict where the mean would fall in relation to the median.)

2.12) Compute the mean and median values of `i1` from the last problem.

2.13) Using the `Births78` dataset in the `mosaicData` package, create a scatterplot of `births` by `dayofyear`.

- Discuss some reasons for the peaks and valleys in the plot.
- Why are there two parallel waves? (Hint: The command `xyplot(births ~ dayofyear, Births78, groups=dayofyear%%7)` may be helpful.)

The following four exercises refer to variables in the `HELPrct` dataset.

2.14) Create a histogram of the variable `age` and discuss the shape of the distribution.

2.15) Calculate the mean, median, and standard deviation of the variable `age`. Give an interpretation of the standard deviation in the context of this problem.

2.16) Use the `favstats` command to compare the distribution of `age` over the three different levels of `substance`. Discuss how substance affects age based on this output.

2.17) Create a table of percentages corresponding to all three levels of `substance` and two levels of `sex`. Also include the marginal totals.

Chapter 3

Review of Inference

3.1 Inputting Your Own Data Into R an Existing Data File

Before we get into hypothesis testing and confidence intervals, it's important that we discuss how to get our own data into R. Up to this point we have discussed ways to work with data in the many packages that are available built-in to R or via the downloading of packages using the `install.packages()` and `library()` commands. If you are actually conducting experiments and collecting data, it is also important to understand how to import these datasets into R.

3.1.1 Importing data from CSV files

Most often it is easiest to import data that is stored in the form of a table in the CSV data format. CSV stands for comma separated values and it is one of the formats with which you can save data in Excel, Google Spreadsheets, and many other statistical software packages. R has a built-in command to read data from CSV files and it is the `read.csv()` function. An example is below which gives data, from the Social Security Administration, on the proportion of baby names for boys and girls from 1880 to 2008

```
> babyNames <- read.csv('http://www.calvin.edu/~rpruim/data/baby-names.csv',
+                       header=TRUE)
> head(babyNames)
  year    name     prop sex
1 1880    John 0.081541 boy
2 1880 William 0.080511 boy
3 1880   James 0.050057 boy
4 1880 Charles 0.045167 boy
5 1880  George 0.043292 boy
6 1880   Frank 0.027380 boy
```

You can also read data from a file stored locally on your computer. It is easiest to read from a file stored in the same directory as where your current working directory is in RStudio Server. Let's do this with the `baby-names.csv` file from Randall Pruim's website. We first need to download the CSV file to our computer. We will then upload it to one of our RStudio Server folders.

Go to `http://www.calvin.edu/~rpruim/data/` and then right-click on `baby-names.csv` and select to save the file. After saving, click on the **Upload** option in the bottom right panel of `RStudio`. Make sure to put the file in the same directory as where you are currently saving files. You can then follow the same type of commands as we did earlier when we retrieved from the web.

```
> babyNames <- read.csv('baby-names.csv', header=TRUE)
> head(babyNames)
  year    name     prop sex
1 1880    John 0.081541 boy
2 1880 William 0.080511 boy
3 1880   James 0.050057 boy
4 1880 Charles 0.045167 boy
5 1880  George 0.043292 boy
6 1880   Frank 0.027380 boy
```

3.1.2 Importing data from other types of files

Another common format for data is white space delimited files. These are often stored as plain text files in the TXT format. `R` can easily import these types of data files using the `read.table()` function as seen below in the example using data on the live births in the US for each day in 1978.

```
> births <- read.table('http://www.calvin.edu/~rpruim/data/births.txt',
+                      header = TRUE)
> head(births)
    date births datenum dayofyear
1 1/1/78   7701    6575         1
2 1/2/78   7527    6576         2
3 1/3/78   8825    6577         3
4 1/4/78   8859    6578         4
5 1/5/78   9043    6579         5
6 1/6/78   9208    6580         6
```

If you load the `mosaic` package, you can also make use of the `read.file()` function which automatically can figure out which one of the `R` import commands to call based on the filename's extension. If you use `read.file()` you can omit the `header=TRUE` option since, by default, the `mosaic` package authors assume that you have meaningful column names corresponding to your variables. You do, right!?

```
> babyNames2 <- read.file('http://www.calvin.edu/~rpruim/data/baby-names.csv')
> head(babyNames2)
  year    name     prop sex
1 1880    John 0.081541 boy
2 1880 William 0.080511 boy
3 1880   James 0.050057 boy
4 1880 Charles 0.045167 boy
5 1880  George 0.043292 boy
6 1880   Frank 0.027380 boy
```

3.2 Hypothesis Testing using Simulation on a Proportion

We will begin our review of inference with a famous example, often called "The Lady Tasting Tea" experiment. What follows is taken from a book by David Salsburg entitled *The Lady Tasting Tea*:

> It was a summer afternoon in Cambridge, England, in the late 1920s. A group of university dons, their wives, and some guests were sitting around an outdoor table for afternoon tea. One of the women was insisting that tea tasted different depending upon whether the tea was poured into the milk or whether the milk was poured into the tea. The scientific minds among the men scoffed at this as sheer nonsense. What could be the difference? They could not conceive of any difference in the chemistry of the mixtures that could exist. A thin, short man, with thick glasses and a Vandyke beard beginning to turn gray, pounced on the problem. "Let us test the proposition," he said excitedly. He began to outline an experiment in which the lady who insisted there was a difference would be presented with a sequence of cups of tea, in some of which the milk had been poured into the tea and in others of which the tea had been poured into the milk...
>
> So it was that sunny summer afternoon in Cambridge. The lady might or might not have been correct about the tea infusion. The fun would be in finding a way to determine if she was right, and, under the direction of the man with the Vandyke beard, they began to discuss how they might make that determination.
>
> Enthusiastically, many of them joined with him in setting up the experiment. Within a few minutes, they were pouring different patterns of infusion in a place where the lady could not see which cup was which. Then, with an air of finality, the man with the Vandyke beard presented her with her first cup. She sipped for a minute and declared that it was one where the milk had been poured into the tea. He noted her response without comment and presented her with the second cup...
>
> The man with the Vandyke beard was Ronald Aylmer Fisher, who was in his late thirties at the time. He would later be knighted Sir Ronald Fisher. In 1935, he wrote a book entitled *The Design of Experiments*, and he described the experiment of the lady tasting tea in the second chapter of that book. In his book, Fisher discusses the lady and her belief as a hypothetical problem. He considers the various ways in which an experiment might be designed to determine if she could tell the difference. The problem in designing the experiment is that, if she is given a single cup of tea, she has a 50 percent chance of guessing correctly which infusion was used, even if she cannot tell the difference. If she is given two cups of tea, she still might guess correctly. In fact, if she knew that the two cups of tea were each made with a different infusion, one guess could be completely right (or completely wrong).
>
> Similarly, even if she could tell the difference, there is some chance that she might have made a mistake, that one of the cups was not mixed as well or that the infusion was made when the tea was not hot enough. She might be presented with a series of ten cups and correctly identify only nine of them, even if she could tell the difference.
>
> In his book, Fisher discusses the various possible outcomes of such an experiment. He

> describes how to decide how many cups should be presented and in what order and how much to tell the lady about the order of presentations. He works out the probabilities of different outcomes, depending upon whether the lady is or is not correct. Nowhere in this discussion does he indicate that such an experiment was ever run. Nor does he describe the outcome of an actual experiment.

It's amazing to me that there is no actual evidence that such an event actually took place. This problem is a great introduction into hypothesis testing though and we can proceed by testing to see how likely it is for a person to guess correctly, say, 9 out of 10 times assuming that that person is just guessing. In other words, is the person just lucky or do we have reason to suspect that they can actually detect whether milk was put in first or not?

Recall that we need to think about this problem from the standpoint of hypothesis testing. We need to identify some important parts of a hypothesis test before we proceed with the analysis.

- Q: What does "by chance" mean in this context?
  A: We are assuming that the person is just guessing. This corresponds to them being equally likely to guess whether or not milk was entered first, i.e., the probability of "success" is 0.5.

- Q: What is our observed statistic?
  A: It isn't ever given just how many times out of 10 trials the supposed lady guessed correctly. We assumed above that the person got 9 out of 10 correct. This corresponds to our sample statistic of $\hat{p} = 0.9$.

- Q: What is the statistic trying to estimate?
  A: The statistic provides one guess as to a value of the parameter. We denote this parameter by the Greek letter π and here it corresponds to the long-run probability that the person will correctly guess whether milk or tea was added first, if the experiment was repeated many times.

- Q: How do we test to see whether the person is just guessing or if they have some special talent of identifying milk before tea or vice versa?
  A: Let's begin with an experiment. I will flip a coin 10 times. Your job is to try to predict the sequence of my 10 flips. Write down 10 H's and T's corresponding to your predictions. We will compare your guesses with my actual flips and then we will note how many correct guesses each of you have.

You may be asking yourself how this models a way to test whether the person was just guessing or not. All we are trying to do is see how likely it is to have 9 matches out of 10 if the person was truly guessing. When we say "truly guessing" we are assuming that we have a 50/50 chance of guessing correctly. This can be modeled using a coin flip and then seeing whether we guessed correctly for each of the coin flips. If we guessed correctly, we can think of that as a "success."

We often don't have time to do the physical flipping over and over again and we'd like to be able to do more than just 20 different simulations or so. Luckily, we can use `R` to simulate this process many times. The `mosaic` package includes a function called `rflip()` which can be used to flip one coin. Well, not exactly. It uses pseudorandom number generation to "flip" a virtual coin. Let's see an example of this: (Remember to load the `mosaic` package!)

```
> do(1) * rflip(1)

  n heads tails prop
1 1     1     0    1
```

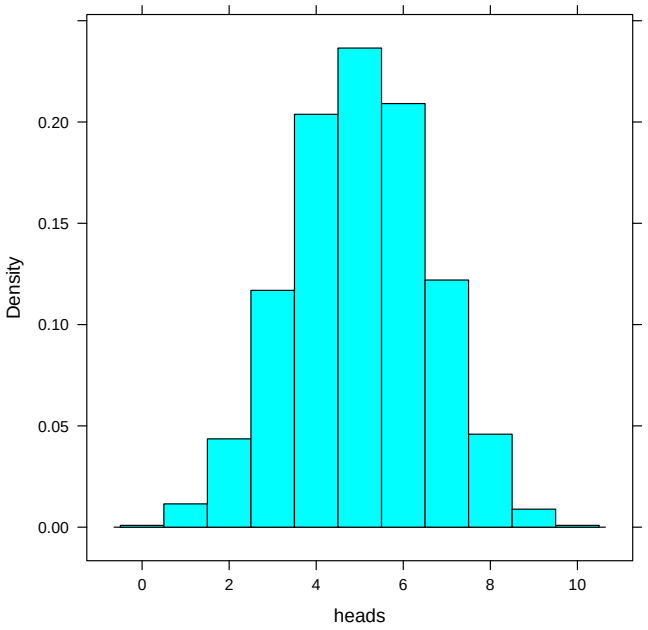
This shows us the proportion of "successes" in one flip of a coin. The `do()` function will be useful and you can begin to understand what it does with another example.

```
> do(13) * rflip(10)

    n heads tails prop
1  10     3     7  0.3
2  10     6     4  0.6
3  10     4     6  0.4
4  10     4     6  0.4
5  10     5     5  0.5
6  10     3     7  0.3
7  10     4     6  0.4
8  10     5     5  0.5
9  10     6     4  0.6
10 10     6     4  0.6
11 10     3     7  0.3
12 10     6     4  0.6
13 10     5     5  0.5
```

We've now done a simulation of what actually happened in class. We have 13 different simulations of flipping a coin 10 times. Note here that `heads` now corresponds to the number of correct guesses and `tails` corresponds to the number of incorrect guesses. (This can be tricky to understand at first since we've done a switch on what the meaning of "heads" and "tails" are.)

If you look at the output above for our simulation of 13 student guesses, we can begin to get a sense for what an "expected" sample proportion of successes may be. Around five out of 10 seems to be the most likely value. What does this say about our assumed $\hat{p}$ of 9/10? To better answer this question, we can simulate 10,000 student guesses and then look at the distribution of the simulated sample proportion of successes, also known as the null distribution.

```
> simGuesses <- do(10000) * rflip(10)
> tally(~heads, data=simGuesses, format='perc')

    0     1     2     3     4     5     6     7     8     9    10
 0.09  1.15  4.36 11.69 20.38 23.65 20.91 12.20  4.59  0.89  0.09

> histogram(~heads, data=simGuesses, width=1, center=0)
```



Density makes sense here since it gives the relative frequency of observing the different number of heads in the 10,000 simulations. We see that 5 heads occurs about 25% of the time and 4 and 6 heads about 20% each. Remember that our goal is to determine how likely it is for 9 heads (or more) to occur, assuming that the person is just guessing. This value is called the *p-value* and is given in the definition below.

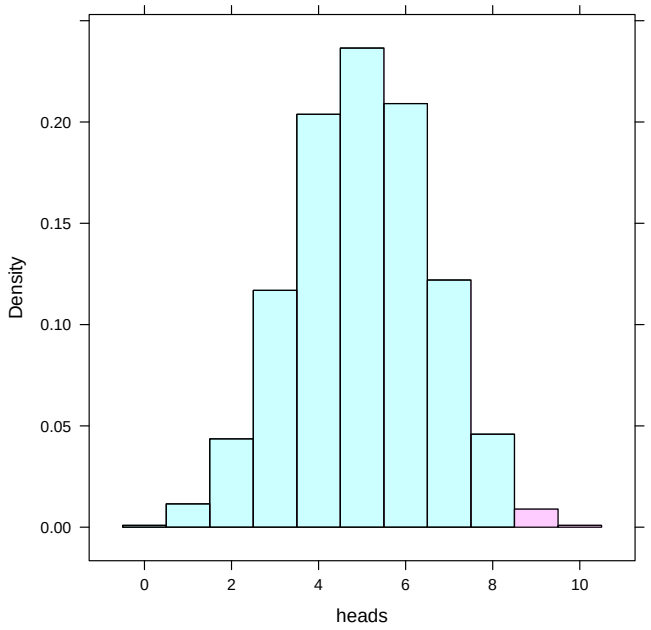
Definition 3.1. *The **p-value** is the probability of observing a sample statistic as extreme or more extreme than what was observed, assuming that the null hypothesis of a by chance operation is true.*

We can use another neat feature of `R` to calculate the p-value for this problem. Remember that "more extreme" in this case corresponds to looking at values of 9 or greater since our alternative hypothesis invokes a right-tail test corresponding to a "greater than" hypothesis of $H_a : \pi > 0.5$.

```
> pval <- prop(~heads >= 9, data=simGuesses); pval

  TRUE
0.0098
```

We can see that the observed statistic of 9 correct guesses is not a likely outcome assuming the null hypothesis is true. Only around 1% of the outcomes in our 10,000 simulations fall at or above 9 successes. We have evidence supporting the conclusion that the person is actually better than just guessing at random at determining whether milk has been added first or not. To better visualize this we can also make use of pink shading on the histogram corresponding to the p-value:

```
> histogram(~heads, groups = (heads >= 9), data=simGuesses, width=1, center=0)
```



3.3 Inference using Simulation on a Mean

Just as we did in the previous section when making hypotheses about a population proportion with which we would like to test which one is more plausible, we can also use simulation to infer conclusions about a population quantitative statistic such as the mean. In this case, we will focus on constructing confidence intervals to produce plausible values for a population mean. (We could do a similar analysis for a population median or other summary measure as well.)

Traditionally, the way to construct confidence intervals for a mean is to assume a normal distribution for the population or to invoke the Central Limit Theorem and get, what often appears to be magic, results. These methods are often not intuitive, especially for those that lack a strong mathematical background. They also come with their fair share of assumptions and often turn Statistics, a field that is full of tons of useful applications to many different fields and disciplines, into a robotic procedural-based topic. It doesn't have to be that way!

In this section, we will introduce the concept of *bootstrapping*. It will be a useful tool that will allow us to estimate the variability of our statistic from sample to sample. One neat feature of bootstrapping is that it enables us to approximate the sampling distribution and estimate the distribution's standard deviation using ONLY the information in the one selected (original) sample.

It sounds just as plagued with the magical type qualities of traditional theory-based inference on initial glance but we will see that it provides an intuitive and useful way to make inferences, especially when the samples are of medium to large size. We will begin by investigating an example on the selling prices of used Ford Mustang cars taken from the textbook *Statistics: UnLOCKing the Power of Data* by Lock, Lock, Lock, Lock, and Lock. (That isn't me hitting Copy+Paste too many times. Patti and Robin Lock both work in the Mathematics Department at St. Lawrence University and they have three children that are statisticians. They are often referred to as the Lock5.)

Example 3.1. A student collected data on the selling prices for a sample of used Mustang cars being offered for sale at an internet website. The price (in $1,000's), age (in years) and miles driven (in 1,000's) for the 25 cars in the sample. Use these data to construct a 90% confidence interval for the mean price (in $1,000's) of used Mustangs.

Solution: First, we need to get ahold of the dataset so we can begin to explore it a bit. I have copied it into a CSV file in my Public Folder on `RStudio` Server and it can be accessed and stored in your `R` environment via the following command:

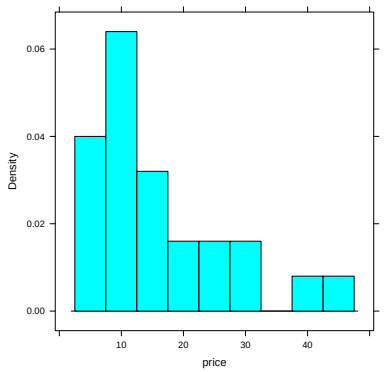
```
> mustangCars <- read.csv("/shared/ismayc@ripon.edu/mustang.csv")
```

We can get a preview of the `mustangCars` data and also summarize the different variables in the data set using the functions below.

```
> head(mustangCars)
  age miles price
1   6   8.5  32.0
2   7  33.0  45.0
3   9  82.8  11.9
4   2   7.0  24.8
5   3  23.0  22.0
6  15 111.0  10.0
> summary(mustangCars)
      age            miles            price
 Min.   : 1.00   Min.   :  1.10   Min.   : 3.00
 1st Qu.: 5.00   1st Qu.: 26.40   1st Qu.: 8.20
 Median : 9.00   Median : 71.80   Median :11.90
 Mean   : 8.32   Mean   : 66.34   Mean   :15.98
 3rd Qu.:12.00   3rd Qu.:100.80   3rd Qu.:22.00
 Max.   :15.00   Max.   :144.90   Max.   :45.00
```

The `summary()` function produces an easy way to get a snapshot of the summary quantities of the dataset of interest in a way similar to that of the `favstats()` command in `mosaic`. Following what we did in the last chapter it will also be helpful to look at the shape of the `price` variable. It is a quantitative variable so we will use the `histogram()` function to look at the distribution.

```
> histogram(~price, data=mustangCars, center = 10, width = 5)
```



We can see that the distribution is skewed to the right and far from symmetric. Further, our sample size is 25 which may or may not be large enough for the Central Limit Theorem to apply well. We don't have to worry though. We will use bootstrapping to create a distribution of simulated statistics.

It's important to remember what our goal is here. We want to try to understand the variability in the sample mean from sample to sample if we were able to take samples of size 25 over and over again from the population data on Ford Mustangs. The idea behind *bootstrapping* is to sample with replacement from the original sample to create new *resamples* of the same size as our original sample.

Returning to our example, let's investigate what one such resample of the `mustangCars` dataset accomplishes. We can create one resample/bootstrap sample by using the `resample()` function.

```
> resample(mustangCars)
     age miles price orig.ids
12    14 144.9   3.0       12
6     15 111.0  10.0        6
9      1  26.1  23.0        9
15     5  38.5  14.9       15
11     4  18.2  32.5       11
25    14 115.1   4.9       25
3      9  82.8  11.9        3
18     1  26.4  21.0       18
2      7  33.0  45.0        2
11.1   4  18.2  32.5       11
18.1   1  26.4  21.0       18
14    10  51.4  13.0       14
7     10 136.2   5.0        7
17     6  71.2  16.0       17
11.2   4  18.2  32.5       11
23    13  71.8  11.8       23
8      9  78.2   9.0        8
25.1  14 115.1   4.9       25
20    14 102.0   8.2       20
10     1   1.1  37.9       10
21    10  86.4   9.7       21
24    12  72.9  12.9       24
21.1  10  86.4   9.7       21
1      6   8.5  32.0        1
8.1    9  78.2   9.0        8
```

The important thing to note here is the case numbers in the far left column. Since we are sampling with replacement, there is a strong likelihood that some of the 25 observational units are going to be selected again. Each time this occurs the case number is appended with a period and the number of the repetition. You should see many ".1" and possibly some ".2" values if some elements were selected more than once.

You may be asking yourself what does this mean and how to this lead us to creating a distribution for the sample mean. Recall that the original sample mean of our data was calculated using the `summary()` function above. As a refresher, we can calculate it using the `mean()` function that is in the same form as the `histogram()` function above.

```
> mean(~price, data=mustangCars)
[1] 15.98
```

We can also look at the mean of a bootstrap sample created as well with a slight modification. Note the subtle change here!

```
> mean(~price, data=resample(mustangCars))
[1] 15.556
```

More than likely the calculated bootstrap sample mean is different than the original sample mean. This is what I meant earlier when I said that the sample means have some variability. What we are trying to do is replicate many different samples being taken from a larger population. Our best guess at what the population looks like is multiple copies of the sample we collected. We then can sample from that larger "created" population by generating bootstrap samples.

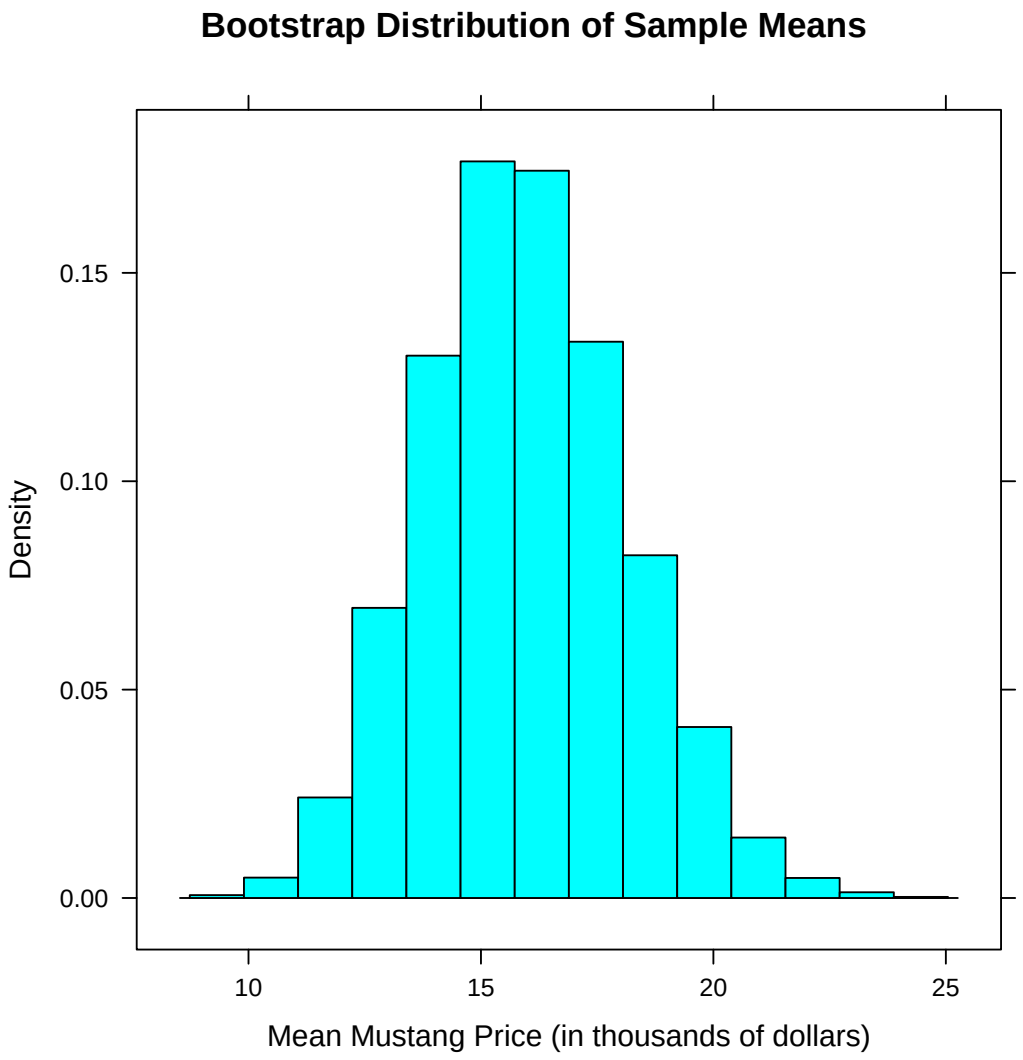
Similar to what we did in the previous section, we can repeat this process using the `do()` function followed by an asterisk. Let's look at 10 different bootstrap means for the `price` variable in `mustangCars`.

```
> do(10) * mean(~price, data=resample(mustangCars))
   result
1  18.000
2  11.948
3  16.432
4  19.260
5  21.872
6  18.032
7  16.108
8  21.016
9  16.452
10 15.608
```

You should see some variability begin to tease its way out here. Many of the simulated means will be close to our original sample mean but many will stray pretty far away. This occurs because outliers may have been selected a couple of times in the resampling or small values were selected more than larger. There are myriad reasons why this might be the case.

So what's the next step now? Just as we repeated the repetitions thousands of times with the "Lady Tasting Tea" example, we can do a similar thing here.

```
> trials <- do(10000) * mean(~price, data=resample(mustangCars))
> histogram(~result, data=trials,
+           main="Bootstrap Distribution of Sample Means",
+           xlab="Mean Mustang Price (in thousands of dollars)")
```



We can see from the distribution above that even with 10,000 replications we still see some remnants of the right-skewedness that we witnessed in the original dataset. This provides a good example of when the normal distribution (and/or its sibling t distribution) would likely overestimate or underestimate probabilities.

At this point, we can easily calculate a confidence interval. In fact, we have a couple different options. We will first use the percentiles of the distribution we just created to isolate the middle 90% of values. This will correspond to our 90% confidence interval for the population mean mustang price, denoted by μ, in thousands of dollars.

```
> confint(trials, level = 0.9, method = "quantile")
    name lower   upper level   method
1 result 12.58 19.6922   0.9 quantile
```

It's always important at this point to interpret the results of this confidence interval calculation. In this context, we can say something like the following:

> Based on the sample data and bootstrapping techniques, we can be 90% confident that the true mean Ford Mustang Price is between \$12,580 and \$19,692 dollars.

We can also get an idea of how the theory-based inference techniques would have approximated this confidence interval by using the formula

$$\bar{x} \pm (2 * SE),$$

where $\bar{x}$ is our original sample mean and SE corresponds to the standard deviation of the bootstrap distribution. This formula assumes that the bootstrap distribution is symmetric in a way similar to that done with the theory-based tests and confidence intervals. This is often the case with bootstrap distributions, especially those in which the original distribution of the sample is not highly skewed.

To compute this type of confidence interval, we only need to make a slight modification to the `confint()` function seen above. (Recall that the expression after the $\pm$ sign is known as the *margin of error*.)

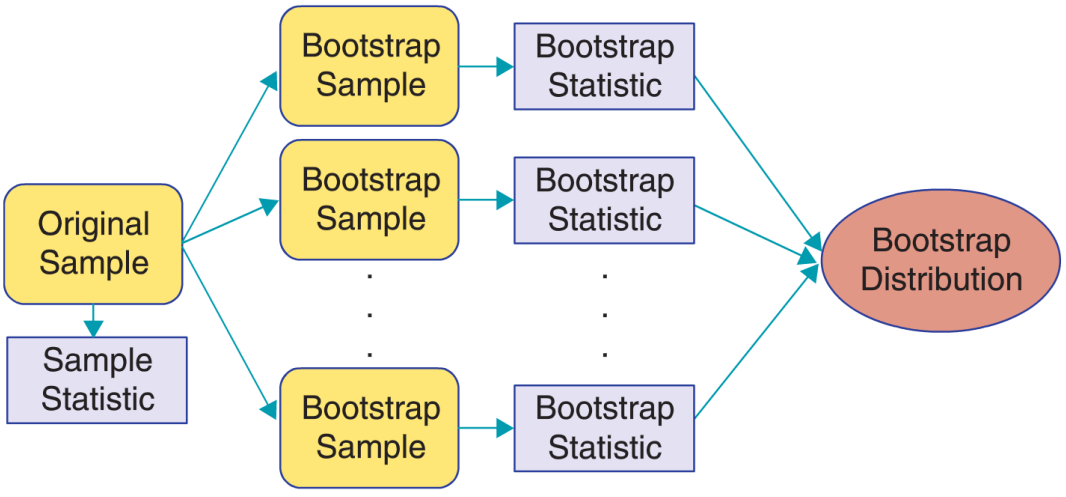
```
> confint(trials, level = 0.9, method = "stderr")
    name    lower    upper level method estimate margin.of.error
1 result 12.36807 19.52131   0.9 stderr 15.94469       3.576618
```

3.3.1 Review of Bootstrapping

We can summarize the process to generate a bootstrap distribution here in a series of steps (given by the aforementioned Lock5's in their textbook) that clearly identify the terminology we will use.

- Generate *bootstrap samples* by sampling with replacement from the original sample, using the same sample size.
- Compute the statistic of interest, called a *bootstrap statistic*, for each of the bootstrap samples.
- Collect the statistics for many bootstrap samples to create a *bootstrap distribution*.

Visually, we can represent this process in the following diagram.



3.4 Conducting Hypothesis Tests and Calculating Confidence Intervals Comparing Two Population Means

In the last two sections of this chapter, we have investigated hypothesis testing on a single proportion and developed methods for determining confidence intervals for a single mean. Often we are interested in understanding how to use data collected to understand the relationship between a quantitative variable over different levels of a categorical variable. We've already made conditional plots to get a better idea of the distributions in these cases. We will now investigate ways to determine, for example, whether a *treatment* has an effect over a *control* and other ways to statistically analyze if one group performs better than, worse than, or different than another. We will also use confidence intervals to determine the size of the effect if it exists.

Example 3.2 (Smiles and Leniency)**.**

© Cameron Whitman/iStockphoto

A neutral expression and a smiling expression: Which student gets the harsher punishment?

Research has shown that a smiling person is judged to be more attractive, sociable, and even competent than a non-smiling person. (Ever been around a salesperson who is constantly smiling? There may be reasons why they are doing this, though hopefully it is just because they are actually happy.) Two researchers, Marianne LeFrance and Marvin Hecht, in 1995 set forth with studying the effect of smiling on a judicial process.

Can a simple smile have an effect on punishment assigned following an infraction? The researchers conducted a study examining the effect of a smile on the leniency of disciplinary action for wrongdoers. Participants in the experiment took on the role of members of a college disciplinary panel judging students accused of cheating. For each suspect, along with a description of the offense, a picture was provided with either a smile or neutral facial expression. A leniency score was calculated based on the disciplinary decisions made by the participants. This data is available

in a CSV file in my Public folder.

```
> smileLen <- read.csv("/shared/ismayc@ripon.edu/smileLeniency.csv")
```

The researchers wanted to test to see if the average lenience score is higher for smiling students than it is for students with a neutral facial expression (or, in other words, that smiling students are given more leniency and milder punishments.)

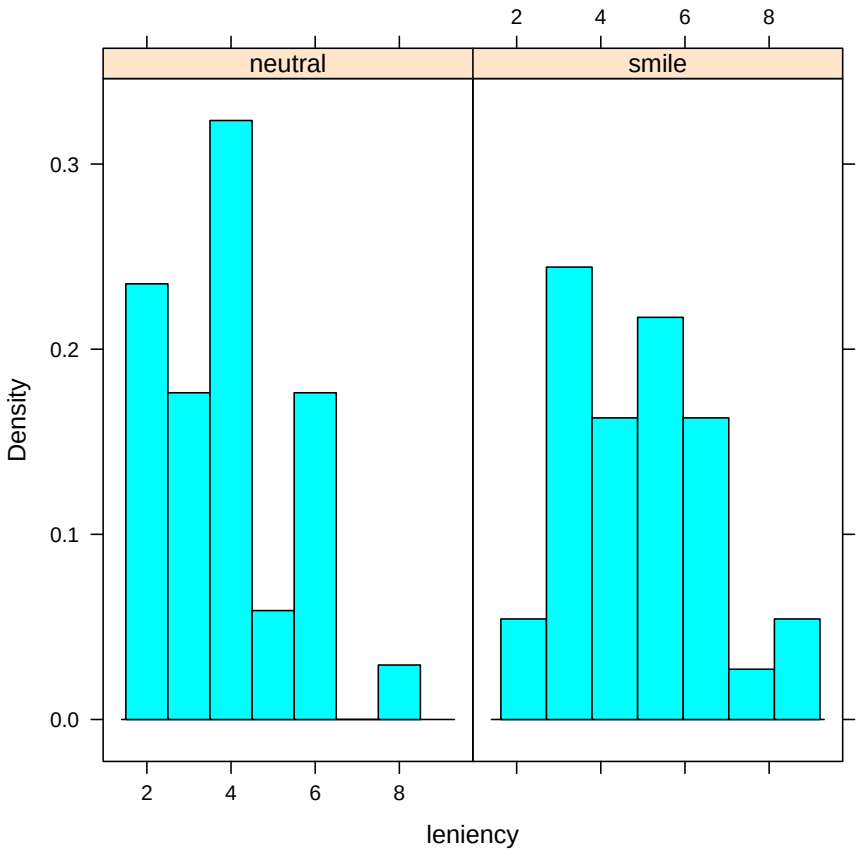
Q: In testing whether smiling increases leniency, define the relevant parameters and state null and alternative hypotheses.

A: We are comparing two means in this test, so the relevant parameters are μ_s, the true mean score for all smiling students, and μ_n, the true mean score for all neutral students. We are testing to see if there is evidence that the average leniency score is higher for smiling students, so the alternative hypothesis is $\mu_s > \mu_n$. The null hypothesis is that facial expression has no effect on the punishment given, so we assume the two means are equal:

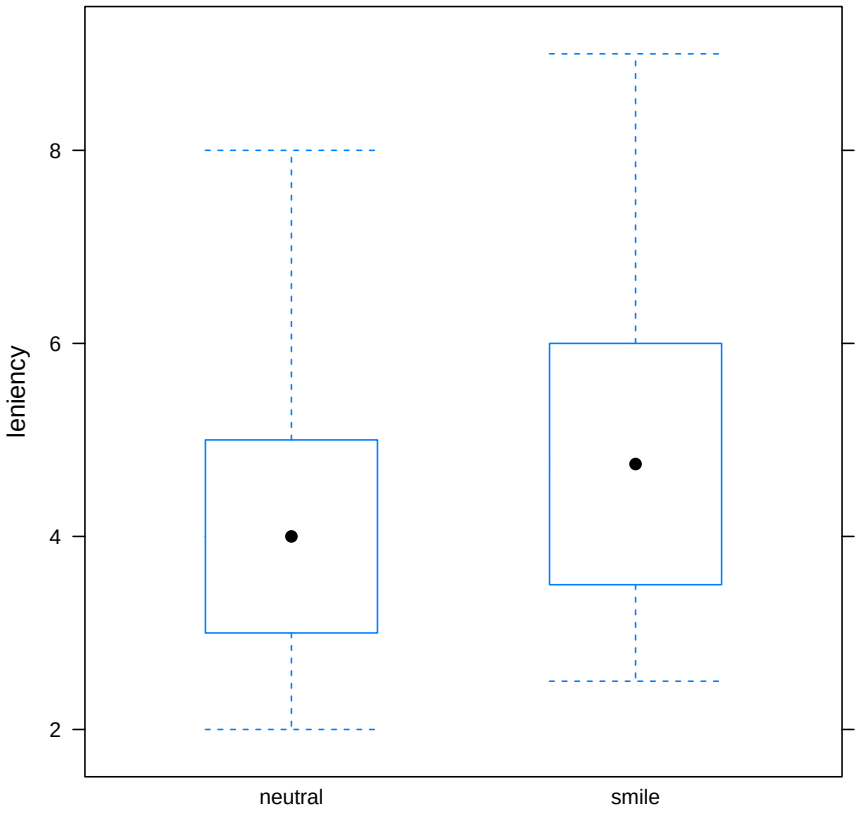
$$H_0 : \ \mu_s = \mu_n$$
$$H_a : \ \mu_s > \mu_n$$

It's important to get a feeling for what the data looks like before we conduct any type of inference. First, we will plot the quantitative variable of interest conditioned on the different levels of the qualitative variable here.

```
> histogram( ~ leniency | expression, data=smileLen)
```



```
> bwplot( leniency ~ expression, data=smileLen)
```



Do we have reason to believe, based on the distributions of `leniency` scores over these two groups, that there is a significant difference between the mean leniency score for neutral faces compared to smiling faces? It's hard to say just based on the plots. The two groups have differently shaped distributions but they are both over similar values of `leniency`. It's often useful to calculate the mean and standard deviation as well conditioned on the two levels.

```
> mean( ~ leniency | expression, data=smileLen)
 neutral    smile
4.117647 4.911765
> sd( ~ leniency | expression, data=smileLen)
 neutral    smile
1.522850 1.680866
```

We see that the sample mean leniency for those smiling, $\bar{x}_s$, is greater than the similar measure for those with neutral faces, $\bar{x}_n$. But is it statistically significantly greater? The standard deviation can provide some insight here but with these standard deviations being so similar it's still hard to say for sure.

The hypotheses we specified can also be written in another form to better give us an idea of what we will be simulating to create our null distribution.

$$H_0 : \ \mu_s - \mu_n = 0$$
$$H_a : \ \mu_s - \mu_n > 0$$

We are, therefore, interested in seeing whether the difference in the sample means, $\bar{x}_s - \bar{x}_n$, is statistically greater than 0. Note that order matters here in where you place the smiling mean and the neutral mean. R has a built-in command that can calculate the difference in these two sample means. (We will modify this slightly as we did in the previous two sections when we simulate.)

```
> diff(mean(leniency ~ expression, data = smileLen))
    smile
0.7941176
```

We will now proceed similarly to how we conducted the hypothesis in the second section of this chapter. We can look at this from a tactile point of view by using index cards. There are $n_s = 34$ data elements corresponding to those subjects that were smiling and $n_n = 34$ for those that had a neutral face. We can write the 34 leniency scores for smiling on one set of index cards and the 34 leniency scores for neutral on another set of index cards.

The next step is to put the two stacks of index cards together, creating a new set of 68 cards. If we assume that the two population means are equal, we are saying that there is no association between leniency scores and whether or not a subject was smiling. We can use the index cards to create two new stacks for those smiling and those with neutral faces. First, we must shuffle all the cards thoroughly. After doing so, in this case with equal values of sample sizes, we split the deck in half.

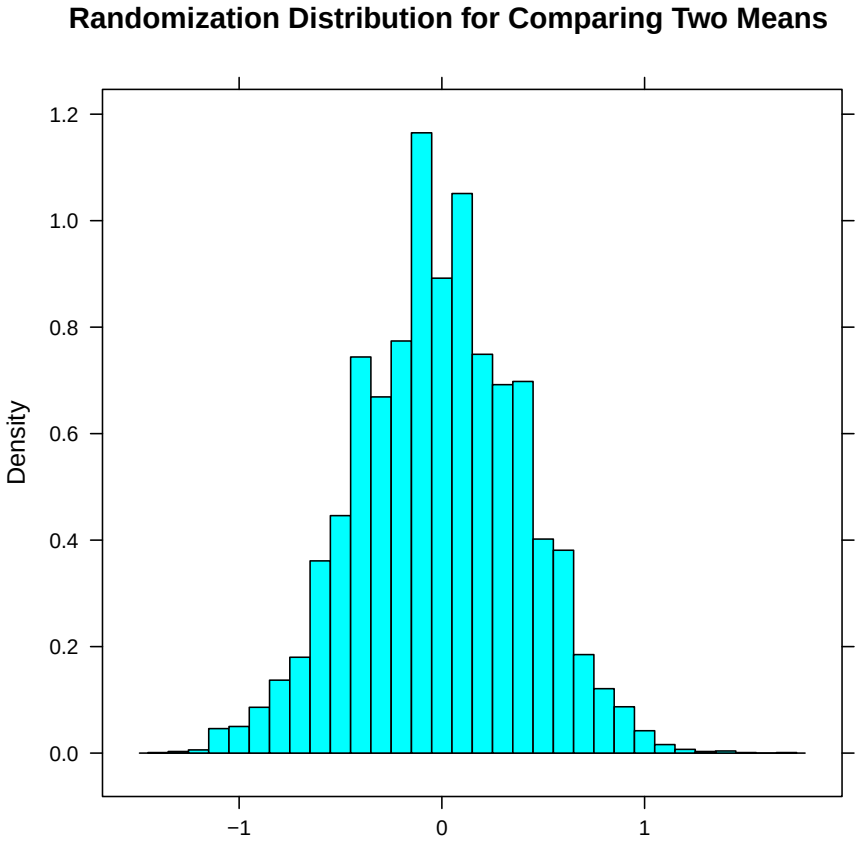
We then calculate the new sample mean leniency score of the smiling deck, and also the new sample mean leniency score of the neutral deck. This creates one simulation of the samples. We next want to calculate a statistic from these two samples. Instead of actually doing the calculation using index cards, we can use R as we have before to simulate this process.

```
> diff(mean(shuffle(leniency) ~ expression, data = smileLen))
     smile
0.02941176
```

The only new command here is `shuffle()` which does what we would expect it to do. It simulates a shuffling of the leniency scores between the two levels of `expression` just as we could have done with index cards. We can now proceed in a similar way to what we have done previously in this chapter. Using the `head()` function here tells us that the simulation in R puts the results into a variable called `smile`. This is why we choose to create a histogram on this variable.

```
> smileSim <- do(10000) * diff(mean(shuffle(leniency) ~ expression,
+                                   data = smileLen))
> head(smileSim)
       smile
1  0.1176471
2  0.2941176
```

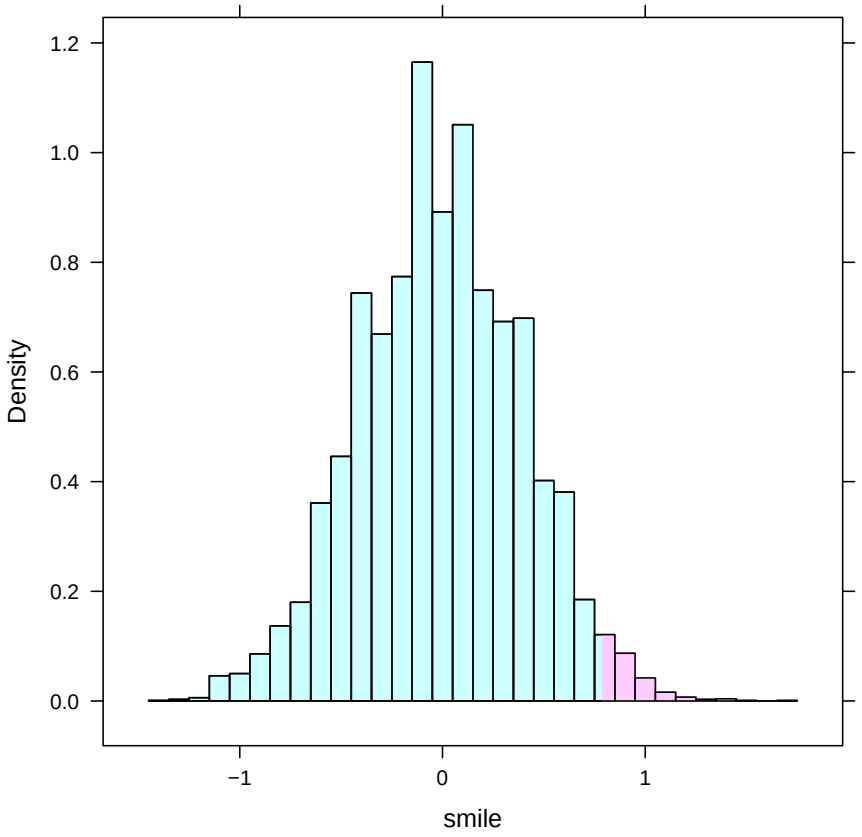
```
3 -0.2058824
4  0.4411765
5  0.3823529
6 -0.2647059
> histogram( ~smile, data=smileSim, width=0.1, center=0,
+            main="Randomization Distribution for Comparing Two Means",
+            xlab="Simulated Differences in the Mean Leniency Scores")
```



So we have now simulated this shuffling of the leniency scores and calculating of the resulting difference in sample means 10,000 times. Remember that the goal of all of this is determine if what we observed in the difference in the two sample means is "far away" from what we would expect if the null hypothesis is true.

Thus, we want to calculate how many values on our resulting histogram are as extreme or more extreme than the value of 0.794 that we observed. Remember that "more extreme" refers to the direction of the alternative hypothesis. We want to count how many of the 10,000 simulated differences are greater than or equal to our observed statistic of 0.7941176. After determining how many of these there are, we can divide by 10,000 to get our p-value.

```
> pval <- prop(~smile >= 0.7941176, data=smileSim); pval
  TRUE
0.0233
> histogram(~smile, groups = (smile >= 0.7941176), data=smileSim,
+                         width=0.1, center=0)
```



With this small p-value, we have evidence supporting the conclusion that people with smiling faces tend to receive more lenient punishment than those with neutral faces. The next important idea is to better understand just how much more lenient is punishment for those that smile compared to those with neutral faces. This can be addressed by creating a 95% confidence interval as follows.

We could use bootstrapping in a way similar to that done in the last section, except now on a difference in sample means, to create a distribution and then use the `confint()` function with the option of `quantile` to determine a confidence interval for the plausible values of the difference in population means. With our null distribution quite symmetric, the standard error method likely provides a good estimate. Another nice option here is that we can use the standard deviation of the null/randomization distribution we just found with our hypothesis test.

```
> sdSmile <- sd(~smile, data=smileSim); sdSmile
[1] 0.3998702
```

Remembering that we can use the general formula of $statistic \pm (2 * SE)$ we get the following result for plausible values of the difference in population means.

```
> lower <- 0.7941176 - (2*sdSmile); lower
[1] -0.005622842
```

```
> upper <- 0.7941176 + (2*sdSmile); upper
[1] 1.593858
```

Thus, we are 95% confident that the difference in true mean leniency scores for smiling individuals over neutral faced individuals is between -0.00562284232269838 and 1.5938580423227. This means that the mean leniency score could be as much as 0.00562284232269838 units lower for smiling individuals up to 1.5938580423227 units higher.

The important thing to check here is whether 0 is contained in the confidence interval. (It just barely is here...) If it is, it is plausible that the difference in the two population means between the two groups is 0. This means that the null hypothesis is plausible. The results of the hypothesis test and the confidence interval should match, with only slight discrepancies due to using the standard error method as an approximation.

3.5 Exercises

Your solutions to these problems should be done in `RStudio` using an `RMarkdown` document and choosing the **Knit HTML** or **Knit PDF** option. Create a document that clearly labels each of the problems with their exercise number. Your document should include your `R` commands as well as the plots that were created by these commands. Make sure to also carefully answer the questions by discussing the output of the plots or the code wherever necessary.

3.1) Enter the following tiny data set into an Excel file or Google Spreadsheet, save the file in CSV format to your computer, and import it into `RStudio` Server.

number	letter
42	A
654	D
7	A
52	A
4	D
88	B
223	B
6	A
92	D

- Store the data to a variable in `R` and output the data.
- After importing calculate the mean of the `number` variable.
- Create a bar graph of the `letter` variable.

3.2) A recent study (Ackerman, Griskevicius, and Li, 2011) examined expressions of commitment between two partners in a committed romantic relationship. One aspect of the study involved 47 heterosexual couples who are part of an online pool of people willing to participate in surveys. These 47 couples were asked about which person was the first to say "I love you." For 7 of those couples, the two people disagreed about the answer to this question. But both people agreed for the other 40 couples, so those 40 responses were included in the analysis. Previous studies have suggested that males tend to say "I love you" first.

a) Identify the observational units in this study.

b) Identify the variable of interest in this study. **Also classify the variable as categorical or quantitative.**

c) State the parameter of interest in words.

d) State the appropriate null and alternative hypotheses (in words) for testing whether males are more likely to say "I love you" first.

e) State the appropriate null and alternative hypotheses (in symbols) for testing whether males are more likely to say "I love you" first.

f) It turned out that for 28 of the 40 couples in the sample (after the 7 couples who could not agree were excluded), the man said "I love you" before the woman did. Determine the sample proportion (written as a decimal) of couples for whom the man was the first to say "I love you." What symbol do we use to denote this proportion?

g) Describe how you could use a coin-flipping model to find the p-value for these data. (Explain this using actual coins not the simulated coins in `R`.)

h) Conduct a simulation analysis using `R` similar to that done in this chapter. Provide relevant histograms with appropriate shading.

i) What is the value of the p-value for this problem?

j) Interpret the p-value for this problem using the definition of the p-value. (Note that this DOES NOT mean to state the conclusion of the test. You'll do that in the next part.)

k) Summarize your conclusion from this p-value in terms of the support for the alternative hypothesis.

3.3) A recent article (Hauser, Glynn, and Wood, 2007) described a study that investigated whether rhesus monkeys have some ability to understand gestures made by humans. In one part of the study, the experimenter approached individual rhesus monkeys and placed 2 boxes an equal distance from the monkey. The experimenter then placed food in one of the boxes, making sure that the monkey could tell that one of the boxes received food without revealing which one. Finally, the researcher made eye contact with the monkey and then gestured toward the box with the food by jerking his head toward that box. This process was repeated for a total of 40 rhesus monkeys. It turned out that 30 of the monkeys approached the box that the human had gestured toward, and 10 approached the other box. The purpose is to investigate whether rhesus monkeys can interpret the head jerk better than random chance.

a) Identify the observational units and variable in this study. Also classify the variable as categorical or quantitative.

b) Describe in words the parameter of interest for this study, and assign a symbol to it.

c) Determine the sample proportion of monkeys who picked the box towards which the human had gestured. Is this value a parameter or a statistic? What is the symbol you should use to denote this proportion?

d) State the appropriate null and alternative hypotheses in the context of this study, first in words, and then in symbols.

e) Describe how you could use a coin to conduct a simulation analysis of this study and its result. Give sufficient detail that someone else could implement this simulation analysis based on your description. Be sure to indicate how you would decide whether the observed data provide convincing evidence that rhesus monkeys can interpret human gestures better than random chance.

f) Use `R` to conduct a simulation analysis with at least 10,000 repetitions.

g) Based on the null distribution produced in `R`, explain how you are deciding whether the observed data provide convincing evidence that rhesus monkeys can read human gestures better than random chance.

h) Summarize the conclusion that you would draw about the research question of whether rhesus monkeys have some ability to understand gestures made by humans.

3.4) State whether the quantity described is a parameter or a statistic and give the correct notation.

- Average household income for all houses in the US, using data from the US Census.
- Average number of television sets per household in North Carolina, using data from a sample of 1000 households.

3.5) Give the correct notation for the quantity described and give its value.

- Average of enrollment in charter schools in Illinois. In 2010, there were 95 charter schools in the state of Illinois and the total number of students attending the charter schools was 30,795.
- Spread of the number of cell phone calls made or received per day by cell phone users. In a survey of 1917 cell phone users, the standard deviation was 13.10 phone calls a day.

3.6) Fish can be trained quite easily. With just seven days of training, golden shiner fish learn to pick a color (yellow or blue) to receive a treat, and the fish will swim to that color immediately. On the first day of training, however, it takes them more time. In a study described in *Science News*, the mean time for the fish in the study to reach the yellow mark is $\bar{x} = 51$ seconds with a standard error for this statistic of 2.4.

- Find and interpret an approximate 95% confidence interval for the mean time it takes a golden shiner fish to reach the yellow mark. (You don't need to use any functions in `R` here. You can (and SHOULD) use `R` as a calculator though for this problem.)
- Is it plausible that the average time it takes fish to find the mark is 60 seconds? Explain.
- Is it plausible that it is 55 seconds? Explain.

3.7) An original sample of $n = 6$ values are 17, 10, 15, 21, 13, and 18. Do the values given constitute a possible bootstrap sample from the original sample? Explain why or why not.

- 10, 12, 17, 18, 20, 21
- 10, 15, 17
- 10, 13, 15, 17, 18, 21
- 18, 13, 21, 17, 15, 13, 10
- 13, 10, 21, 10, 18, 17

3.8) How many ants will climb on a piece of a peanut butter sandwich left on the ground near an ant hill? To study this, a student in Australia left a piece of a sandwich for several minutes, then covered it with a jar and counted the number of ants. He did this eight times.

- Enter the following dataset into a CSV file in Excel in one column: $43, 59, 22, 25, 36, 47, 19, 21$. Make sure to add a header called `numAnts` in the first row. Then read this dataset into `R` using the **Upload** option.
- Find the mean and the standard deviation of the sample.
- Describe how we could use eight slips of paper to create one bootstrap statistic. Be specific.
- What is the population parameter of interest in the context of this problem?
- Find and interpret an approximate 95% bootstrap confidence interval (in TWO ways!) for the parameter defined in the previous part.

3.9) To create a confidence interval from a bootstrap distribution using percentiles, we keep the middle values and chop off some number of the lowest values and the highest values. If our bootstrap distribution contains values for 1000 bootstrap samples, indicate how many we chop off at each end for each confidence level given.

(a) 95% (b) 90% (c) 99% (d) 80%

3.10) Anchoring is "the common human tendency to rely too heavily," or 'anchor,' on one trait or piece of information when making decisions." (Source: *Wikipedia*.) A group of students taking an introductory statistics course at a four-year university in California were asked to guess the population of Milwaukee, Wisconsin. Some of the students were randomly chosen to be told that the nearby city of Chicago, Illinois has a population of about 3 million people, while the rest of the students were told that the nearby city of Green Bay, Wisconsin has a population of about 100 thousand. Previous studies have shown that these numbers serve as a psychological anchor, so people told about Chicago tend to guess a higher population for Milwaukee than people told about Green Bay. (For more about this phenomenon, see the book *Nudge: Improving Decisions about Health, Wealth, and Happiness*.) The purpose in analyzing the data is to see if we find strong evidence of this phenomenon among students like the ones in this study. Use the dataset in my Public folder called `Milwaukee.csv` for this problem.

a) Identify the observational units.

b) Identify the explanatory variable and the response variable in this study. Also classify each as categorical or quantitative.

c) Express the null and alternative hypotheses for testing whether the class data give strong evidence in support of the anchoring phenomenon described above. Use words only.

d) Express the null and alternative hypotheses for testing whether the class data give strong evidence in support of the anchoring phenomenon described above. Use symbols, and be sure to define completely any symbols used.

e) Give detailed, step-by-step instructions on how one could conduct a tactile (by hand) simulation to generate a p-value for testing the hypotheses stated. Be sure to include details on the following:

- Would the simulation involve coins, or dice, or index cards?
- How many tosses, or rolls, or cards would be used?
- How many sets of tosses, or rolls, or shuffles would you observe?

f) Create appropriate plot(s) and calculate summary information to get a sense of what to expect before conducting the inferential analysis.

g) Conduct a simulation analysis and determine an approximate p-value.

h) Interpret the p-value in the context of the study. That is, the p-value is a probability of *what* assuming *what*?

i) State your conclusion of your hypothesis test in the context of the study.

j) Also conduct and interpret an appropriate confidence interval for this problem.

Chapter 4

Simple Linear Regression

The remainder of the course will be focused on using models for statistical data analysis. You will see that many of the topics from introductory statistics that we have covered so far will be seen again as we begin to delve into the topic of regression, the form of statistical modeling we will explore further.

Statistical modeling can be used to answer many different types of questions. To give you some ideas, I've listed a few below:

- Can we distinguish among different species of eagles based solely on the length of their beaks?
- Do students with higher GPAs really have a better chance of succeeding in college by completing degrees?
- Are SAT and ACT scores good predictors of student success in their first year in college? How about long term financial success?
- Which is more strongly related to the average score for professional golfers: driving distance, driving accuracy, putting performance, or iron play? Are all of these useful for predicting a golfer's average score? Which are most useful? How much of the variability in golfers' scores can be explained by knowing all of these other values?

From a statistical point of view, these sorts of questions have many purposes. These include:

- Making predictions,
- Understanding relationships, and
- Assessing differences.

To the dismay of many, statistical models are just that: models. They are simplifications of what is actually happening in reality. Any textbook on statistical modeling is bound to provide the following quote by statistician George Box:

> All statistical models are wrong, but some are useful.

Statistical models are not deterministic, meaning that their predictions are not expected to be perfectly accurate in all circumstances. For example, we do not expect to predict the exact price of a used car based on its mileage. Even if we were to record every imaginable characteristic of the car and include them all in the model, we would still not be able to predict its price exactly. Statistical models merely aim to explain as much of the variability as possible in whatever phenomenon is being modeled. In fact, because human beings are notoriously variable and unpredictable, social scientists who develop statistical models are often delighted if the model explains even a small part of the variability.

A distinguishing feature of statistical models is that we pay close attention to possible simplifications and imperfections, seeking to quantify how much the model explains and how much it does not. So, while we do not expect our model's predictions to be exactly correct, we are able to state how confident we are that our predictions fall within a certain range of the truth. And while we do not expect to determine the exact relationship between two (or more) variables, we can quantify how far off our model is likely to be. And while we do not expect to assess exactly how much two (or more) groups differ, we can draw conclusions about how likely they are to differ and by what magnitude.

For the remainder of the course, we will focus on the following description of a statistical model:

$$DATA = MODEL + ERROR$$

which can be written mathematically as

$$Y = f(X) + \varepsilon.$$

The Y here represents the variable being modeled (the response variable), X is the variable(s) used to do the modeling (the predictor/explanatory variable(s)), and f is a function. In this chapter we will focus on just one quantitative response variable Y, one quantitative explanatory variable X, and a linear function f. In later chapters, we will consider more complicated functions for f and also different combinations of multiple categorical/numerical predictors and a qualitative/quantitative response.

The remaining term, ε, is the error term. It corresponds to the part of the response variable Y that remains unexplained after considering the predictor variable(s) X. Often it is standard practice to assume that this error term follows a normal distribution. Throughout this course, we will focus on checking whether that assumption is valid based on the data we have.

4.1 Four-Step Process

We will employ a four-step process for statistical modeling throughout the course. This is laid out below.

- **Choose** a form for the model. This involves identifying the response and explanatory variable(s) and their types (categorical or quantitative). We usually examine graphical displays to help suggest a model that might summarize relationships between these variables.

- **Fit** that model to the data. This usually entails estimating model parameters based on the sample data. We will use R to do the necessary number-crunching to fit models to data instead of having you use a pencil-and-paper and/or a calculator.

- **Assess** how well the model describes the data. One component of this involves comparing the model to other models. Are there elements of the model that are not very helpful in explaining the relationships or do we need to consider a more complicated model? Another component of the assessment step concerns analyzing residuals, which are deviations between the actual data and the model's predictions, to assess how well the model fits the data. This process of assessing model adequacy is as much art as science.

- **Use** the model to address the question that motivated collecting the data in the first place. This might be to make predictions, or explain relationships, or assess differences, bearing in mind possible limitations on the scope of inferences that can be made. For example, if the data were collected as a random sample from a population, then inference can be extended to that population; if treatments were assigned at random to subjects, then a cause-and-effect relationship can be inferred; but if the data arose in other ways, then we have little statistical basis for drawing such conclusions.

4.2 Simple Linear Regression Model

In the rest of this chapter, we consider a single quantitative predictor X and a quantitative response variable Y. A common model to summarize the relationship between two quantitative variables is the *simple linear regression model*. We will now review the structure of this model, the estimation and interpretation of its parameters, the assessment of its fit, and its use in predicting values for the response. To conclude the chapter we will consider one method, called *transformation*, for dealing with relationships between two numerical variables that are not linear.

We will describe the Choose, Fit, Assess, Use process with an example on the prices of Porsche sports cars.

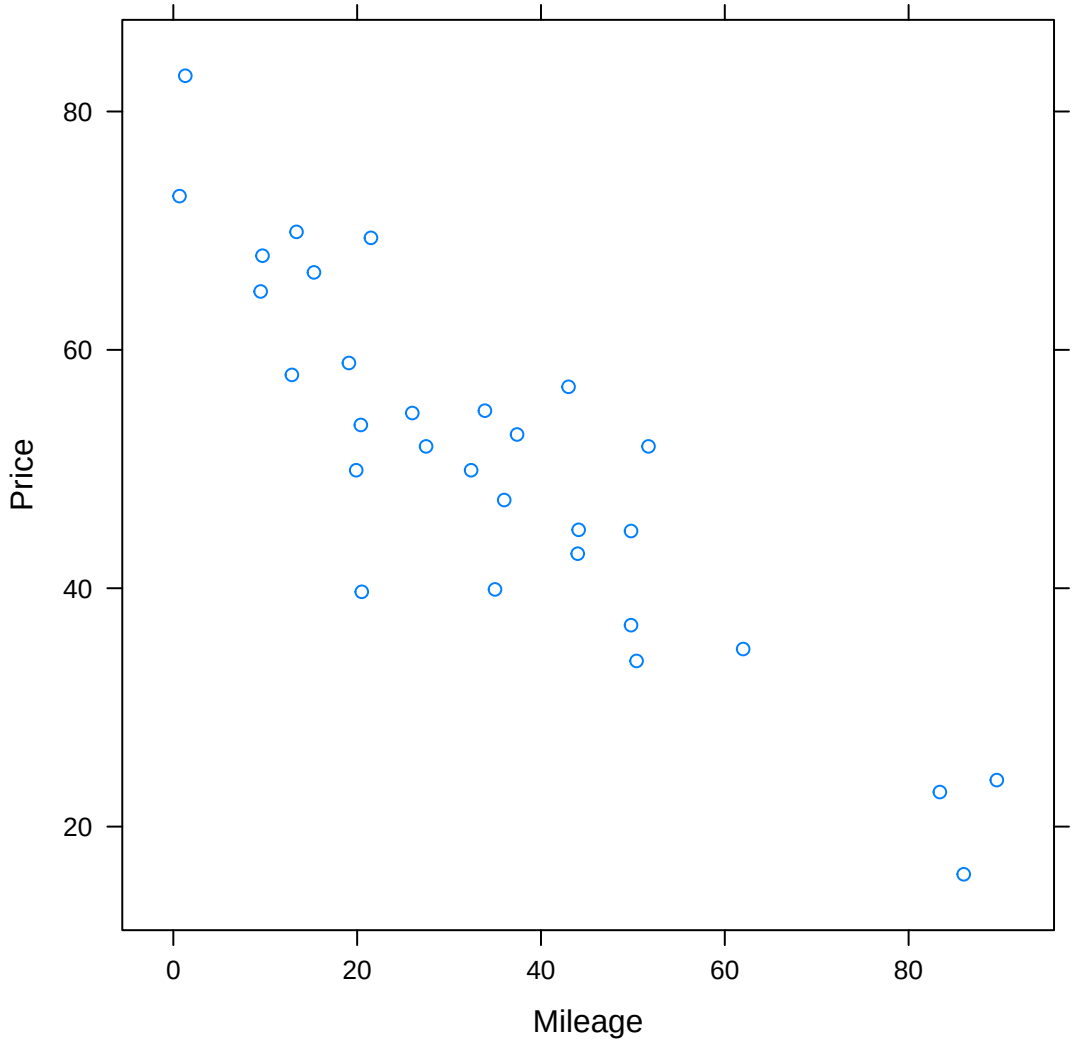
Example 4.1. Suppose that we really, really want a Porsche car but really, really can't afford a new one. Somehow we have scrounged up what we believe is enough money for a used Porsche. Do we have enough? The price of Porsche cars, and really all cars in general, might depend on many variables including the age, condition, and special features of the car. For this example, we will focus on the relationship between $X = Mileage$ of a used Porsche and $Y = Price$. *AutoTrader.com* was used to collect data on a sample of 30 used Porsches, with price (in thousands of dollars) and mileage (in thousands of miles).

The dataset including this information is available in my Public folder via the command

```
> porsche <- read.csv("/shared/ismayc@ripon.edu/porschePriceByMileage.csv")
```

Before we begin to determine which model to fit, it is important to begin by analyzing the graphical relationship between these two variables.

```
> xyplot(Price ~ Mileage, data=porsche)
```



We can see that we have a consistent negative linear trend in the scatterplot. In other words, as mileage increases, price tends to decrease in a constant linear fashion. Next, we can begin to lay out mathematically what the form of our model will be. Using the notation from earlier we have

$$Y = f(X) + \varepsilon = \mu_Y + \varepsilon.$$

We can also think of $f(X)$ as being a function that gives the mean value of Y, denoted μ_Y, at any value of X and that ε represents the error (deviation) from that mean. We will see that in situations like this one with both a quantitative predictor and response that scatterplots will be a major tool we will use to help us choose which type of model to fit.

Definition 4.1 (Simple Linear Regression (SLR) Model)**.**

$$\mu_Y = f(X) = \beta_0 + \beta_1 X$$

which implies $Y = \beta_0 + \beta_1 X + \varepsilon$ *defines a simple linear regression model where* $f(X)$ *is a linear function of* X *with intercept* β_0 *and slope* β_1*.*

Step 1: CHOOSE

Since the rate of decrease in the scatterplot is relatively constant as the mileage increases, a linear model might provide a good summary of the relationship between the average prices and mileages of used Porsches for sale on *AutoTrader.com*. In symbols, we express the mean price as a linear function of mileage:

$$\mu_{Price} = \beta_0 + \beta_1 \cdot Mileage.$$

Thus, the model for actual used Porsche prices would be

$$Price = \beta_0 + \beta_1 \cdot Mileage + \varepsilon.$$

This model indicates that Porsche prices should be scattered around a straight line with deviations from the line determined by the random error component, ε. We will now explore how to determine the slope and intercept for the line that best summarizes this relationship and reduces the deviations in some consistent way.

4.2.1 Fitting a Simple Linear Model

We want the best possible estimates of β_0 and β_1. Thus, we use least squares regression to fit the model to the data. Mathematically, this process is achieved through concepts from calculus. We could derive formulas for these estimates in general using algebra and calculus. That's all fine and dandy but I'm of the opinion that if the formulas exist and have been proven to be correct, we can let algorithms built in `R` actually determine these estimates for us.

The fitted model is represented by

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X.$$

In general, we use Greek letters (β_0, β_1, etc.) to denote parameters and hats ($\hat{\beta}_0$, $\hat{\beta}_1$, etc.) are added to denote estimated (fitted) values of these parameters. A key tool for fitting a model is to compare the values it predicts for the individual data cases to the actual values of the response variable in the dataset.

Definition 4.2 (Residual)**.** *The discrepancy in predicting each response is measured by*

$$residual = observed\ y - predicted\ y = y - \hat{y}.$$

The *sum of squared residuals/errors*, denoted by SSE, provides a measure of how well the line predicts the actual responses for a sample. `R` calculates the fitted values of the slope and intercept so as to minimize this sum of squared residuals; hence, we call this the *least squares line*.

Step 2: FIT

For the i^{th} car in the dataset, with mileage x_i, the model is

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon$$

with the parameters, β_0 and β_1 in the model, representing the true, population-wide intercept and slope for all Porsches for sale. The corresponding statistics, $\hat{\beta}_0$ and $\hat{\beta}_1$, are estimates derived from

this particular sample of 30 Porsches. We can use R to calculate these estimates for $\hat{\beta}_0$ and $\hat{\beta}_1$ via the following:

```
> lm(Price ~ Mileage, data=porsche)

Call:
lm(formula = Price ~ Mileage, data = porsche)

Coefficients:
(Intercept)      Mileage
    71.0905      -0.5894
```

Here `lm()` corresponds to "linear model" and, thus, our least squares line is

$$\hat{Price} = 71.09 - 0.5894 \cdot Mileage.$$

Often, the interpretation of the model is the most important part of the analysis since it can provide insight into whether the regression fit makes sense. Interpretation will be a main focus throughout the course so pay careful attention to how we do this for this example.

Thus, for every additional 1000 miles on a used Porsche, the predicted price goes down by about \$589. Also, if a (used!) Porsche had zero miles on it, we would predict the price to be \$71,090. In many cases, the intercept lies far from the data used to fit the model and has no practical interpretation.

```
> head(porsche)
  Price Age Mileage
1  69.4   3    21.5
2  56.9   3    43.0
3  49.9   2    19.9
4  47.4   4    36.0
5  42.9   4    44.0
6  36.9   6    49.8
```

The first car in the dataset had a mileage of 21,500 miles and a price of \$69,400. The fitted line predicts the price to be

$$\hat{Price}(21.5) = 71.09 - 0.5984 \cdot 21.5 = 58.4.$$

Therefore, the residual for this value of $X = Mileage$ is

$$Price - \hat{Price} = 69.4 - 58.4 = 11.0.$$

We can do a similar calculation for each of the 30 cars, square each of the resulting residuals, and sum the squares to get a resulting $SSE = 1439.6$. If you were to choose any other straight line to make predictions for these Porsche prices based on the mileages, you could never achieve an SSE less than 1439.6.

4.2.2 Model Conditions

In specifying a model, we must make sure that certain conditions and assumptions are met. A key part of assessing the fit of the model is to check whether the conditions are reasonable for the given data that we are working with. In an ideal world, the residuals are "small" and contain no distinguishable pattern. If they did, we would need to fix our model and try a different fit.

We will frequently use plots to give us an idea as to whether conditions have been met. In addition to plots, it is important to actually quantify whether the assumptions have been validated through further numerical testing. Many of the assumptions described below extend to more than just the current simple linear regression case and we will refer back to them frequently.

We have already referred to the first assumption and often it is the easiest to check. It will also be the one that we potentially deviate from when we are no longer in the SLR case.

1. Linearity - The relationship between the variables is of a linear form. The average values of the response Y for each value of X fall on a common straight line.

We also need to make assumptions about the error distribution. We will be able to better understand these conditions as the semester progresses:

2. Zero mean - The error distribution is centered at zero. (By construction of the residuals in least squares regression, this will always be the case. It may not be the case if other regression methods are used though.)

3. Constant variance - The variability in the errors is the same for all values of the predictor variable. In other words, the spread of the points in the scatterplot remains close to constant throughout the dataset.

4. Independence - The errors are assumed to be independent from one another. One point falling above or below the line has no influence on the location of any other point.

5. Normality - Often, we will assume that the errors follow a normal distribution. We will carefully check whether this condition is met. If it is, many of our analyses can be simplified to using the common normal-based distributions.

These conditions are summarized in the following definition. (Don't let the mathematics scare you yet!)

Definition 4.3 (Simple Linear Regression Model)**.** *For a quantitative response variable Y and a single quantitative explanatory variable X, the* **simple linear regression model** *is*

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

where ε is normally distributed with mean 0 and standard deviation σ_ε [$\varepsilon \sim N(0, \sigma_\varepsilon)$], and the errors are independent from one another.

4.2.3 Assessing Model Conditions

We have already discussed a variety of plots that we could produce to assess statistical properties of variables. For quantitative variables, we have focused on scatterplots and histograms. When assessing many of the more complicated assumptions about the distribution of the error terms, we will use plots of residuals versus fitted values and normal plots.

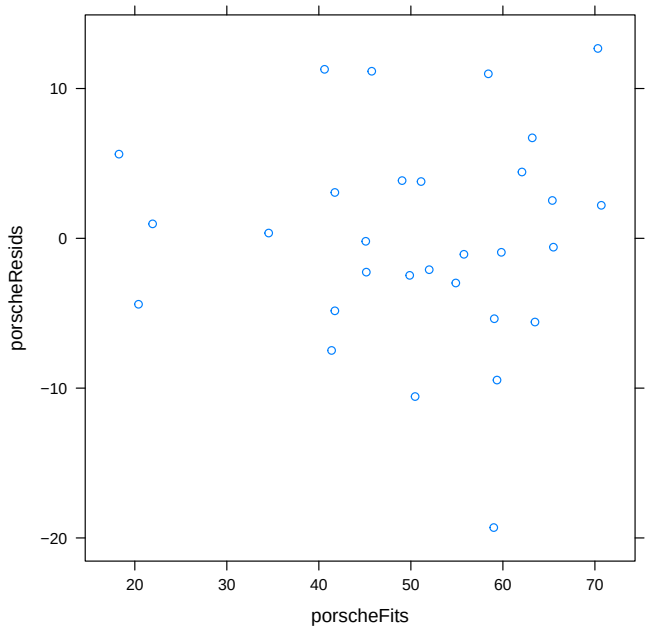
4.2.3.1 Residuals versus Fits Plots

A scatterplot with the fitted line provides one visual method of checking linearity. Points will be randomly scattered above and below the line when the linear model is appropriate. Clear patterns, such as clusters of points above and below the line in some systematic way, indicate that the linear model is not appropriate.

A more informative way of looking at how the points vary about the regression line is a scatterplot of the residuals versus the fitted values for the predicted equation. This plot shifts the axes so that the regression line is represented as a horizontal line through zero. Positive residuals represent points that are above the regression line and negative ones are below. We then look for clear patterns in the estimated errors (the residuals) for one check to see if our model is appropriate.

Going back to our linear regression model on Porsche prices, we can create this **Residuals versus Fits Plot**. This can be done in R using the following command. We will also add

```
> porscheFits <- fitted.values(lm(Price ~ Mileage, data=porsche))
> porscheResids <- residuals(lm(Price ~ Mileage, data=porsche))
> xyplot(porscheResids ~ porscheFits)
```



Remember that we are looking to see if the points are scattered about the horizontal line corresponding to the mean value of 0 for the residuals. This plot appears to show that there is no distinguishable pattern. In addition, we can check to see if the variance is constant throughout the plot. While it does appear to have a little more spread for higher fitted values, there isn't anything striking in this plot to have us reject our original assumptions. Think carefully about what sorts of residuals versus fits plots would lead you to believe that linearity and/or constant variance are not met. Often the plots only provide us with a general idea of whether patterns exist that contradict our assumptions of random errors and constant variance. We rarely will get perfectly textbook examples when we actually do real data analysis so it is important to remember that these plots can only provide us with suggestions of things to investigate further.

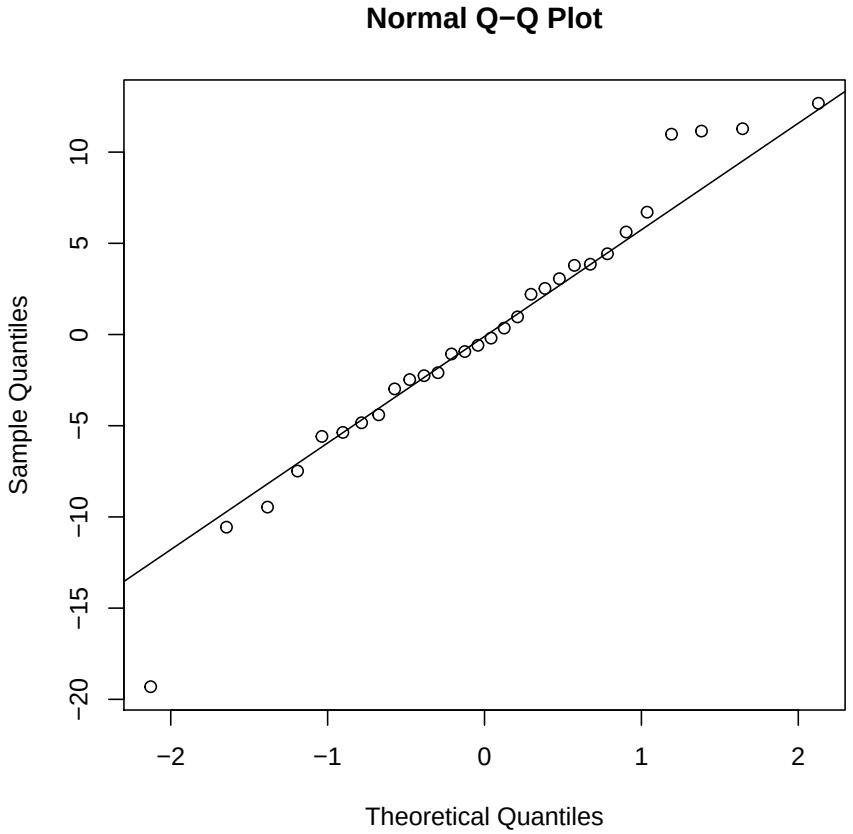
4.2.3.2 Normal Plots

In the last sub-subsection we discussed ways to use plots to assess the conditions of linearity and constant variance using a residuals versus fits plot. We can also use plots to check to see if the errors do follow a normal distribution. These types of plots are called "normal plots" and, they display data that has normal distribution properties as following a straight line. Most commonly, normal plots fall into either *normal quantile plots* or *normal probability plots*. We will focus on normal quantile plots at this time.

A *normal quantile plot* (Q-Q plot) is a scatterplot of the ordered observed data versus theoretical quantiles that we would expect to see from a "perfect" normal sample of the same size. If the ordered residuals are increasing at the rate we would expect to see for a normal sample, the resulting scatterplot is a straight line. If the distribution of the residuals is skewed in one direction or has tails that are overly long due to some extreme outliers at both ends of the distribution, the normal quantile plot will bend away from a straight line.

We will again shift back to our example of prices of used Porsche cars to see if the normality assumption is violated for our residuals.

```
> qqnorm(porscheResids)
> qqline(porscheResids)
```



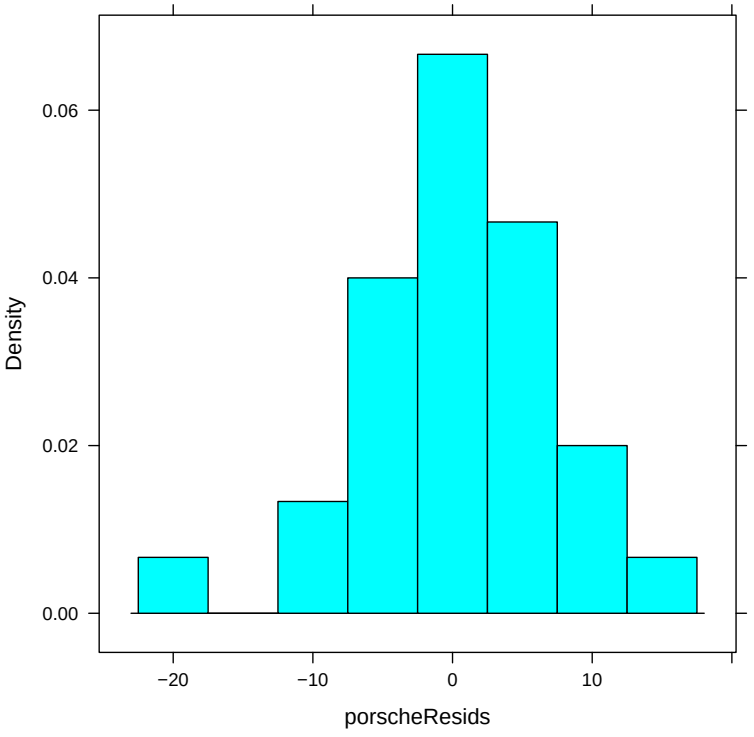
The two functions above produce a Q-Q plot and a reference line. We can see that there isn't much deviation from the straight line here. It does stray a little in the tails but nothing to really alarm as to large problems with our assumptions.

Step 3: ASSESS

We now summarize what we have found with our plots in assessing the model fit.

- *Linearity*: The linearity condition is reasonable based on our first scatterplot of the two variables. In addition, the plot of the residual values versus the fitted values showed that the points were scattered randomly above and below the reference horizontal line at 0. This illustrates that a linear model is appropriate for the relationship between price and mileage.
- *Zero mean*: We used least squares regression, which forces the sample mean of the residuals to be zero when estimating the intercept β_0. Also not that the residuals are scattered on either side of zero in the residual plot which further provides evidence that a zero mean for our error terms is plausible.
- *Constant variance*: The residuals versus fits plot shows that the data points deviate roughly the same from the reference line at zero so we have a reasonably constant spread in the estimated errors.
- *Independence and random*: The sample's random selection process and one seller likely not having a big impact on another seller leads us to suspect that these assumptions are valid.
- *Normality*: Our Q-Q plot also supports the assumption of the residuals being normally distributed since the residuals fall close to what would be expected for a perfectly normal sample. We can further see the normal nature of the residuals by using a histogram. This is often a good choice if we are working with a large dataset, which we don't necessarily have here.

```
> histogram(~porscheResids, center=0, width=5)
```



Step 4: USE

We decided in the last step of assessment that this model provides a reasonable fit to our data. We can now interpret the implications for the question of interest. It's always important to loop back to what the point of all of this statistical analysis was. For example, suppose we find a used Porsche for sale with 50,000 miles and we believe that it is from a population from which our sample of 30 used Porsches was drawn. (Think carefully about why this is important to assume!) What should we expect to pay for this car? Would it be an especially good deal if the owner was asking $38,000?

Based on our model, we would expect to pay $\hat{Price}(50) = 71.09 - 0.5894 \cdot 50 = 41.62$ or $41,620. The asking price of $38,000 is below this expected price, but is this difference large relative to the variability in Porsche prices? We might like to know if this is a really good deal or perhaps such a low price that we should be concerned about the condition of the car.

Definition 4.4 (Regression/Residual Standard Error)**.** *For a simple linear regression model, the estimated standard deviation of the error term based on the least squares fit to a sample of n observations is*

$$\hat{\sigma}_\varepsilon = \sqrt{\frac{SSE}{n-2}}.$$

For our Porsche problem, we can obtain the `Residual standard error` via the `summary()` command we have used before.

```
> summary(lm(formula = Price ~ Mileage, data = porsche))

Call:
lm(formula = Price ~ Mileage, data = porsche)

Residuals:
     Min       1Q   Median       3Q      Max
-19.3077  -4.0470  -0.3945   3.8374  12.6758

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 71.09045    2.36986    30.0  < 2e-16 ***
Mileage     -0.58940    0.05665   -10.4 3.98e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.17 on 28 degrees of freedom
Multiple R-squared:  0.7945,        Adjusted R-squared:  0.7872
F-statistic: 108.3 on 1 and 28 DF,  p-value: 3.982e-11
```

We can observe that the car's residual is about half of what we call a "typical error" (or the regression/residual standard error) below the expected price. Therefore, we likely don't have to be that concerned about this price based on our analysis since it falls within what we would expect for variability from the expected price.

4.3 Transformations

In an ideal world, the conditions are met perfectly for a simple linear regression and we can proceed with the analysis. That's not always the case though. One way to deal with the conditions not being met is to use transformations on one or both of the variables. We will better understand this via an example.

Example 4.2. Do you believe that the number of doctors and the number of hospitals in a given city would be related? Would they be positively or negatively correlated? Would fitting a linear model be appropriate? Given a number of hospitals in a city, can we predict how many doctors are in that city? To investigate these questions, we will use the `MetroHealth83` dataset available in my Public folder. This dataset consists of a sample of 83 metropolitan areas that have at least two community hospitals. It was collected by the U.S. Census Bureau in 2006.

```
> health <- read.csv("/shared/ismayc@ripon.edu/MetroHealth83.csv")
```

This dataset consists of 83 observational units and 16 variables. We can focus our attention on the variables in of interest via the following:

```
> healthSub <- select(health, City, NumMDs, NumHospitals)
> summary(healthSub)
```

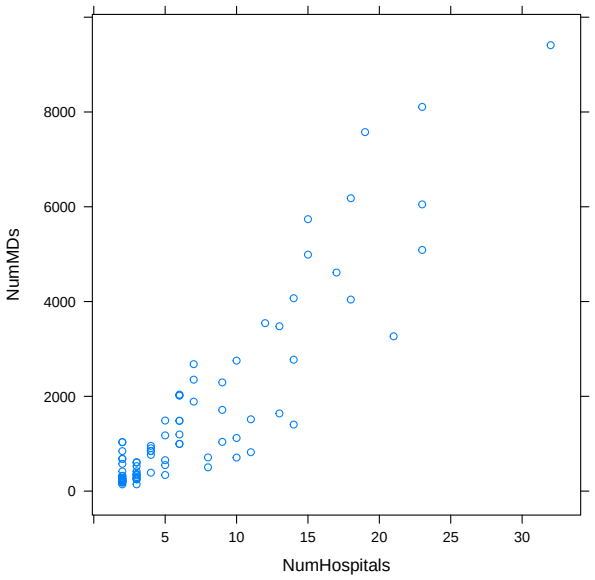
```
                City        NumMDs        NumHospitals
 Anderson, IN     : 2   Min.   : 143.0   Min.   : 2.000
 Binghamton, NY   : 2   1st Qu.: 336.5   1st Qu.: 2.000
 Charleston, WV   : 2   Median : 844.0   Median : 5.000
 Midland, TX      : 2   Mean   :1643.3   Mean   : 7.193
 Springfield, IL  : 2   3rd Qu.:2018.0   3rd Qu.:10.000
 Trenton-Ewing, NJ: 2   Max.   :9410.0   Max.   :32.000
 (Other)          :71
```

We will now work with this subset of the larger `health` dataset.

Step 1: CHOOSE

We begin by identifying the explanatory variable(s) and response variable. We are still working with simple linear regression so we will choose only one explanatory variable. It was given above that we are trying to find a model for predicting the number of doctors (MDs) using the number of hospitals. Our first step should then be to investigate the scatterplot `xyplot()` of the two variables below.

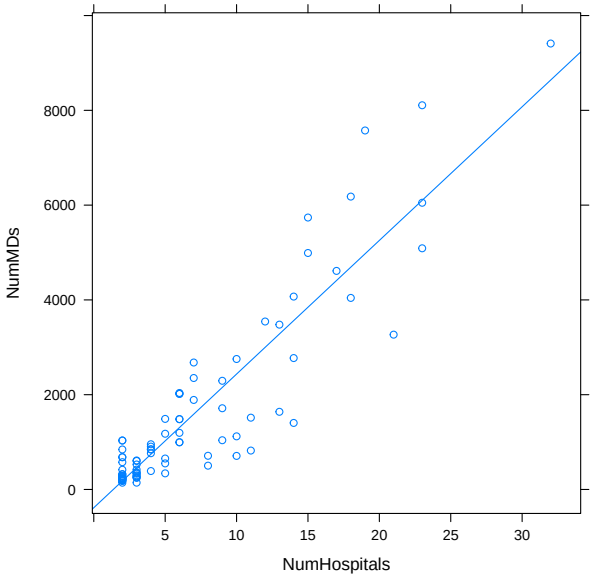
```
> xyplot(NumMDs ~ NumHospitals, data=healthSub)
```



Step 2: FIT

We can see a positive trend with cities having more hospitals also having more doctors. We can next fit a SLR model to this data and begin to assess the quality of the fit.

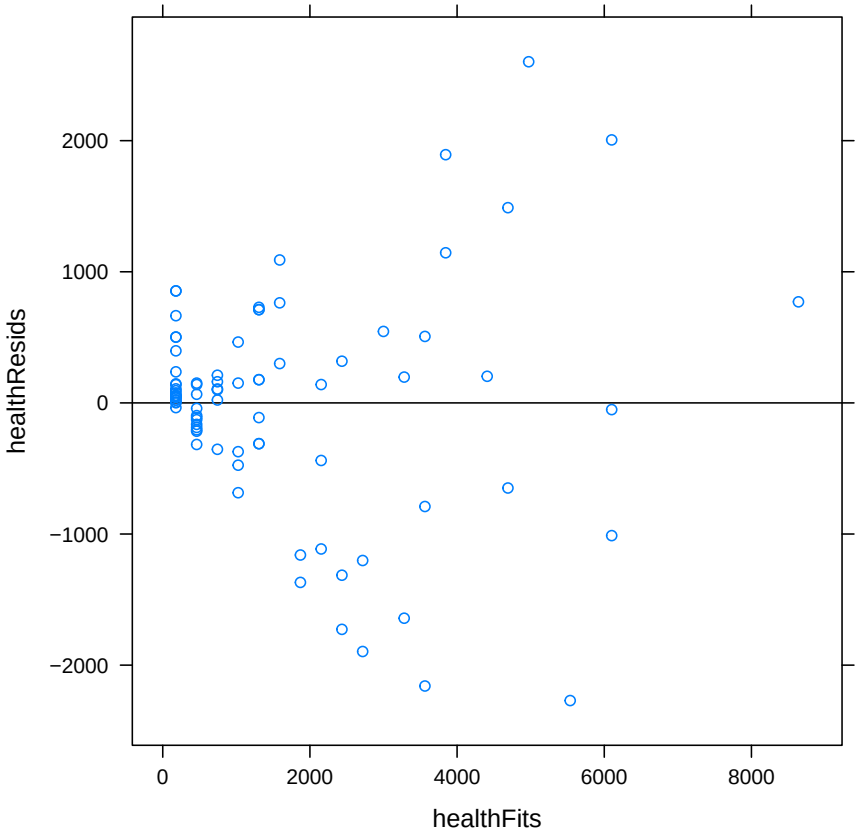
```
> healthModel <- lm(NumMDs ~ NumHospitals, data=healthSub); healthModel

Call:
lm(formula = NumMDs ~ NumHospitals, data = healthSub)

Coefficients:
 (Intercept)  NumHospitals
      -385.1         282.0
> xyplot(NumMDs ~ NumHospitals, data=healthSub,  type = c("p","r"))
```



We don't see any clear reason to not believe a linear model is appropriate here. It's important to not stop here though. We still need to investigate the residuals versus fits plot to check for constant variance and the Q-Q plot to check for normality of the residuals.

Step 3: ASSESS

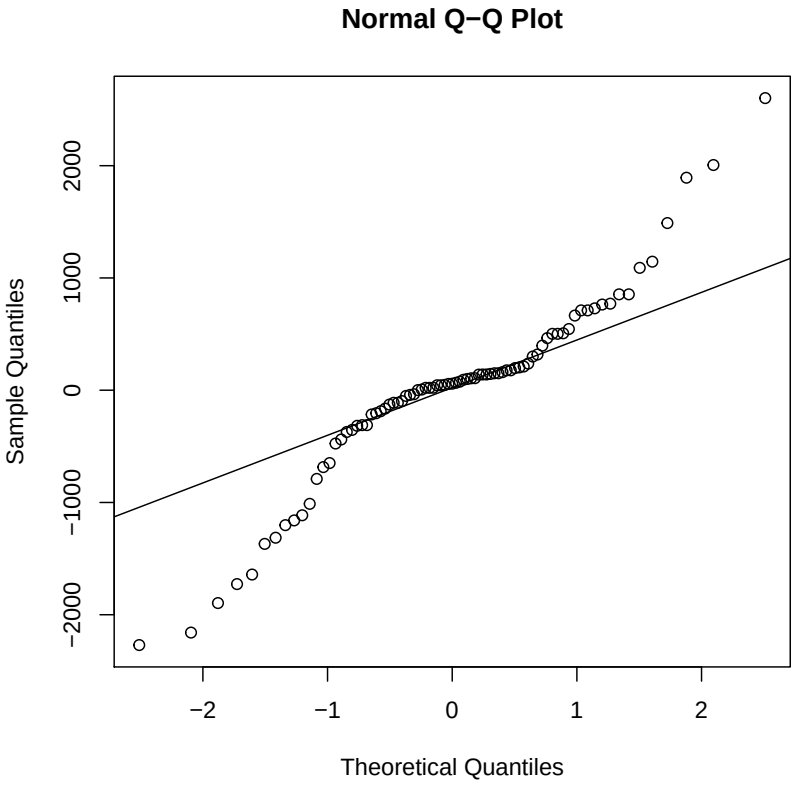
```
> healthFits <- fitted.values(healthModel)
> healthResids <- residuals(healthModel)
> xyplot(healthResids ~ healthFits, abline = c(0, 0))
```



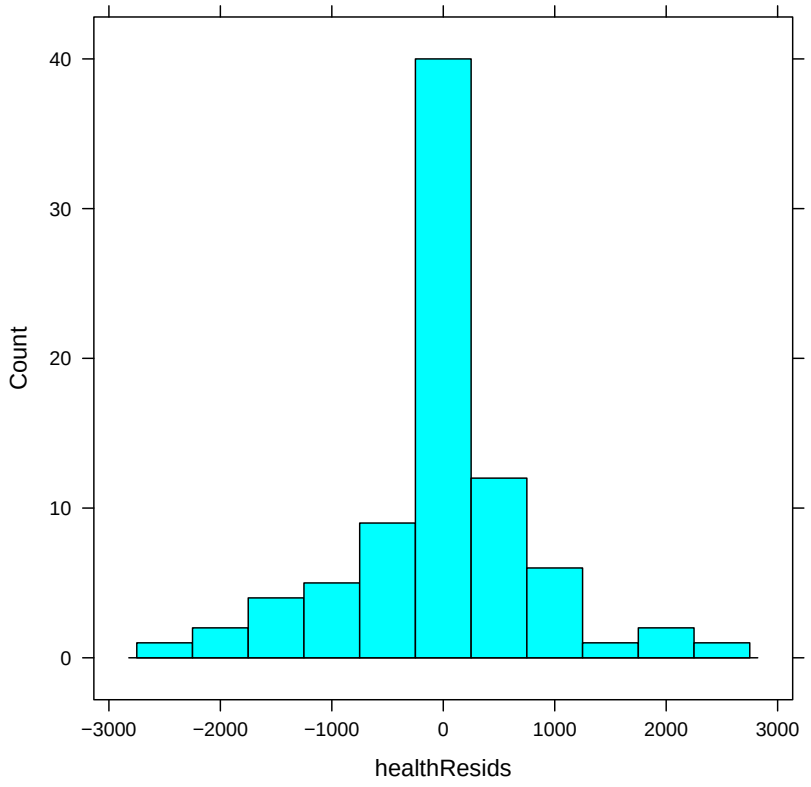
Notice the addition of the `abline = c(0, 0)` parameter which draws a line through the expected mean of 0 for the residuals. This plot does lead us to have a bit of concern with our fit since it is in the notorious "megaphone" shape, corresponding to larger values tending to have much more variability in their residuals than small values. We have reason to doubt the constant variance assumption required in the SLR model.

It's important to also investigate the normality assumption via a Q-Q plot and/or a histogram of the residuals.

```
> qqnorm(healthResids)
> qqline(healthResids)
```




```
> histogram(~healthResids, type='count', center=0, width=500)
```



The assumption of normality of the residuals is also violated here since there is plenty of departure from the straight line in both tails of the distribution in the Q-Q plot. The histogram also deviates from a normal curve since the peak is higher than expected and the tails drift off into either direction being much longer than we would expect.

To account for these issues, we can do a variety of things. One common choice for trying to stabilize the variance in a response variable across the different values of the predictor variable is to transform either Y or X. The usual options include raising a variable to a power (such as $\sqrt{Y}$, X^2, or $1/X$) or taking a logarithm (e.g., using $\log Y$ as the response).

Step 1: CHOOSE (again)

For this type of dataset which gives counts, a square root transformation on Y is often useful. You will get better at determining which type of transformation to use through experience. Finding an appropriate transformation is as much art as it is science. Again, this will be one of those subjective things that we will have to deal with throughout the course.

Step 2: FIT (again)

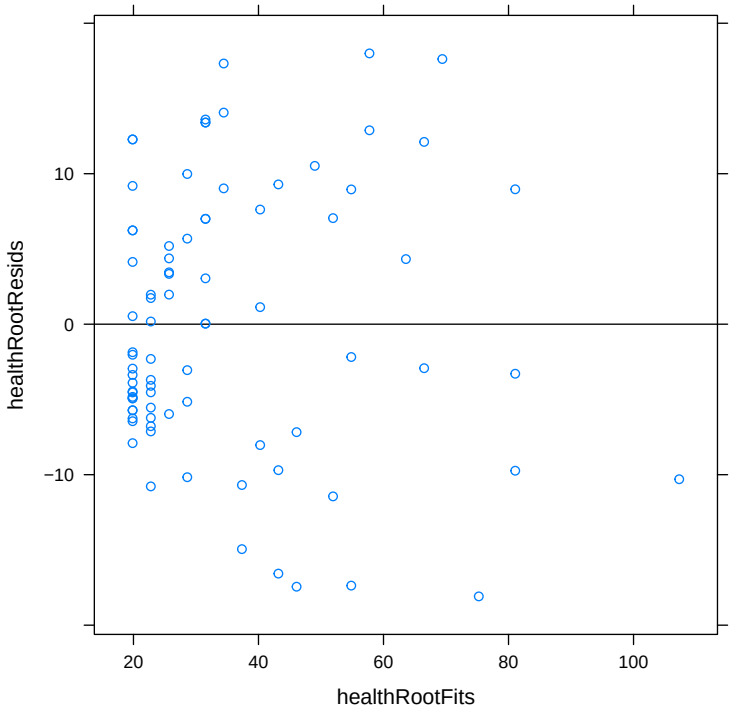
```
> healthRootModel <- lm(sqrt(NumMDs) ~ NumHospitals, data=healthSub)
> healthRootModel

Call:
lm(formula = sqrt(NumMDs) ~ NumHospitals, data = healthSub)

Coefficients:
 (Intercept)  NumHospitals
      14.033         2.915
```

This gives the following fitted equation: $\widehat{\sqrt{NumMDs}} = 14.033 + 2.915 \cdot NumHospitals$.

Step 3: ASSESS (again)

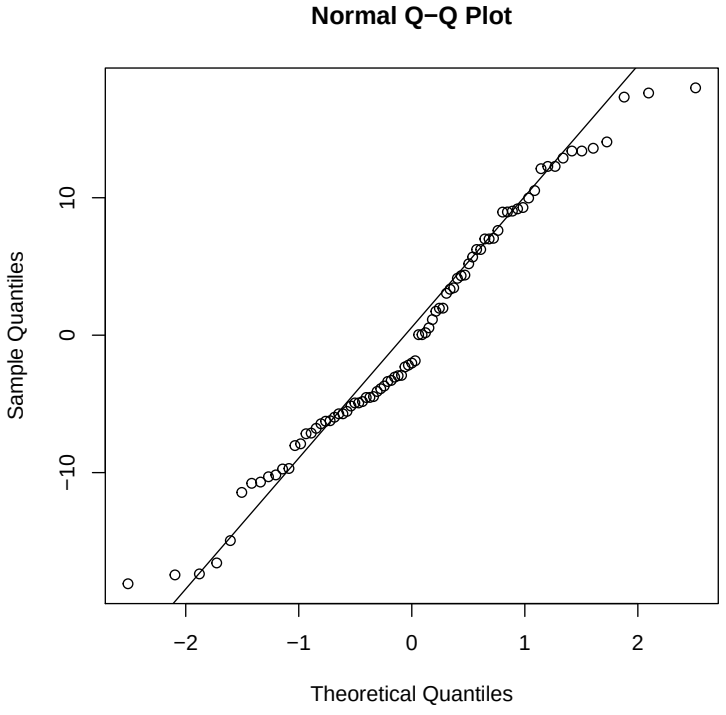
We now can check to see if this transformed model fits the data more appropriately than the SLR model. We will again use the residuals versus fits plot and the Q-Q plot.

```
> healthRootFits <- fitted.values(healthRootModel)
> healthRootResids <- residuals(healthRootModel)
> xyplot(healthRootResids ~ healthRootFits, abline = c(0, 0))
```



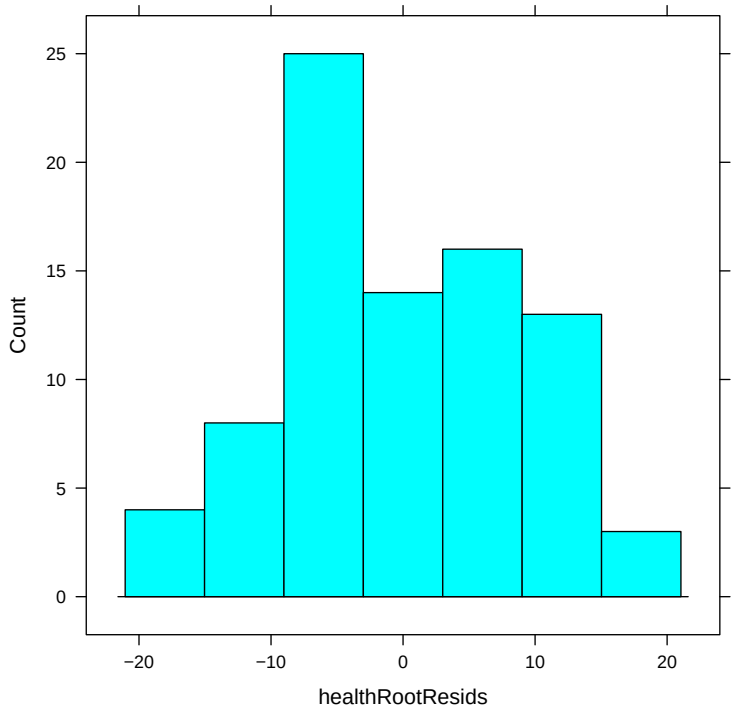
We see that the megaphone shape is no longer as visible as before. The variability is much more uniform throughout the transformed data.

```
> qqnorm(healthRootResids)
> qqline(healthRootResids)
```



The Q-Q plot produces a better fit to what we would expect from normally distributed residuals. Remember that the residuals will hardly ever fall perfectly on the line so the interpretation of the Q-Q plot is even more important. You will need to justify why you believe a model fits the data well and what the advantage of one model is over another.

```
> histogram(~healthRootResids, type='count', center=0)
```



This histogram appears a little more normal in shape but leaves something to be desired as well. We can say that the square root transformation does do a better job than the SLR model in this case. Transformation is often useful but may not produce perfect results.
Step 4: USE The last remaining step is crucial in understanding how the model works. Since we have transformed our model, we are now predicting the square root of the number of doctors instead of the actual number of doctors. ALWAYS stop to check before you proceed with calculations blindly. You will get very strange results if you forget that the data has been transformed.

Flagstaff, AZ corresponds to the 59th city listed in the `healthSub` dataset. They have 2 community hospitals. We can get their fitted value based on the square root model via the following command:

```
> healthRootFits[[59]]
[1] 19.86251
```

We can also use the `predict()` function to get these values. We must first assign the values we would like to use as predictors into a dataframe.

```
> new.data <- data.frame(NumHospitals=c(2))
> predict(healthRootModel, new.data)
       1
19.86251
```

We, therefore, have the following equation for the predicted square root of doctors for Flagstaff: $\sqrt{\hat{NumMDs}}(2) = 14.033 + 2.915 \cdot 2 = 19.863$. The predicted actual number of doctors is then given as

```
> (healthRootFits[[59]])^2
[1] 394.5192
```

Flagstaff had an actual number of doctors of 324 compared to this predicted value of around 395. We can use this model to make predictions for other cities, but we should be only comfortable doing so if we believe that the city chosen falls into the population with which the sample was selected.

4.4 Inference for Simple Linear Regression

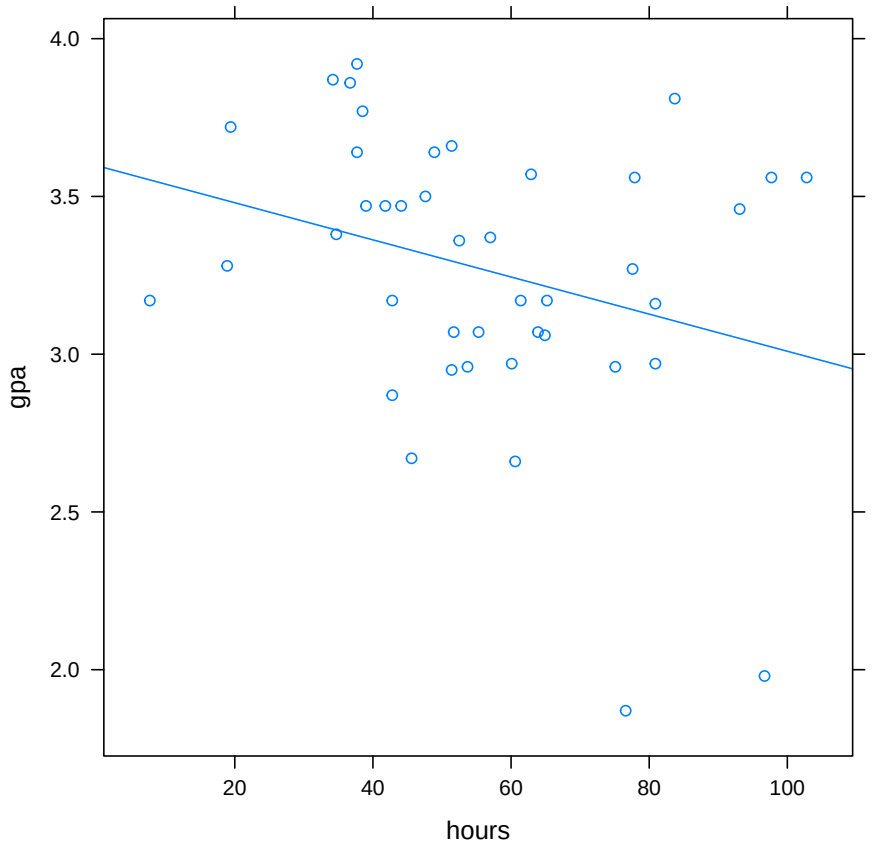
In addition to the different plots used to assess the conditions required in a simple linear regression model, we can also use statistical inferential techniques to test to see if a linear fit is reasonable. In what follows, we will use resampling techniques similar to those done in Section 3.4. We will be shuffling the response variable over all of the values of the explanatory variable. Then, we will calculate the different slopes and intercepts for each of these new simulated datasets. Lastly, we will test to see if what we saw in our original sample is statistically significant by evaluating its position in this simulated distribution. We will also check to see how these results compare with the theory-based methods that are traditionally given.

Example 4.3. College students are busy. They have lots of things expected of them. They have many courses that require them to study often, do required readings, and complete homework. In addition, colleges also offer opportunities for students to participate in a wide variety of activities. Sometimes, students seek out these chances and join as many clubs as they can. Also, many students also spend time working on a job on or off campus, watching TV, and socializing with other students. But is this good for them academically? Does spending all this extra time with non-academic activities hinder academic performance? Many studies have been conducted to test to see if this is the case.

We will look at results from a study carried out by undergraduate students at the University of Minnesota in the data set `hoursGPA.txt` in my Public folder. They surveyed 42 students and asked questions about time spent per week on various non-academic activities such as work, watching TV, exercising, and socializing. They calculated the total time spent on all these activities and looked at the relationship between that and the students' GPAs.

```
> extraHoursGPA <- read.file("/shared/ismayc@ripon.edu/hoursGPA.txt")
> head(extraHoursGPA)
  hours  gpa
1   7.7 3.17
2  18.9 3.28
3  19.4 3.72
4  34.7 3.38
5  42.8 3.17
6  42.8 2.87
```

```
> xyplot(gpa ~ hours, data=extraHoursGPA, type = c("p", "r"))
```



You can (and should on your own!) check that the conditions are met for linear regression here. We will proceed on the assumption that an SLR model is appropriate. We are now interested in checking to see if the slope from this regression model that we observed in this sample is statistically less than 0. In other words, we want to see if we have evidence that a negative linear relationship exists on `gpa` and `hours`. We begin by first determining what the sample slope is for this example.

```
> gpaModel <- lm(gpa ~ hours, data=extraHoursGPA)
> gpaModel

Call:
lm(formula = gpa ~ hours, data = extraHoursGPA)

Coefficients:
(Intercept)        hours
   3.597691    -0.005884
```

We can further isolate the coefficients of the fit using the command below.

```
> coef(gpaModel)
 (Intercept)        hours
3.597690950 -0.005883873
```

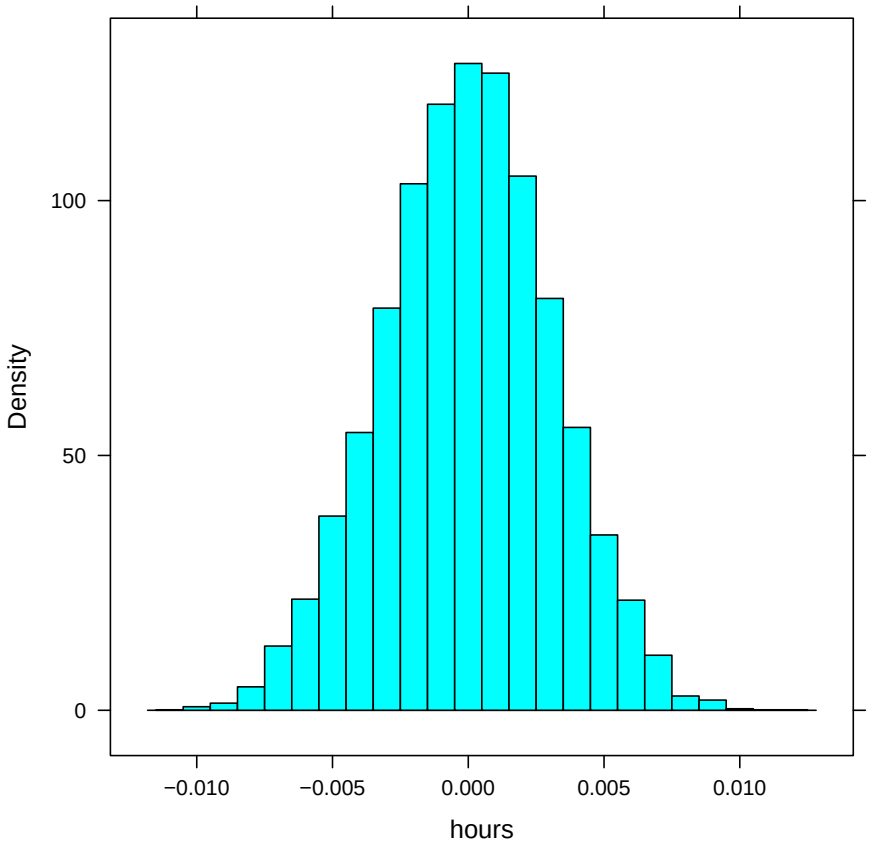
4.4.1 Simulation Approach

Our sample slope is -0.005884. Is this far enough from zero to conclude that this observed slope is significant? We will attempt to answer this question using randomization/shuffling first. Recall how we did this before. We used index cards to assign the different numerical values to two groups. We can extend this notion to two quantitative variables by shuffling all of the response variable values and assigning them to each of the values of the explanatory variable.

```
> gpaSim <- do(10000)*coef(lm(shuffle(gpa) ~ hours, data=extraHoursGPA))
> head(gpaSim)
  Intercept        hours
1  3.016554  0.004401307
2  3.189056  0.001348296
3  3.401398 -0.002409817
4  2.948519  0.005605413
5  2.950103  0.005577374
6  3.283498 -0.000323164
```

Remember that we can now make a histogram of these simulated non-academic `hours`. We then calculate the p-value by observing what proportion of simulated slopes fall at or below our observed sample slope of -0.005884.

```
> histogram(~hours, center=0, width=0.001, data=gpaSim)
```



```
> pValue <- prop(~hours <= -0.005884, data=gpaSim); pValue
  TRUE
0.0315
```

Remember that we had the following hypotheses in this problem: $H_0 : \beta_1 = 0$ versus $H_a : \beta_1 < 0$. With our p-value being 0.0315, we have evidence supporting the claim that a negative linear relationship exists between student GPA and the number of activities not related to their college/university. In other words, spending too many hours doing non-academic stuff can be detrimental to your GPA. That may seem like a DUH! kind of thing but it's important to realize that this, albeit self-reported, data supports that conclusion.

4.4.2 Theory-Based Approach

With our resulting simulated histogram being bell-shaped and assuming the other conditions are met, we can also use the t distribution to approximate this simulation. Traditionally, this is the way that statistics has been taught. We make assumptions and if they are met, we can use probability distributions to calculate p-values. In this case, we can calculate a standardized/studentized statistic as follows:

$$t = \frac{\hat{\beta}_1}{SE_{\hat{\beta}_1}}.$$

The value of this studentized statistic tells us how many standard deviations our observed statistic is from the hypothesized value of 0. We can get to these values using the following command in R:

```
> summary(gpaModel)
Call:
lm(formula = gpa ~ hours, data = extraHoursGPA)

Residuals:
     Min       1Q   Median       3Q      Max
-1.27699 -0.20544  0.05476  0.33931  0.70479

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.597691   0.185733  19.370   <2e-16 ***
hours       -0.005884   0.003070  -1.917   0.0624 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4303 on 40 degrees of freedom
Multiple R-squared:  0.08411,        Adjusted R-squared:  0.06121
F-statistic: 3.673 on 1 and 40 DF,  p-value: 0.06245
```

The estimate of our slope from this sample is $\hat{\beta}_1 = -0.005884$ with corresponding standard error (the standard deviation of the resulting t distribution) of $SE_{\hat{\beta}_1} = 0.003070$. Therefore, our observed studentized statistic is $t_{obs} = -1.914$. R calculates the two-sided p-value corresponding to this observed t statistic as 0.0624. Our one-sided, left-tailed p-value is, thus, 0.0312. We see that

this value is close to our simulated p-value above and leads us to the same conclusion to reject the null hypothesis.

4.4.3 Confidence Intervals

Remember that we can use the standard deviation of the simulated distribution to calculate a confidence interval for plausible values of the population slope.

```
> sdGPA <- sd(~hours, data=gpaSim); sdGPA
[1] 0.003101815
```

We can then create a 95% confidence interval that will be centered at the sample slope we calculated earlier.

```
> lower <- -0.005884 - 2*sdGPA; lower
[1] -0.01208763
> higher <- -0.005884 + 2*sdGPA; higher
[1] 0.0003196303
```

We see here that 0 is included in the confidence interval. One could have guessed that this would be the case since our p-values were around 3%. Recall that these p-values correspond to the one-sided, left-tailed tests. When we double these p-values we get something closer to 6% and with that being greater than 5%, we obtain that zero is a plausible value for our population slope.

We can also use the theory-based approach in this case (assuming that the conditions have been met).

```
> confint(gpaModel)
                 2.5 %       97.5 %
(Intercept)  3.22231023 3.9730716696
hours       -0.01208838 0.0003206377
```

We see that this gives similar values for the confidence interval. Again, we should expect this to be true since the simulated distribution is bell-shaped.

4.5 Exercises

Your solutions to these problems should be done in `RStudio` using an `RMarkdown` document and choosing the **Knit HTML** or **Knit PDF** option. Create a document that clearly labels each of the problems with their exercise number. Your document should include your `R` commands as well as the plots that were created by these commands. Make sure to also carefully answer the questions by discussing the output of the plots or the code wherever necessary.

Use the following information for the next two problems. Suppose that a statistics professor records the following for each student enrolled in her class:

- Gender
- Major
- Score on first exam
- Number of quizzes taken (a measure of class attendance)
- Time spent sleeping the previous night
- Handedness
- Political inclination
- Time spent on the final exam
- Score on the final exam

4.1) Identify whether each of the variables listed above is categorical or quantitative and give your reason why you made the choice for each.

4.2) Identify the response variable and the explanatory variable(s) in the following questions.

a) Do the proportions of left-handers differ between males and females on campus?

b) Are sleeping time, exam 1 score, and number of quizzes taken useful for predicting time spent on the final exam?

c) Does knowing a student's gender help to predict his or her major?

d) Does knowing a student's political inclination and time spent sleeping help to predict his or her gender?

4.3) For each of the following sports-related projects, identify observational units and the response and explanatory variables when appropriate. Also, classify the variables as quantitative or categorical.

a) Interested in predicting how long it takes to play a Major League Baseball game, an individual recorded the following information for all 15 games played on August 26, 2008: time to complete the game, total number of runs scored, margin of victory, total number of pitchers used, ballpark attendance at the game, and which league (National or American) the teams were in.

b) Over the course of several years, a golfer kept track of the length of all of his putts and whether or not he made the putt. He was interested in predicting whether or not he would make a putt based on how long it was.

c) Some students recorded lots of information about all of the football games played by LaDainian Tomlinson during the 2006 season. They recorded his rushing average, number of rushes, rushing touchdowns, receiving yardage, number of receptions, and receiving touchdowns.

4.4) In a study reported in the *Journal of Preventative Medicine*, 85 nutrition experts were asked to scoop themselves as much ice cream as they wanted. Some of them were randomly given a large bowl (34 ounces) as they entered the line, and the others were given a smaller bowl (17 ounces). Similarly, some were randomly given a large spoon (3 ounces) and the others were given a small spoon (2 ounces). Researchers then recorded how much ice cream each subject scooped for him- or herself. Their conjecture was that those given a larger bowl would tend to scoop more ice cream, as would those given a larger spoon.

a) Identify the observational units in this study.

b) Is this an observational study or a controlled experiment? Explain how you know.

c) Identify the response variable in this study, and classify it as quantitative or categorical.

d) Identify the explanatory variable(s) in this study, and classify it(them) as quantitative or categorical.

4.5) The number of calories and number of grams of sugar per serving were measured for 36 breakfast cereals. The data is available in my Public folder in the `Cereal.csv` file. We are interested in trying to predict the calories using the sugar content.

a) Make a scatterplot and comment on what you see.

b) Find the least squares regression line for predicting calories based on sugar content.

c) Interpret the value (not just the sign) of the slope of the fitted model in the context of this setting.

d) How many calories would the fitted model predict for a cereal that has 10 grams of sugar?

e) Cheerios has 110 calories but just 1 gram of sugar. Find the residual for this data point.

f) Does the linear regression model appear to be a good summary of the relationship between calories and sugar content of breakfast cereals?

4.6) Consider the fitted regression equation $\hat{Y} = 100 + 15X$. Identify whether the following statements are TRUE or FALSE and explain the reasons behind your choices.

a) The sample slope is 100.

b) The population slope is 15.

c) The predicted value of Y when $X = 0$ is 15.

d) The predicted value of Y when $X = 0$ is 100.

e) Larger values of X are associated with larger values of Y.

f) With every one unit increase in X, we expect Y to increase by 100 units.

g) With every one unit increase in X, we expect Y to increase by 15 units.

4.7) Consider the fitted regression equation $\hat{Y} = 25 + 7 \cdot X$. If $x_1 = 10$ and $y_1 = 100$, what is the residual for the first data point?

4.8) Priscilla Erickson from Kenyon College collected data on a stratified random sample of 116 Savannah sparrows at Kent Island. The weight (in grams) and wing length (in mm) were obtained for birds from nests that were reduced, controlled, or enlarged. The data is available in my Public folder as `Sparrows.csv`. Construct and interpret the following plots for the residuals of this model. In each case, discuss what the plot tells you about potential problems (if any) with the regression conditions.

- Histogram of residuals
- Normal quantile plot of the residuals
- Scatterplot that includes the least squares line. Are there any obvious outliers or influential points in this plot?

4.9) A capacitor was charged with a 9-volt battery and then a voltmeter recorded the voltage as the capacitor was discharged. Measurements were taken every 0.02 seconds. The data are in my Public folder in the `Volts.csv` file.

a) Make a scatterplot with `Voltage` on the vertical axis versus `Time` on the horizontal axis. Comment on the pattern.

b) Transform `Voltage` using a log transformation and then plot `log(Voltage)` versus `Time`. Comment on the pattern.

c) Regress `log(Voltage)` on `Time` and give the prediction equation.

d) Make a plot of residuals versus fitted values from the regression from part (c). Comment on the pattern.

e) Make an appropriate Q-Q plot. Discuss whether the conditions have been met for this transformed model.

4.10) Use the `Pines.csv` file in my Public folder which contains data from an experiment conducted by the Department of Biology at Kenyon College at a site near the Kenyon campus. In April 1990, student and faculty volunteers planted 1000 white pine seedlings at the Brown Family Environmental Center. Consider fitting a line for predicting height in 1997 from height in 1996.

a) Before doing any calculations, do you think that the height in 1996 will be a better predictor than the initial seedling height in 1990? Explain.

b) Fit a least squares line for predicting height in 1997 from height in 1996.

c) Does this simple linear regression model provide a good fit? Explain.

4.11) Biology students collected measurements on goldenrod galls at the Brown Family Environmental Center. The file `Goldenrod.csv` in my Public folder contains the gall diameter (in mm), stem diameter (in mm), wall thickness (in mm), and codes for the fate of the gall in 2003 and 2004.

a) Are stem diameter and gall diameter positively associated in 2003?

b) Plot wall thickness against stem diameter and gall diameter on two separate scatterplots for the 2003 data. Based on the scatterplots, which variable has a stronger linear association with wall thickness? Explain.

c) Fit a least squares regression line for predicting wall thickness from the variable with the strongest linear relationship in part (b).

d) Check that the conditions are met for an SLR.

e) Find the fitted value and residual for the first observation using the fitted model in (c).

f) What is the value of a typical residual for predicting wall thickness based on your linear model in part (c)?

4.12) Using the `porschePriceByMileage.csv` file and corresponding description referred to in the course notes, check to see if the observed sample slope is statistically DIFFERENT than 0. Use the randomization method and theory-based method for calculating the p-value. Also find and interpret 95% confidence intervals for the population slope in the two ways referenced in the course notes. Discuss similarities and differences between the results of the simulation and theory-based methods and some reasons for these comparisons.

Chapter 5

Multiple Regression

When a scatterplot shows a linear relationship between a quantitative explanatory variable X and a quantitative response variable Y, we fit a regression line to the data in order to describe the relationship. We can also use the line to predict the value of Y for a given value of X. For example, in Chapter 4 we used regression lines to describe relationships between:

- The price Y of a used Porsche and its mileage X.
- The number of doctors Y in a city and the number of hospitals X.

In both of these cases, other explanatory variables might improve our understanding of the response and help us to better predict Y:

- The price Y of a used Porsche may depend on its mileage X_1 and also its age X_2.
- The number of doctors Y in a city may depend on the number of hospitals X_1, the number of beds in those hospitals X_2, and the number of medicare recipients X_3.

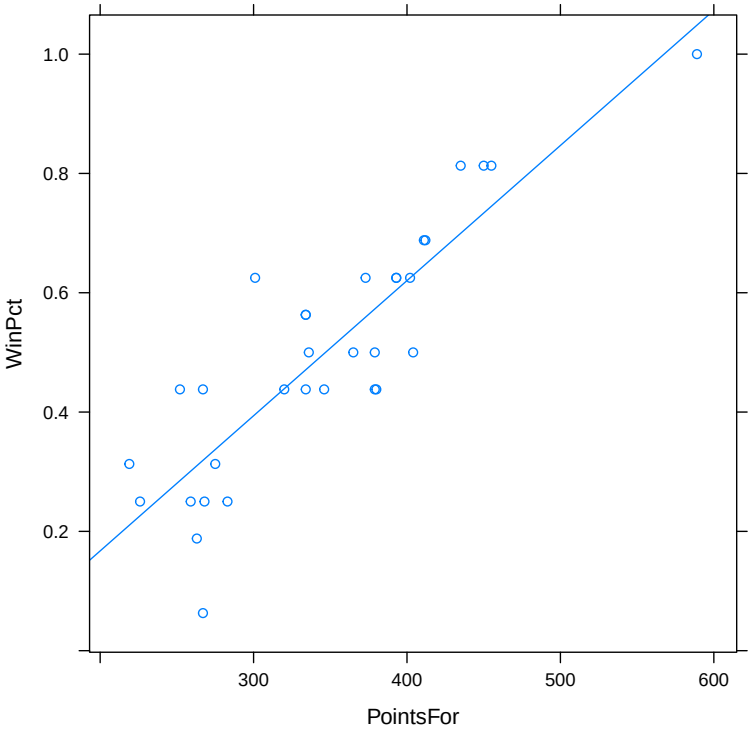
Thus far we have studied simple linear regression with a single quantitative predictor. This chapter introduces the more general case of **multiple linear regression**, which allows several explanatory variables to combine in explaining a response variable.

Example 5.1. Is offense or defense more important in winning football games? The data in my Public folder called `NFL2007Standings.csv` contains the records for all NFL teams during the 2007 regular season, along with the total number of points scored (`PointsFor`) and points allowed (`PointsAginst`).

Winning percentage Y could be related to points scored X_1 and/or points allowed X_2. The simple linear regressions for both of these relationships are shown below.

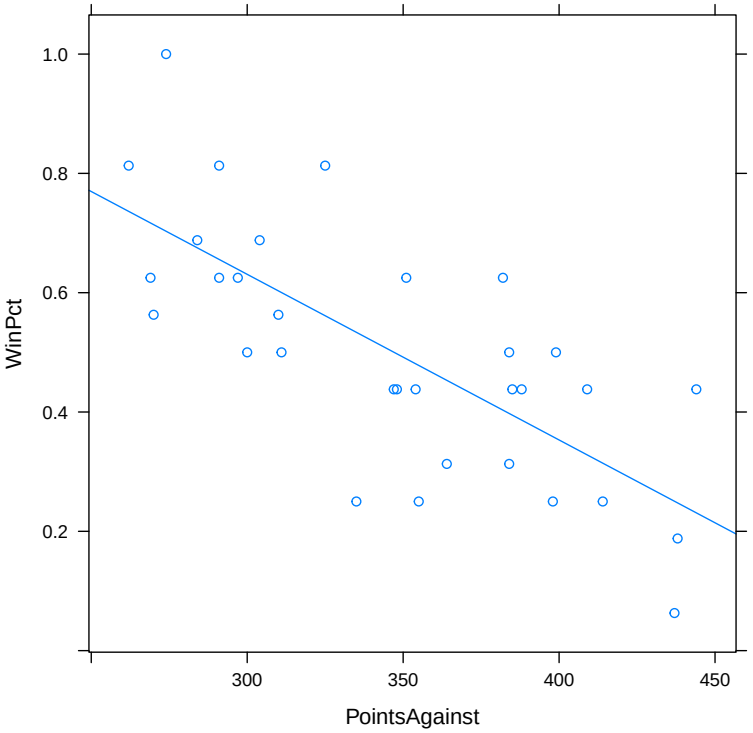
```
> nfl2007 <- read.csv("/shared/ismayc@ripon.edu/NFL2007Standings.csv")
> head(nfl2007)
                 Team Conference Division Wins Losses WinPct PointsFor
1 New England Patriots        AFC      ACE   16      0  1.000       589
2       Dallas Cowboys        NFC      NCE   13      3  0.813       455
3    Green Bay Packers        NFC      NCN   13      3  0.813       435
```

```
4   Indianapolis Colts        AFC      ACS   13      3  0.813       450
5 Jacksonville Jaguars        AFC      ACS   11      5  0.688       411
6   San Diego Chargers        AFC      ACW   11      5  0.688       412
  PointsAgainst NetPts TDs
1           274    315  75
2           325    130  54
3           291    144  49
4           262    188  54
5           304    107  50
6           284    128  49
> winScoreModel <- lm(WinPct ~ PointsFor, data=nfl2007)
> winScoreModel

Call:
lm(formula = WinPct ~ PointsFor, data = nfl2007)

Coefficients:
(Intercept)    PointsFor
  -0.285859     0.002266
> xyplot(WinPct ~ PointsFor, data=nfl2007, type=c("p","r"))
```




```
> winAgainstModel <- lm(WinPct ~ PointsAgainst, data=nfl2007)
> winAgainstModel

Call:
lm(formula = WinPct ~ PointsAgainst, data = nfl2007)
```

```
Coefficients:
  (Intercept)  PointsAgainst
     1.463785      -0.002777
> xyplot(WinPct ~ PointsAgainst, data=nfl2007, type=c("p","r"))
```



Not surprisingly, scoring more points is positively associated with increased winning percentage, while points allowed has a negative relationship. Could we improve the prediction of winning percentage by using both variables in the same model?

5.1 Multiple Linear Regression Model

Recall from Chapter 4 that the model for simple linear regression based on a single predictor X is

$$Y = \beta_0 + \beta_1 X + \varepsilon$$

where $\varepsilon \sim N(0, \sigma_\varepsilon)$ and the errors are independent from one another.

Choosing a Multiple Linear Regression Model

Moving to the more general case of multiple linear regression, we have k explanatory variables X_1, X_2, ..., X_k. The model now assumes that the mean response μ_Y for a particular set of values of the explanatory variables is a linear combination of those variables:

$$\mu_Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k.$$

As with the simple linear regression case, the model also assumes that repeated Y responses are **independent** of each other and that Y has a **constant variance** for any combination of

the predictors. When we need to do formal inference using theory-based methods for regression parameters, we also continue to assume that the distribution of Y for any fixed set of values for the explanatory variables follows a **normal distribution**. These conditions are summarized by assuming the errors in a multiple regression model are independent values from a $N(0, \sigma_\varepsilon)$ distribution.

Definition 5.1 (The Multiple Linear Regression Model)**.** *We have n observations on k explanatory variables X_1, X_2, ..., X_k and a response variable Y. Our goal is to study or predict the behavior of Y for the given set of the explanatory variables. The multiple linear regression model is*

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \varepsilon$$

where $\varepsilon \sim N(0, \sigma_\varepsilon)$ and the errors are independent from one another.

Fitting a Multiple Linear Regression Model

Once we have chosen a tentative set of predictors as the form for a multiple linear regression model, we need to estimate values for the coefficients based on data and then assess the fit. The estimation uses the same procedure of computing the sum of squared residuals, where the residuals are obtained as the differences between the actual Y values and the values obtained from a prediction equation of the form

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \cdots + \hat{\beta}_k X_k.$$

As in the case of simple linear regression, we will use `R` to choose estimates for the coefficients, β_0, β_1, β_2, $\cdots$, β_k, that minimize the sum of the squared residuals.

Example 5.2. We will continue working with the data from the 2007 NFL season. We now fit a multiple linear regression model to predict the winning percentages for NFL teams based on both the points scored and points allowed.

```
> winMultModel <- lm(WinPct ~ PointsFor + PointsAgainst, data=nfl2007)
> winMultModel
Call:
lm(formula = WinPct ~ PointsFor + PointsAgainst, data = nfl2007)

Coefficients:
  (Intercept)      PointsFor  PointsAgainst
     0.417223       0.001766      -0.001527
```

The fitted prediction equation in this example is

$$\hat{WinPct} = 0.417 + 0.00177 PointsFor - 0.00153 PointsAgainst.$$

If we consider the Green Bay Packers who scored 435 points while allowing 291 points during the 2007 regular season, the predicted winning percentage is

$$\hat{WinPct}(435, 291) = 0.417 + 0.00177 \cdot 435 - 0.00153 \cdot 291 = 0.742.$$

Remember this can also be found by using the `fitted.values()` function in `R` and by selecting the appropriate value in that array. The data is sorted based on the number of wins and the Packers are the third team in the list.

```
> winMultFits <- fitted.values(winMultModel)
> winMultFits[[3]]
[1] 0.7412102
```

Since the Packers' 13-3 record produced an actual winning percentage of 0.813, the residual in this case is $0.813 - 0.741 = 0.072$. This can also be found in R:

```
> winMultResids <- residuals(winMultModel)
> winMultResids[[3]]
[1] 0.07178975
```

In addition to the estimates of the regression coefficients, the other parameter of the multiple regression model that we need to estimate is the standard deviation of the error term, σ_ε. Recall that for the simple linear model, we estimated the variance of the error by dividing the sum of the squared residuals SSE by $n-2$ degrees of freedom. For each additional predictor we add to a multiple regression model, we have a new coefficient to estimate and thus lose 1 more degree of freedom. In general, if our model has k predictors (plus the constant term), we lose $k+1$ degrees of freedom when estimating the error variability, leaving $n-k-1$ degrees of freedom in the denominator. This gives the estimate for the **standard error of the multiple regression model** with k predictors as

$$\hat{\sigma}_\varepsilon = \sqrt{\frac{SSE}{n-k-1}}.$$

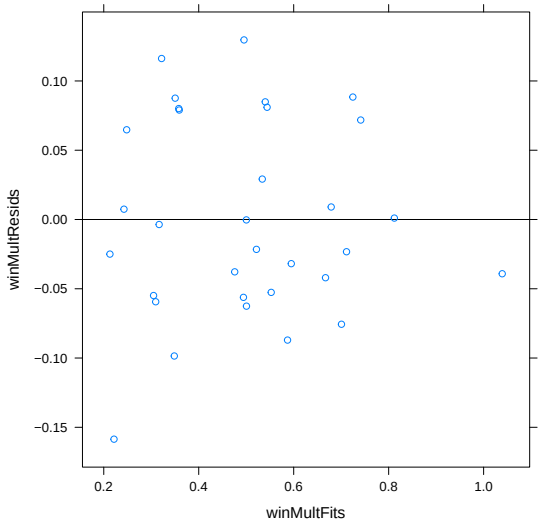
Using the `summary()` function in R for the multiple regression model for NFL winning percentages, we obtain this `Residual standard error` value of 0.07298.

```
> summary(winMultModel)
Call:
lm(formula = WinPct ~ PointsFor + PointsAgainst, data = nfl2007)

Residuals:
     Min       1Q   Median       3Q      Max
-0.15857 -0.05318 -0.01259  0.07360  0.12962

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.4172230  0.1394480   2.992  0.00561 **
PointsFor      0.0017662  0.0001870   9.445 2.37e-10 ***
PointsAgainst -0.0015268  0.0002751  -5.551 5.50e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.07298 on 29 degrees of freedom
Multiple R-squared:  0.8844,        Adjusted R-squared:  0.8764
F-statistic: 110.9 on 2 and 29 DF,  p-value: 2.598e-14
```

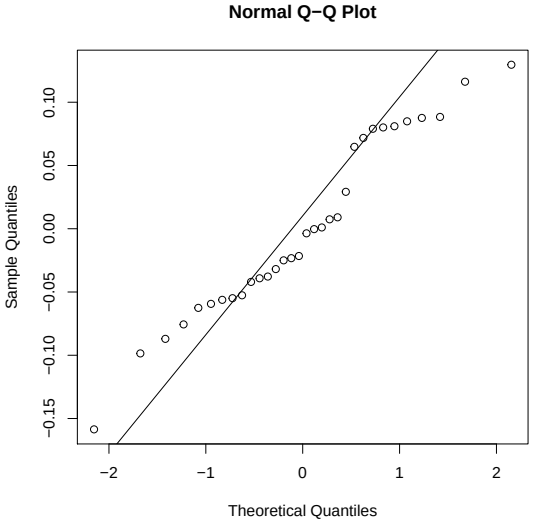
Assessing a Multiple Regression Model

In the next section, we will explore ways of using inference to determine whether or not an individual predictor is helpful to include in the model for predicting the response. Just as we used Residuals versus Fits plots and Q-Q plots with simple linear regression, we can also use those as a check on the conditions required for the multiple regression model.

```
> xyplot(winMultResids ~ winMultFits, abline=c(0,0))
```




```
> qqnorm(winMultResids)
> qqline(winMultResids)
```



The Residuals versus Fits plot leaves us with little reason to doubt the constant variance assumption. The Q-Q plot does have us questioning the normality assumption a little though. In this case, it is probably a better idea to use bootstrapping and/or resampling techniques or to transform the model. We will do the former in the next section.

5.1.1 Coefficient of Determination

Another common way to assess the quality of a model fit is by using the **coefficient of determination**, R^2:

$$R^2 = \frac{\text{Variability explained in the model}}{\text{Total variability in } y}.$$

It is a measure of the percentage of the total variability in the response that is explained by the regression model. In general, values close to 1 imply that the predictor(s) explain the response variable well while values of R^2 that are close to 0 usually mean that the predictor(s) do not predict the response well.

For our multiple regression model on the 2007 NFL data, we can use `R` to find this value:

```
> rsquared(winMultModel)
[1] 0.8843678
```

Interpretting the coefficient of determination

Thus, we can conclude that 88.4% of the variability in winning percentage of NFL teams for the 2007 regular season can be explained by the regression model based on the points scored and points allowed.

We could also look at how well the individual predictors did at fitting winning percentage using the R^2 values for those SLR models:

```
> rsquared(winScoreModel)
[1] 0.7615114
> rsquared(winAgainstModel)
[1] 0.5286501
```

As we noticed with the scatterplots earlier, scoring more points is a better predictor than allowing fewer points on a team's winning percentage. As individual predictors in separate linear regression models, both `PointsFor` and `PointsAgainst` were less effective at explaining the variability in winning percentages than they are as a combination in the multiple regression model. This will always be the case. Adding a new predictor to a multiple regression model can never decrease the percentage of variability explained by that model. At the very least, we could put a coefficient of zero in front of a new predictor and obtain the same level of fit effectiveness. In general, adding a new predictor will decrease the sum of squared errors and thus increase the variability explained by the model. But does that increase reflect important new information provided by the new predictor or just extra variability explained due to random chance?

We will explore using inference on the individual predictors in the next section as a way to check this. Another way to account for the fact that R^2 tends to increase as new predictors are added to a model is to use an **adjusted coefficient of determination** that reflects the number of predictors in the model as well as the amount of variability explained. One common way to do this adjustment is to divide the total sum of squares and sum of squared errors by their respective degrees of freedom and subtract the result from one.

Definition 5.2 (Adjusted Coefficient of Determination)**.** *The adjusted R^2, which helps to account for the number of predictors in the model, is computed with*

$$R^2{}_{adj} = 1 - \frac{SSE/(n-k-1)}{SStotal/(n-1)}.$$

Here $SSTotal = \sum(y - \bar{y})^2$ and $SSE = \sum(y - \hat{y})^2$. This value is calculated in `R` using the `summary()` function and is called `Adjusted R-squared` in the output.

```
> summary(winMultModel)
Call:
lm(formula = WinPct ~ PointsFor + PointsAgainst, data = nfl2007)

Residuals:
     Min       1Q   Median       3Q      Max
-0.15857 -0.05318 -0.01259  0.07360  0.12962

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.4172230  0.1394480   2.992  0.00561 **
PointsFor      0.0017662  0.0001870   9.445 2.37e-10 ***
PointsAgainst -0.0015268  0.0002751  -5.551 5.50e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.07298 on 29 degrees of freedom
Multiple R-squared:  0.8844,        Adjusted R-squared:  0.8764
F-statistic: 110.9 on 2 and 29 DF,  p-value: 2.598e-14
```

Thus, for our two-predictor model of NFL winning percentages, we have $R^2_{adj} = 0.876$. While this number reveals relatively little new information on its own, it is particularly useful when comparing competing models based on different numbers of predictors.

5.2 Assessing a Multiple Regression Model Using Inference

In the last section, we mentioned that the normality assumption may have been violated after examining the Q-Q plot. Therefore, it makes sense that we should be leery of using theory-based inference since that depends on the residuals being close to normally distributed. Recall that we can use bootstrapping and/or resampling techniques in these sorts of situations to check for significance. In this case, we will be checking for the significance of the two different predictors in our NFL winning percentage example using bootstrapping.

5.2.1 Review of Bootstrapping

The basic idea behind bootstrapping is to use the data to generate an approximate sampling distribution for the statistic of interest, rather than relying on conditions being met to justify some theoretical distribution. In general, a sampling distribution shows how the values of a statistic (such as a mean, standard deviation, or regression coefficient) vary when taking many samples of the same size from the same population. In practice, we generally have just our original sample and cannot generate lots of new samples from the population. The bootstrap procedure involves creating new samples from the original sample (not the whole population) by sampling with replacement. We are

essentially assuming that the population looks roughly like many copies of the original sample, so we can simulate what additional samples might look like. For each simulated sample we calculate the desired statistic, repeating the process many times to generate a bootstrap distribution of possible values for the statistic. We can then use this bootstrap distribution to estimate quantities such as the standard deviation of the statistic or to find bounds on plausible values for the parameter.

Example 5.3. We now return to our NFL winning percentage example. Remember that we are interested in determining whether each of the regression coefficients on $PointsFor$ and $PointsAgainst$ are statistically different from 0. In other words, we want to see whether each of these two explanatory variables add value to predicting the response variable of winning percentage. Recall that our NFL data has 32 observational units corresponding to each of the NFL teams and 10 variables. We are only interested in four of those variables: $WinPct$, $PointsFor$, $PointsAgainst$, and $Team$. We keep $Team$ mostly to allow us to go back and better understand what each of the values represent. Remember that we can isolate particular variables in our data set such as below:

```
> nflSub <- select(nfl2007, Team, PointsFor, PointsAgainst, WinPct)
> head(nflSub)
                 Team PointsFor PointsAgainst WinPct
1 New England Patriots       589           274  1.000
2       Dallas Cowboys       455           325  0.813
3    Green Bay Packers       435           291  0.813
4   Indianapolis Colts       450           262  0.813
5 Jacksonville Jaguars       411           304  0.688
6   San Diego Chargers       412           284  0.688
```

Significance of $PointsFor$ and/or $PointsAgainst$

We can now check to see if the coefficient in the multiple regression model fit for $PointsFor$ is significant. We follow methodology similar to that given in Section 3.3 of the notes. Remember that bootstrapping depends on us resampling from our observed sample of values. We begin by examining the values of one such resampling of our NFL data.

```
> resample(nflSub)
                    Team PointsFor PointsAgainst WinPct orig.ids
18          Buffalo Bills       252           354  0.438       18
24     New Orleans Saints       379           388  0.438       24
25       Baltimore Ravens       275           384  0.313       25
8         New York Giants       373           351  0.625        8
16      Minnesota Vikings       365           311  0.500       16
7        Cleveland Browns       402           382  0.625        7
32         Miami Dolphins       267           437  0.063       32
25.1     Baltimore Ravens       275           384  0.313       25
9     Pittsburgh Steelers       393           269  0.625        9
11       Tennessee Titans       301           297  0.625       11
20          Chicago Bears       334           348  0.438       20
31         St. Louis Rams       263           438  0.188       31
```

```
27          Atlanta Falcons       259          414  0.250       27
20.1          Chicago Bears       334          348  0.438       20
9.1    Pittsburgh Steelers        393          269  0.625        9
25.2      Baltimore Ravens        275          384  0.313       25
31.1        St. Louis Rams        263          438  0.188       31
10        Seattle Seahawks        393          291  0.625       10
19       Carolina Panthers        267          347  0.438       19
32.1        Miami Dolphins        267          437  0.063       32
29           New York Jets        268          355  0.250       29
1     New England Patriots        589          274  1.000        1
19.1    Carolina Panthers        267          347  0.438       19
29.1         New York Jets        268          355  0.250       29
15          Houston Texans        379          384  0.500       15
30         Oakland Raiders        283          398  0.250       30
1.1   New England Patriots        589          274  1.000        1
16.1     Minnesota Vikings        365          311  0.500       16
6       San Diego Chargers        412          284  0.688        6
14       Arizona Cardinals        404          399  0.500       14
4       Indianapolis Colts        450          262  0.813        4
12    Tampa Bay Buccaneers        334          270  0.563       12
```

These results may be a little strange on first sight. Some teams are selected multiple times. Recall that this is fine since we are sampling with replacement. It might be worthwhile to think of the population in this case as many years of data on NFL winning percentages. We want to see whether the 2007 data produces results that are significant in terms of the prediction of winning percentage based on our two predictor variables.

We can now look at the coefficients of a regression fit on a new resample.

```
> coef(lm(WinPct ~ PointsFor + PointsAgainst, data = resample(nflSub)))
  (Intercept)     PointsFor PointsAgainst
  0.317938836   0.002028336  -0.001520103
```

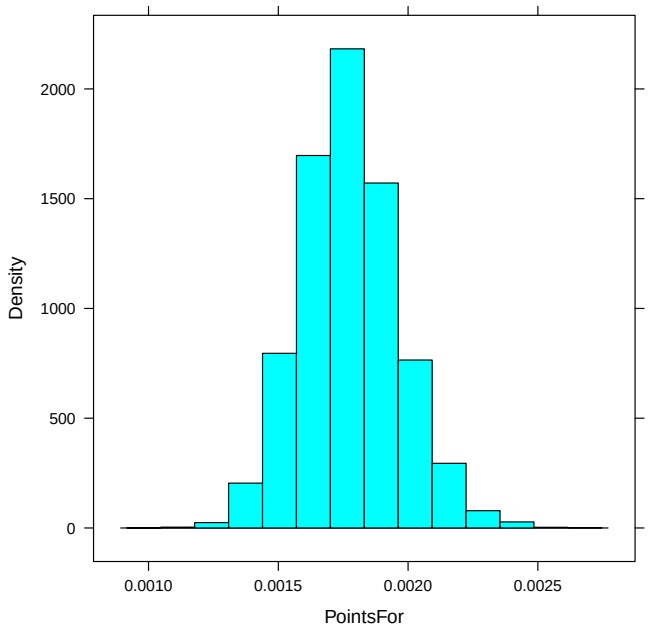
Remember that the goal is to see if it makes sense to say that our observed sample statistics of the regression coefficients on *PointsFor* and *PointsAgainst* are statistically different from 0. We can get to testing these conclusion by looking at the variability in many simulations of bootstrapped coefficients of *PointsFor* and *PointsAgainst*. We then can build a confidence interval and see whether 0 is contained in those intervals.

```
> nflSim <- do(10000)*coef(lm(WinPct ~ PointsFor + PointsAgainst,
+                  data = resample(nflSub)))
> head(nflSim)
  Intercept   PointsFor PointsAgainst
1 0.5536433 0.001607878  -0.001786242
2 0.3003080 0.001808782  -0.001227563
3 0.2864667 0.001880976  -0.001251631
```

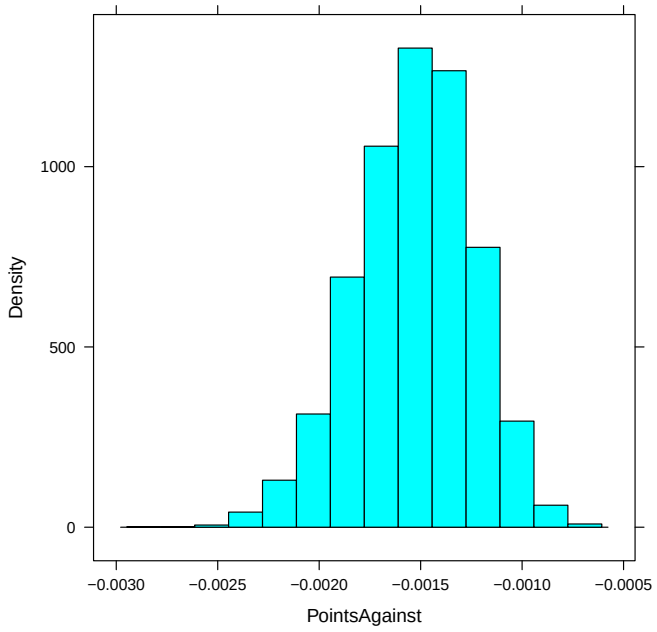
```
4 0.6348444 0.001795586  -0.002176157
5 0.4714942 0.001722245  -0.001623823
6 0.3088277 0.001877956  -0.001258700
```

We can now make a histogram of these simulated coefficients. We will focus on the *PointsFor* coefficients first and then look at the *PointsAgainst* coefficients. Recall that these histograms will NOT be centered at 0, but INSTEAD at the observed sample statistic of 0.001766 and -0.0015268, respectively, from our initial fit of the multiple regression model.

```
> histogram(~PointsFor, center=0.001766, data=nflSim)
```




```
> histogram(~PointsAgainst, center=-0.0015268, data=nflSim)
```



Note that both of these histograms deviate from a normal bell-shaped distribution, which we might have suspected given the Q-Q plot earlier. We now want to see whether 0 is contained in either of the confidence intervals based on these histograms. It's important to check here whether you believe that 0 will fall in a confidence interval based on looking at the histogram. In other words, does 0 fall in the left-tail of the distributions or closer to the center?

```
> confint(nflSim, level=0.95, method="quantile")

           name        lower         upper level   method
1     Intercept  0.127384597  0.7139446208  0.95 quantile
2     PointsFor  0.001424447  0.0021849199  0.95 quantile
3 PointsAgainst -0.002128595 -0.0009966411  0.95 quantile
```

We see that 0 does not fall in the confidence interval for either *PointsFor* or *PointsAgainst*. We can thus say that we have evidence that both *PointsFor* and *PointsAgainst* are important predictor variables for $WinPct$.

We then could look at ways to USE the model to predict different values of the winning percentage given different values of the predictor variables. This exercise is left for the reader to explore.

5.3 Polynomial Regression

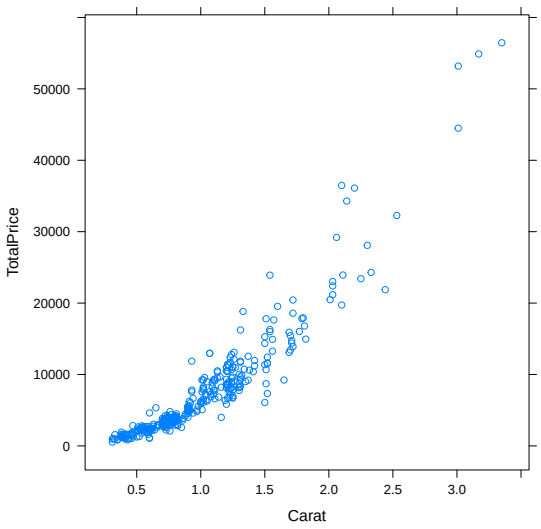
Earlier we looked at methods for dealing with data that showed a curved rather than linear relationship by transforming one or both of the variables. Now that we have a multiple regression model, another way to deal with curvature is to add powers of one or more predictor variables to the model.

Example 5.4. A young couple are shopping for a diamond and are interested in learning more about how these gems are priced. They have heard about the four C's: carat, color, cut, and clarity. Now they want to see if there is any relationship between these diamond characteristics and the price. This information is stored in my Public folder in the `Diamonds.csv` file. It contains quantitative information on the size ($Carat$), price ($PricePerCt$ and $TotalPrice$), and the *Depth* of the cut. *Color* and *Clarity* are coded as categorical variables.

Step 1: CHOOSE

Since the young couple are primarily interested in the total cost, they decide to begin by examining the relationship between the $TotalPrice$ and $Carat$. The scatterplot below examines this relationship for the 351 diamonds in the sample.

```
> diamonds <- read.csv("/shared/ismayc@ripon.edu/Diamonds.csv")
> xyplot(TotalPrice ~ Carat, data=diamonds)
```



Not surprisingly, the price tends to increase as the size of a diamond increases with a bit of curvature (a faster increase occurs among larger diamonds). One way to model this sort of curvature is with a quadratic regression model.

Definition 5.3 (Quadratic Regression Model)**.** *For a single quantitative predictor X, a quadratic regression model has the form*

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \varepsilon.$$

We now fit the quadratic model as shown below.

```
> diamondQuadModel <- lm(TotalPrice ~ Carat + I(Carat^2), data=diamonds)
> diamondQuadModel
Call:
lm(formula = TotalPrice ~ Carat + I(Carat^2), data = diamonds)

Coefficients:
(Intercept)        Carat   I(Carat^2)
     -522.7       2386.0       4498.2
```

Note the use of the `I()` function here. Without it, the term $Carat^2$ would be interpreted as the interaction of $Carat$ with itself, which would reduce to just $Carat$ and, thus, would not be what we are wanting to fit. You will be exploring the remaining steps of **Fit**, **Assess**, and **Use** in the homework questions at the end of the chapter.

We can easily generalize the idea of quadratic regression to include additional powers of a single quantitative predictor variable. However, note that additional polynomial terms may not improve the model much.

Definition 5.4 (Polynomial Regression Model)**.** *For a single quantitative predictor X, a polynomial regression model of degree k has the form*

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_k X^k + \varepsilon.$$

Example 5.5. We will now examine the effect of adding a cubic term to the price versus carat fit.

```
> diamondCubeModel <- lm(TotalPrice ~ Carat + I(Carat^2) + I(Carat^3),
+                        data=diamonds)
> diamondCubeModel

Call:
lm(formula = TotalPrice ~ Carat + I(Carat^2) + I(Carat^3), data = diamonds)

Coefficients:
(Intercept)        Carat   I(Carat^2)   I(Carat^3)
    -723.44      2942.02      4077.65        87.92
```

We can use the theory-based methods here to check to see if adding the cubic term here provides a better fit to the data. (We should be a little leery, in general, in using the theory-based methods without a check of the Residuals versus Fits plot for the quadratic model. We proceed here so that we can gain some insights into whether or not the cubic term is valuable. In reality, you should check that the conditions are met. If they are not, we should proceed using bootstrapping to test for significance of terms as done earlier in this chapter.)

```
> summary(diamondCubeModel)

Call:
lm(formula = TotalPrice ~ Carat + I(Carat^2) + I(Carat^3), data = diamonds)

Residuals:
     Min       1Q   Median       3Q      Max
-10136.8   -725.2   -182.1    380.5  12220.8

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -723.44     875.50  -0.826  0.40919
Carat        2942.02    2185.44   1.346  0.17912
I(Carat^2)   4077.65    1573.80   2.591  0.00997 **
I(Carat^3)     87.92     324.38   0.271  0.78652
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2130 on 347 degrees of freedom
Multiple R-squared:  0.9257,        Adjusted R-squared:  0.9251
F-statistic:  1442 on 3 and 347 DF,  p-value: < 2.2e-16
```

We see here that the p-value corresponding to the test of the significance of the $Carat^3$ term is large. This means that zero is a plausible value for the coefficient on the $Carat^3$ term, which gives us evidence that the cubic model is not appropriate here. We also note that it is convention to keep all lower order terms in our model. Therefore, even though the p-value is large for the coefficient of $Carat$ in this case, we still keep $Carat$ in the model since we have a significant result for $Carat^2$.

5.4 Exercises

Your solutions to these problems should be done in `RStudio` using an `RMarkdown` document and choosing the **Knit HTML** or **Knit PDF** option. Create a document that clearly labels each of the problems with their exercise number. Your document should include your `R` commands as well as the plots that were created by these commands. Make sure to also carefully answer the questions by discussing the output of the plots or the code wherever necessary.

5.1) A statistics professor assigned various grades during the semester including a midterm exam (out of 100 points) and a logistic regression project (out of 30 points). The prediction equation below was fit, using data from 24 students in the class, to predict the final exam score (out of 100 points) based on the midterm and project grades:

$$\hat{Final} = 11.0 + 0.53 \cdot Midterm + 1.20 \cdot Project.$$

a) What would this tell you about a student who got perfect scores on the midterm and project?

b) Michael got a grade of 87 on his midterm, 21 on the project, and an 80 on the final. Compute his residual and write a sentence to explain what that value means in Michael's case.

5.2) A regression model was fit to a sample of breakfast cereals. The response variable Y is calories per serving. The predictor variables are X_1, grams of sugar per serving, and X_2, grams of fiber per serving. The fitted regression model is

$$\hat{Y} = 109.3 + 1.0 \cdot X_1 - 3.7 \cdot X_2.$$

In the context of this setting interpret -3.7, the coefficient of X_2. That is, describe how fiber is related to calories per serving, in the presence of the sugar variable.

5.3) In an example in this chapter, we considered a model for the winning percentages of football teams based on measures of offensive and defensive ability. The file in my Public folder `MLB2007Standings.csv` contains similar data on many variables for Major League Baseball (MLB) teams from the 2007 regular season. The winning percentages are in the variable `WinPct` and scoring variables include `Runs` (scored by a team for the season) and `ERA` (essentially the average runs against a team per game).

a) Fit a multiple regression model to predict `WinPct` based on `Runs` and `ERA`. Write down the prediction equation.

b) Check that the model conditions of constant variance and normality are met by analyzing appropriate plots.

c) The Boston Red Sox had a winning percentage of 0.593 for the 2007 season. They scored 867 runs and had an ERA of 3.87. Use this information and the fitted model to find the residual for the Red Sox.

d) Comment on the effectiveness of each of the two predictors in this model. Would you recommend dropping one or the other (or both) from the model? Explain why or why not using appropriate statistical techniques.

e) Does this model for team winning percentages in baseball appear to be more or less effective than the model for football teams? Give a numerical justification for your answer.

5.4) Data in the file `MathEnrollment.csv` provides information on predicting spring enrollment in mathematics courses at a small liberal arts college.

a) Fit a multiple regression model for predicting spring enrollment (`Spring`) from fall enrollment (`Fall`) and academic year (`AYear`), after removing the data from 2003 that had special circumstances. Report the fitted prediction equation.

b) Prepare appropriate residual plots and comment on the conditions for the model.

c) What percent of the variability in spring enrollment is explained by the multiple regression model based on fall enrollment and academic year?

d) What is the size of the typical error for this multiple regression model?

5.5) In the Diamond example in this chapter, we looked at quadratic and cubic polynomial models for the price of diamonds (`TotalPrice`) based on the size (`Carat`). Another variable in the `Diamonds` datafile gives the `Depth` of the cut for each stone (as a percentage of the diameter). Run each of the models listed below, keeping track of the values for R^2, adjusted R^2, and which terms (according to bootstrap tests) are important in each model. In addition, produce appropriate Residuals versus Fits plots and Q-Q plots to check for the validity of the conditions of the linear model fits.

- A quadratic model using `Depth`
- A two-predictor model using `Carat` and `Depth`
- A model using `Carat`, `Carat`^{2}, and `Depth`

Among these four models as well as the quadratic and cubic models from the example, which would you recommend using for `TotalPrice` of diamonds? Explain your choice using the values of R^2, adjusted R^2, and which terms (according to bootstrap tests) are important in each model.

5.6) The young couple described in the Diamonds example has found a 0.5 carat diamond with a depth of 62% that they are interested in buying. Use the model that you selected in the previous exercise to answer the following questions.

a) What average total price does the model you chose predict for a 0.5-carat diamond?

b) Calculate the residual if the actual price of the diamond is $1000.

c) Does this residual fall within the range we would expect based on the `Residual standard error` of the model? Explain some practical reasons for why it does or does not.

Chapter 6

Logistic Regression

Are students with higher GPAs more likely to get into medical school? If you carry a heavier backpack, are you more likely to have back problems? You can think of these two questions of terms of the observational units, explanatory variables, and response variables:

- Are students with higher GPAs more likely to get into medical school?

- Observational units: students
- Explanatory variable: GPA (quantitative)
- Response variable: Accepted into medical school? (Y/N)

- If you carry a heavier backpack, are you more likely to have back problems?

- Observational units: students
- Explanatory variable: weight of backpack (quantitative)
- Response variable: back problems? (Y/N)

Up to this point in the course, we have dealt with a quantitative response. As you can see in the above two questions, the response here is *categorical*. More precisely, the response is *binary*, consisting of only two possible values.

Statistical modeling when your response is binary uses **logistic regression**. We will discuss what "logistic" means as we progress through this chapter. There are many similarities to the linear regression we have worked with already, but there are many differences due to the Yes/No nature of the response as well.

6.1 Choosing a Logistic Regression Model

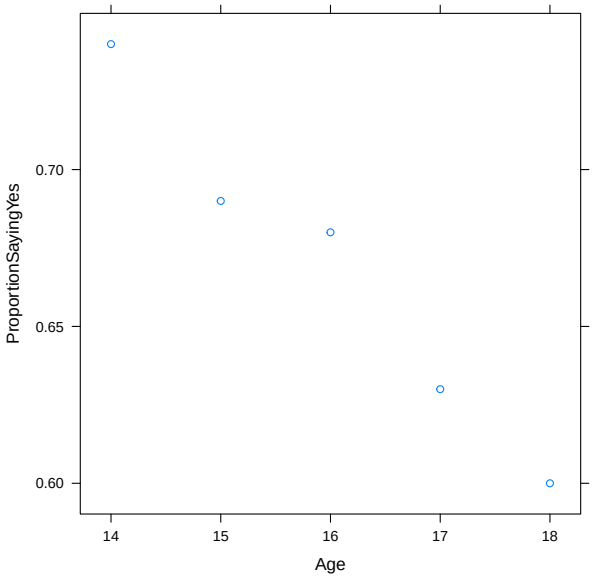
We will begin with an example to show how a logistic regression model differs from the ordinary linear regression model. We will also discuss why a transformation is needed when we have a binary response. It is this need to transform that makes logistic regression somewhat more complicated than ordinary regression.

Example 6.1. As teenagers age, their sleep habits change. The data table below summarizes the sleep habits of a random sample of 346 teens aged 14 to 18, who answered the question, "On an average school night, how many hours of sleep do you get?" The different ages and their corresponding proportions of successes are given in the **teenSleep.csv** files in my Public folder.

		Age				
		14	15	16	17	18
At least 7 hours of sleep?	No	12	35	37	39	27
	Yes	34	79	77	65	41
	Total	**46**	**114**	**114**	**104**	**68**
	Proportion of Yes	0.74	0.69	0.68	0.63	0.60

We will now plot the proportion of Yes answers against age. Notice that the plot looks roughly linear. You may be tempted to fit an ordinary regression line. DON'T! There are many reasons not to use ordinary linear regression. Here is just one. For ordinary regression, we model the response Y with a linear predictor of the form $\beta_0 + \beta_1 X$. If the error terms are not large, you can often see the relationship between the variables clearly enough to get rough estimates of the slope and intercept by eye, as seen in the figure below.

```
> teenSleep <- read.csv("/shared/ismayc@ripon.edu/teenSleep.csv")
> xyplot(ProportionSayingYes ~ Age, data=teenSleep)
```

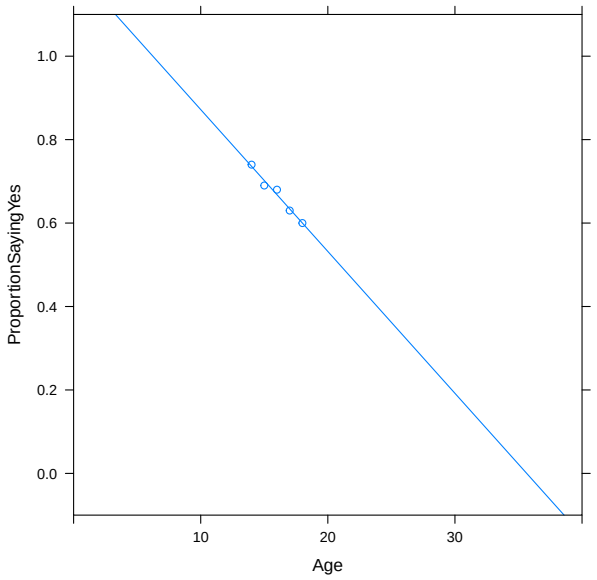


For logistic regression, the response is Yes or No, and we want to model $p = P(success) =$ the probability of a success. We still use a linear predictor of the form $\beta_0 + \beta_1 X$, but not in the usual way. If we were to use the model $\hat{p} = \hat{\beta}_0 + \hat{\beta}_1 X$, we would run into the problem of impossible values for $\hat{p}$. We need, instead, a model that takes values of $\beta_0 + \beta_1 X$ and gives back probabilities between 0 and 1.

You may be wondering what I mean when I say "impossible values" above. If we fit a line to the plot above, we see the line gives a good fit for ages between 14 and 18, but if you extend

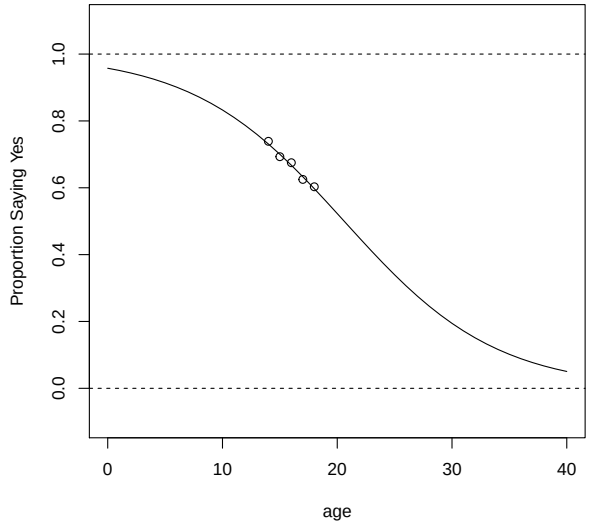
the fitted line, you run into big trouble. (Check what the model says for 1-year-olds, and for 40-year-olds.) Notice, also, that unless your fitted slope is exactly 0, *any* regression line will give fitted probabilities less than zero and greater than one. The way to avoid impossible values is to transform.

```
> xyplot(ProportionSayingYes ~ Age, data=teenSleep, type=c("p", "r"),
+        ylim=c(-0.1,1.1), xlim=c(0,40))
```



The Logistic Transformation

The logistic regression model always gives fitted values between 0 and 1. The figure below shows the fitted logistic model for the teenage sleep data. Note the elongated "S" shape—a backward S here—that is typical of logistic regression models.



The shape of the graph suggests a question: How can we get a *curved* relationship from a *linear* predictor of the form $\beta_0 + \beta_1 X$? In fact, you've already seen an answer to this question. For

example, for the data on doctors and hospitals, the relationship between Y = number of doctors and X = number of hospitals was curved. To get a linear relationship, we transformed the response (to square roots), fit a line, and then transformed back to get the fitted curve.

That's what we'll do here for logistic regression, although the transformation is more complicated and there are other complications as well. Here's a schematic summary so far:

Ordinary regression: Response $\approx$ Intercept + Slope $\cdot X$
Doctors and hospitals: (Number of doctors)$^{1/2} \approx$ Intercept + Slope $\cdot X$
Logistic regression: ?? $\approx$ Intercept + Slope $\cdot X$

The ?? on the left-hand side of the logistic equation will be replaced by a new transformation called the $log(odds)$.

Definition 6.1 (Odds and log(Odds))**.** *Let $\pi = P(Y = 1)$ be a probability with $0 < \pi < 1$. Then the odds that $Y = 1$ is the ratio*

$$odds = \frac{\pi}{1 - \pi}$$

and the

$$log(odds) = log\left(\frac{\pi}{1 - \pi}\right).$$

Here the log is the natural log (base e).

You may have run into odds before. Often, they are expressed using two numbers, for example, "4 to 1" or "2 to 1" or "3 to 2." Mathematically, all that matters is the ratio, so, for example, 4 to 2 is the same as 2 to 1, and both are equal to 2: $4/2 = 2/1 = 2$. To go from odds to log(odds), you do just what the words suggest: Take the (natural) logarithm. Natural logs are base e and are oftentimes easier to work with.

Using "linear on the right, log(odds) on the left" gives the linear logistic model:

$$log(odds) = log\left(\frac{\pi}{1 - \pi}\right) = \beta_0 + \beta_1 X$$

Notation: For any fixed value of the predictor x, there are four probabilities:

	True value	Fitted value
Actual probability	p = true P(Yes) for this x	$\hat{p}$ = #Yes/(#Yes + #No)
Model probability	π = true P(Yes) from model	$\hat{\pi}$ = fitted P(Yes) from model

If the model is exactly correct, then $p = \pi$ and the two fitted values estimate the same number.

Two Versions of the Logistic Model: Transforming Back

The logistic model approximates the log(odds) using a linear predictor $\beta_0 + \beta_1 X$. If we know $\log(\pi/(1-\pi)) = \beta_0 + \beta_1 X$, what is the formula for π?

1. To go from log(odds) to odds, we use the exponential function e^x:

$$\text{odds} = e^{\log(\text{odds})}$$

2. You can check that if odds $= \pi/(1-\pi)$, then solving for π gives $\pi = \text{odds}/(1+\text{odds})$.

Putting 1 and 2 together, gives

$$\pi = \frac{e^{\log(\text{odds})}}{1 + e^{\log(\text{odds})}}.$$

The function $f(x) = \frac{e^x}{1+e^x}$ is commonly known as the *logistic function* and we now see where logistic regression gets its name.

Definition 6.2 (Logistic Regression Model for a Single Predictor)**.** *The* **logistic regression** *model for the probability of success π of a binary response variable based on a single predictor X has either of two equivalent forms:*

$$\textbf{\textit{Logit form:}}\ logit(\pi) = log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 X$$

or

$$\textbf{\textit{Probability form:}}\ \pi = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

Example 6.2. Every year in the United States, over 120,000 undergraduates submit applications in hopes of realizing their dreams to become physicians. Medical school applicants invest endless hours studying to boost their GPAs. They also invest considerable time in studying for the medical school admission test or MCAT. Which effort, improving GPA or increasing MCAT scores, is more helpful in medical school admission? We investigate these questions using data gathered on 55 medical school applicants from a liberal arts college in the Midwest. For each applicant, medical school *Acceptance* status (accepted or denied), *GPA*, *MCAT* scores, and *Gender* were collected. The data is stored in my Public folder in the **MedGPA.csv** file. The response variable *Acceptance* status is a binary response, where 1 is a "success" and 0 a "failure."

We will focus our attention on a logistic regression model with *Acceptance* as the response and *GPA* as the predictor. Before we used the `lm()` function in `R` to fit a linear regression model. We now will use the `glm()` function which stands for "generalized linear model." GLMs are very flexible statistical methods for fitting models to response-explanatory variable situations that do not conform to the somewhat strict conditions of the linear model or `lm()` function. We use the `glm()` function with the `family=binomial` argument to obtain the logistic regression model for predicting the probability of acceptance to medical school (the response variable) from the student's GPA (the explanatory variable).

```
> medGPA <- read.csv("/shared/ismayc@ripon.edu/MedGPA.csv")
> head(medGPA)

  Accept Acceptance Sex BCPM  GPA VR PS WS BS MCAT Apps
1      D          0   F 3.59 3.62 11  9  9  9   38    5
2      A          1   M 3.75 3.84 12 13  8 12   45    3
3      A          1   F 3.24 3.23  9 10  5  9   33   19
4      A          1   F 3.74 3.69 12 11  7 10   40    5
5      A          1   F 3.53 3.38  9 11  4 11   35   11
6      A          1   M 3.59 3.72 10  9  7 10   36    5

> medGPALogitModel <- glm(Acceptance ~ GPA, data=medGPA, family=binomial)
> medGPALogitModel

Call:  glm(formula = Acceptance ~ GPA, family = binomial, data = medGPA)

Coefficients:
(Intercept)          GPA
    -19.207        5.454

Degrees of Freedom: 54 Total (i.e. Null);  53 Residual
Null Deviance:            75.79
Residual Deviance: 56.84         AIC: 60.84
```

The fitted model is linear in the log(odds) scale: $logit(P(Accept)) = -19.21 + 5.45GPA$. To find the equation for $P(Accept)$, we "transform back." This takes two steps:

1. Step 1: Exponentiate to go from log(odds) to odds: odds $= e^{\log(\text{odds})}$. So

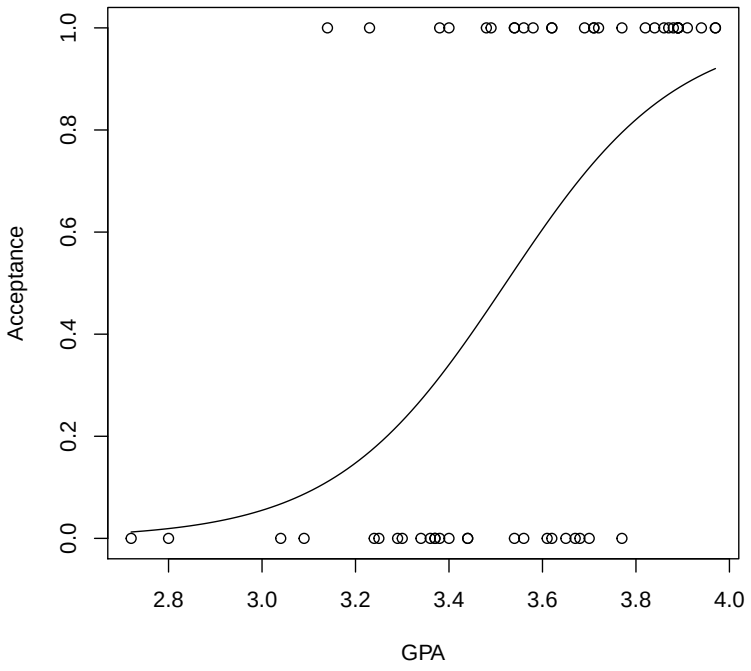
$$\text{odds}(Acceptance) = e^{-19.21+5.45GPA}$$

2. Step 2: Add 1 and form the ratio to go from odds to probability:

$$P(Accept) = \frac{\text{odds}}{1+\text{odds}} = \frac{e^{-19.21+5.45GPA}}{1+e^{-19.21+5.45GPA}}$$

We can also plot the data with the logistic curve defined by the formula for $P(Accept)$ using the following commands. Unfortunately, there is no easy way that I know of to plot a non-linear equation using the `xyplot()` command that we have used so far. The `add=T` option at the end of the `curve()` function specifies that we want to add the curve to the current plot.

```
> b0 <- coef(medGPALogitModel)[1]
> b1 <- coef(medGPALogitModel)[2]
> plot(Acceptance ~ GPA, data=medGPA)
> curve(exp(b0+b1*x)/(1+exp(b0+b1*x)), add=T)
```



For a concrete numerical example, consider a student with a GPA of 3.6. From the plot above, we can estimate that the chance of acceptance is about 0.6. To get the actual fitted value, we transform back with $x = 3.6$.

```
> x <- 3.6
> odds <- exp(b0+b1*x); odds
(Intercept)
   1.534945
> fittedPi <- odds/(1+odds); fittedPi
(Intercept)
  0.6055141
```

Note that the `b0` and `b1` above are calculated as the coefficients from the logistic regression model. According to this model, the fitted chance of acceptance for a student with a 3.6 GPA is 60%.

Randomness in the Logistic Model: Where Did the Error Term Go?

Recall the ordinary regression model $Y = \beta_0 + \beta_1 X + \varepsilon$. The parameters β_0 and β_1 are constant, and the values of the predictor X are regarded as fixed. The randomness in the model is in the error term ε. It is part of the model that these error terms are independent and normal, all with the same variance.

Randomness in the logistic regression model is different. There is no error term, and no normal distribution. To put this in a concrete setting, we will rely again on the medical school example. For each specific student and their value of GPA, we can think of the choice of acceptance as being modeled by a (possibly unfair) coin. The unfair part comes in because there isn't necessarily a

50/50 shot for each GPA. These sorts of random 0,1 outcomes are said to follow a **Bernoulli distribution**, which is a special case of the **Binomial distribution**. This is one reason why we specify `family=binomial` in our `glm` model.

Interpreting the Logistic "Slope" Coefficient

In ordinary simple linear regression, we interpret the slope as the change in the mean response for every increase of one in the predictor. Since the logit form of the logistic regression model relates the log of the odds to a linear function of the predictor, we can interpret the sample "slope" as the typical change in log(odds) for each one-unit increase in the explanatory variable. (Recall that for our medical school GPA problem, we had $\hat{\beta}_0 = -19.21$ and $\hat{\beta}_1 = 5.45$.)

However, log(odds) is not as easily interpretable as odds itself, so if we exponentiate both sides of the logit form of the model for a particular GPA we have

$$\frac{\hat{\pi}_{GPA}}{1-\hat{\pi}_{GPA}} = \text{odds}_{GPA} = e^{-19.21+5.45GPA}.$$

If we increase the GPA by one unit, we get

$$\frac{\hat{\pi}_{GPA+1}}{1-\hat{\pi}_{GPA+1}} = \text{odds}_{GPA+1} = e^{-19.21+5.45(GPA+1)}$$

So an increase of one GPA unit can be described in terms of the ratio of the two odds, commonly called the *odds ratio*:

$$\frac{\text{odds}_{GPA+1}}{\text{odds}_{GPA}} = \frac{e^{-19.21+5.45(GPA+1)}}{e^{-19.21+5.45GPA}} = e^{5.45}.$$

```
> exp(5.45)
[1] 232.7582
```

Therefore, a one-unit increase in GPA is associated with $e^{5.45}$, or a 233.7-fold, increase in the odds of acceptance! We see here a fairly direct interpretation of the estimated slope, $\hat{\beta}_1$: Increasing the predictor by one unit gives an odds ratio of $e^{\hat{\beta}_1}$, that is, the odds of success is multiplied by $e^{\hat{\beta}_1}$.

The magnitude of this increase appears to be extraordinary, but in fact it serves as a warning that the magnitude of the odds ratio depends on the units we use for measuring the predictor (just as the slope in ordinary regression depends on the units). Increasing your GPA from 2.9 to 3.9 is dramatic and you would certainly expect remarkable consequences. It might be more meaningful to think about a tenth of a unit change in grade point as opposed to an entire unit change. We can compute the odds ratio for a tenth of a unit increase by the following `R` command.

```
> exp(5.45*0.1)
[1] 1.724608
```

Note: We have skipped over just how `R` fits logistic regression models to the data. It is a little different than the method of minimizing the sum of the squared residuals as was done with linear regression. We won't discuss the details here but the procedure is known as Maximum Likelihood Estimation and is a topic that is commonly covered in a Mathematical Statistics course. In the next section, we will look at ways to assess the quality of the logistic regression fit and in the last section we will look at inference related to the logistic regression model.

6.2 Assessing the Logistic Regression Model

This section will deal with three issues related to the logistic model: linearity, randomness, and independence. Randomness and independence are essential for formal inference discussed in the last section of this chapter. Linearity is about how close the fitted curve comes to the data. If your data points are "logit-linear," the fitted logistic curve can be useful even if you can't justify formal theory-based inference.

- *Linearity is about pattern, something you can check with a plot.* You don't have to worry about how the data were produced.

- *Randomness and independence boil down to whether a spinner/unfair-coin-flip model is reasonable.* As a rule, graphs can't help you check this. You need to think instead about how the data were produced.

6.2.1 Linearity

The logistic regression model says that the log(odds)—that is, $\log(\pi/(1-\pi))$—are a linear function of x. In what follows, we check linearity for datasets with a *quantitative* predictor. We will do this by producing a plot based on the concept of the *empirical logit*.

Definition 6.3 (Empirical Logit)**.** *The* **empirical logit** *equals the log of the observed odds from the sample:*

$$\textit{Empirical logit} = \textit{logit}(\hat{p}) = \log\left(\frac{\hat{p}}{1-\hat{p}}\right) = \log\left(\frac{\#\,Yes}{\#\,No}\right)$$

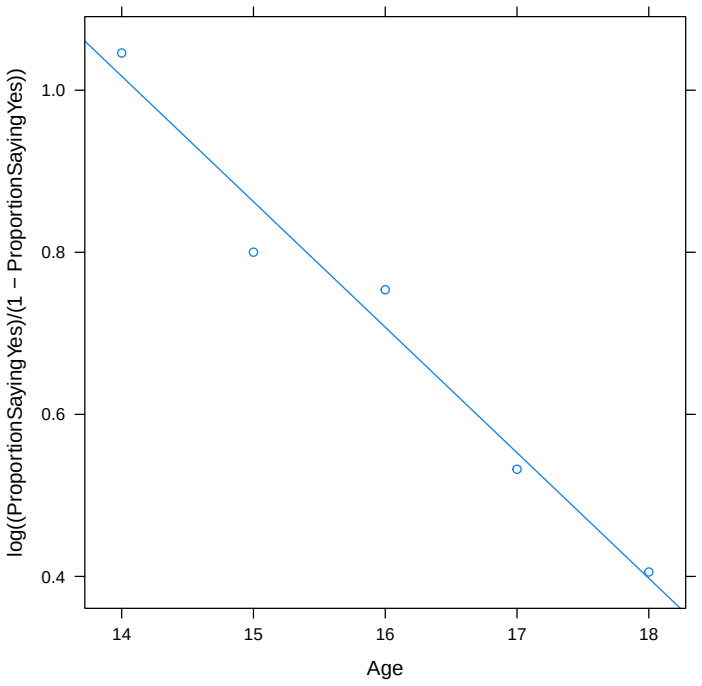
We then plot the empirical logit against the explanatory variable and look to see if the pattern is linear. (Note that this check on linearity only works when we have a one-to-one correspondence between the explanatory variable and the response variable such as the teenagers' sleep example. You can also create related plots for checking linearity in examples like the GPA and medical school example, but they are often hard to analyze and even more subjective than when we created the residuals versus fits plots.)

Example 6.3. Recall the `teenSleep.csv` file that contained the proportion of teens aged 14 to 18 that responded whether or not they slept more than 7 hours. We will now create the empirical logit plot for this example.

```
> teenSleep <- read.csv("/shared/ismayc@ripon.edu/teenSleep.csv")
```

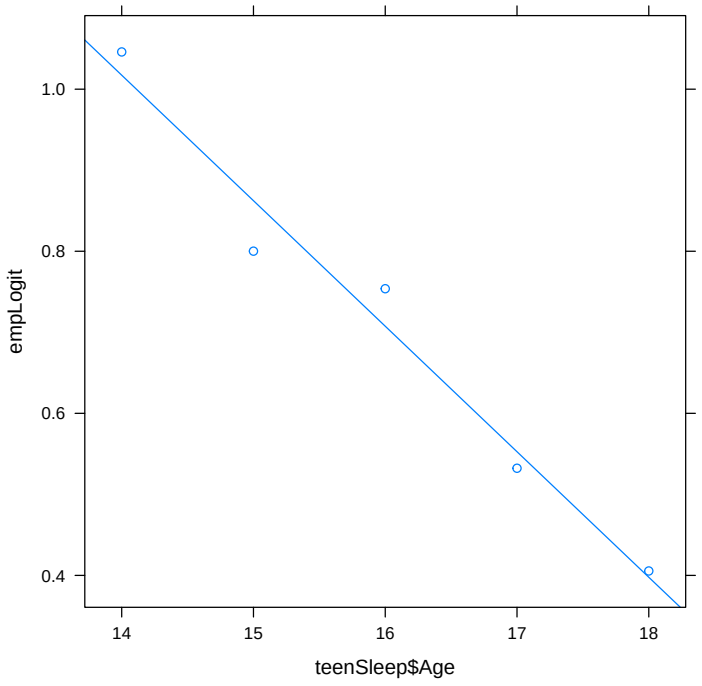
The proportion of successes (the proportion of those that responded "Yes" to the question) is given in the `ProportionSayingYes` variable. To create the empirical logit, we just need to transform each of the $\hat{p}$ values using the formula for the empirical logit.

```
> xyplot(log((ProportionSayingYes)/(1 - ProportionSayingYes)) ~ Age,
+        data=teenSleep, type=c("p","r"))
```



This could also have been done as follows. The $ is another way to access variables in a dataset.

```
> empLogit <- log((teenSleep$ProportionSayingYes)/
+                   (1 - teenSleep$ProportionSayingYes))
> xyplot(empLogit ~ teenSleep$Age, type=c("p","r"))
```



As we would have guessed based on looking at the original plot of $\hat{p}$ versus `Age`, the linearity condition is reasonable here.

The linearity condition tells how the Yes/No proportions are related to the x-values. The next two conditions, randomness and independence, are about whether and in what way the proportions are based on probabilities. Is it reasonable to think of each response value y coming from an independent (potentially unfair) coin flip for each x-value?

6.2.2 Randomness

Some proportions come from probabilities; others don't. For example, 50% of all fair coin flips land Heads. The 50% is based on a probability model. It is reasonable to model the outcomes of a fair coin flip using a spinner divided 50/50 into regions marked Heads and Tails. The unfair coin flip model is often thought of in this spinner model as well with the proportion of success shading some percentage of the spinner and the remaining portion shaded corresponding to the proportion of a failure.

For contrast, your body is about 90% water. This proportion is not random in the Yes/No spinner sense. It would be ridiculous to suggest that a random spinner model decides whether you end up 0% water or 100% water.

Why does the randomness condition matter? Because statistical tests and intervals are based on the probability model (or on the random sample being representative of a larger population). If the spinner model offers a good fit to the data, you can trust the tests and intervals. If not, you should question the results of the inference.

Reality, as usual, is rarely so clear-cut. Most applications of the theory fall in between. Examples will make this more concrete.

Example 6.4. Here are a couple scenarios to consider. You will have more chances to think about randomness in the exercises at the end of the chapter.

1. *Description:* A study investigated whether a handheld device that sends a magnetic pulse into a person's head might be an effective treatment for migraine headaches. Researchers recruited 200 randomly selected subjects who suffered from migraines and randomly assigned them to receive either the transcranial magnetic stimulation treatment or a sham (placebo) treatment from a device that did not deliver any stimulation. Subjects were instructed to apply the device at the onset of migraine symptoms and then assess how they felt two hours later.

   Randomness? Patients were randomly assigned either to the treatment group or the control group and were randomly selected from the population. The randomness in the assignment model and also the randomness in selection allows us to justify using a probability model such as the Bernoulli distribution here. (Either sort of randomness is enough for us to consider the Bernoulli model as being plausible.)

2. *Description:* Back in the early days of "hands-on learning," a professor introduced the logistic curve in his calculus classes using as an example the growth of bread model: Put a slice of non-preservative bread in a plastic bag with a moistened tissue, and wait for the black mold to appear. Each day, put a grid of small squares over the bread, and count the number of squares that show mold. Plot the logit of the proportion versus time (number of days).

Randomness? No, this does not follow any sort of coin flip process. But even so, the logistic fit is definitely useful as a description of the growth of mold over time, a model that ecologists use to describe growth in the presence of limited resources.

6.2.3 Independence

Even if outcomes are random, they may not be independent. For example, if you put tickets numbered 1 to 10 in a box, mix them up, and take them out one at a time, the sequence you get is random, but the individual outcomes are not independent. If your first ticket is 9, your second ticket cannot be. However, if you put 9 back and mix again before you grab the next ticket, your outcomes are both random *and* independent.

If you decide that randomness fails, you don't need to check independence because you already know you don't have the probability model you need to justify formal inference. Suppose, though, you have decided that it is reasonable to regard the outcomes as random. How can you check independence? It may help to think about time, space, and the Yes/No decision.

1. Time: The ticket example suggests one thing to check: Are the results from a time-ordered process? If so, is it reasonable to think that one outcome does not influence the next outcome in the sequence?

2. Space: If your observational units have a spatial relationship, you should ask whether it is reasonable to think that the outcome for one unit is independent of the nearby units. In this context, space may be only implied, as with children in the same grade school class.

3. The Yes/No decision: Some decisions are clear—was the ticket 9? Other decisions may depend on subjective judgment—is this Medicare claim justified? When Yes/No decisions are not objective, there is a possibility that the decision process introduces dependence. Here, as with randomness, many judgments about independence are less clear-cut than we would want, and examples can help.

Example 6.5. We will reconsider the problem descriptions in the previous example about randomness.

1. Migraine headaches

   Independence? Independence is reasonable because of the random assignment. There should be no dependence built in due to this process.

2. Moldy bread and space.

   Independence? Independence fails. A little square with no mold today is more likely to have mold tomorrow if it is next to a square with mold, less likely if not.

6.3 Inference for (Simple) Logistic Regression

This sections reveals how to use simulation and theory-based methods to do tests and create confidence intervals in the context of the example on medical school admissions and GPAs. We will look at the parallels between how we conducted inference with simple linear regression and how we can conduct inference in the case of one quantitative predictor and one binary response, i.e., simple logistic regression.

Recall that in simple linear regression we had several ways to assess whether the relationship between the predictor and response was stronger than one would expect by random chance alone and to assess the strength of the relationship. These included:

- Randomization-based methods to see if the slope differs from zero and confidence intervals to produce a range of plausible values for the slope and
- A theory-based t-test to see if the slope differs from zero.

Do we have analogous tests and measurements to help assess the effectiveness of a model in the logistic setting? The answer is Yes and we will see that the procedures are quite similar to those we did before with linear regression.

Example 6.6. We are now going to revisit the medical school GPA example from earlier in the chapter. We will also introduce how to add a new column into a dataframe in `R`.

```
> medGPA10 <- read.csv("/shared/ismayc@ripon.edu/MedGPA.csv")
> medGPA10$GPA10 <- medGPA10$GPA * 10
> head(medGPA10)
  Accept Acceptance Sex BCPM  GPA VR PS WS BS MCAT Apps GPA10
1      D          0   F 3.59 3.62 11  9  9  9   38    5  36.2
2      A          1   M 3.75 3.84 12 13  8 12   45    3  38.4
3      A          1   F 3.24 3.23  9 10  5  9   33   19  32.3
4      A          1   F 3.74 3.69 12 11  7 10   40    5  36.9
5      A          1   F 3.53 3.38  9 11  4 11   35   11  33.8
6      A          1   M 3.59 3.72 10  9  7 10   36    5  37.2
```

Here we introduce a new variable into our dataframe called 'GPA10' which is 10 times the `GPA` variable. This will make the GPA be measured in tenths of points rather than full points.

```
> medGPA10Logit <- glm(Acceptance ~ GPA10, data=medGPA10, family=binomial)
> medGPA10Logit
Call:  glm(formula = Acceptance ~ GPA10, family = binomial, data = medGPA10)

Coefficients:
(Intercept)        GPA10
   -19.2065       0.5454

Degrees of Freedom: 54 Total (i.e. Null);  53 Residual
Null Deviance:            75.79
Residual Deviance: 56.84        AIC: 60.84
```

```
> sampleLogitSlope <- coef(medGPA10Logit)[[2]]
> sampleLogitSlope
[1] 0.5454166
```

6.3.1 Randomization-Based Inference for the "Slope" Coefficient

Recall that randomization refers to the shuffling of the response variable and then assigning them to the values of the explanatory variable.

```
> medGPASim <- do(10000) * coef(glm(shuffle(Acceptance) ~ GPA10,
+                                   data=medGPA10, family=binomial))
```

We can now make a histogram of these simulated predicted variable values. After observing the shape and distribution of these values, we can start to think about how far our observed "slope" statistic of 0.545416591556059 is from the hypothesized value. Recall that we are testing

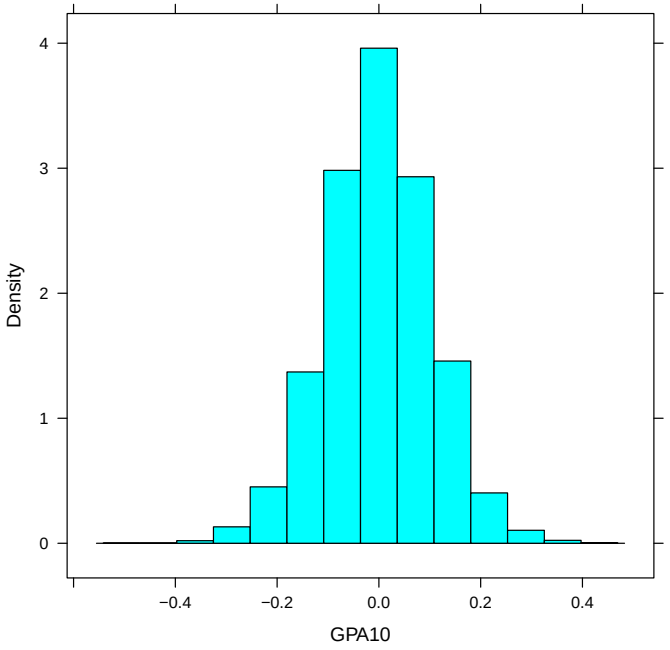
$$H_0 : \beta_1 = 0 \text{ versus } H_a : \beta_1 \neq 0$$

in the model

$$\text{logit}[P(Accept)] = \beta_0 + \beta_1 GPA10,$$

so our hypothesized value is 0.

```
> histogram(~GPA10, center=0, data=medGPASim)
```



We can see that our observed statistic falls far in the tail of the distribution and we have evidence to reject the null hypothesis that the slope coefficient is 0. We can calculate this p-value exactly using the `prop` function below. (Remember that this is a two-tailed test. We can use the absolute value function to add the corresponding probabilities from the two tails.)

```
> pValue <- prop(~abs(GPA10) >= sampleLogitSlope, data=medGPASim); pValue
TRUE
   0
```

With our p-value being 0, we have strong evidence supporting the claim that GPA is a quality predictor of whether or not a student was accepted into medical school.

With the shape of the simulated distribution being symmetric and bell-shaped, we can also calculate a confidence interval based on this method by using the standard deviation of the distribution. This tells us with, say, 95% confidence what plausible values are for the population "slope" coefficient β_1. (Remember that confidence intervals are always centered at the observed statistic NOT at the hypothesized value of 0.)

```
> sdSim <- sd(~GPA10, data=medGPASim)
> lower <- sampleLogitSlope - 2 * sdSim; lower
[1] 0.3365655
> upper <- sampleLogitSlope + 2 * sdSim; upper
[1] 0.7542677
```

With 0 being far outside this interval, we again have evidence to reject the null hypothesis that GPA is not a significant predictor of med school acceptance. We can say with 95% confidence that the true population "slope" coefficient on this logit model for GPA is between 0.336565516930413 and 0.754267666181705.

Remember that the slope here is measuring the change in log(odds) for every unit change in the predictor, and this is often difficult to interpret in a practical sense. As you saw in the previous section, you can convert a slope to an estimated odds ratio using $e^{0.5454} = 1.73$. Applying this same process to the confidence interval bounds for the slope produces a confidence interval for the odds ratio.

```
> lowerOddsRatio <- exp(lower); lowerOddsRatio
[1] 1.400131
> upperOddsRatio <- exp(upper); upperOddsRatio
[1] 2.126054
```

Using simulation-based methods, we can be 95% confidence that the odds of acceptance into medical school in the population of all students is between 1.40013059852392 and 2.12605397187797.

6.3.2 Theory-based t-test and Confidence Interval

With the randomization distribution being symmetric and bell-shaped, we suspect that we could also use the theory-based methods that are built-in to `R` to test for the significance of `GPA10` in the model. Recall that this can be done by using the `summary` function on the original fit of the generalized linear model.

```
> summary(medGPA10Logit)
```

```
Call:
glm(formula = Acceptance ~ GPA10, family = binomial, data = medGPA10)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7805  -0.8522   0.4407   0.7819   2.0967

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -19.2065     5.6287  -3.412 0.000644 ***
GPA10         0.5454     0.1579   3.454 0.000553 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

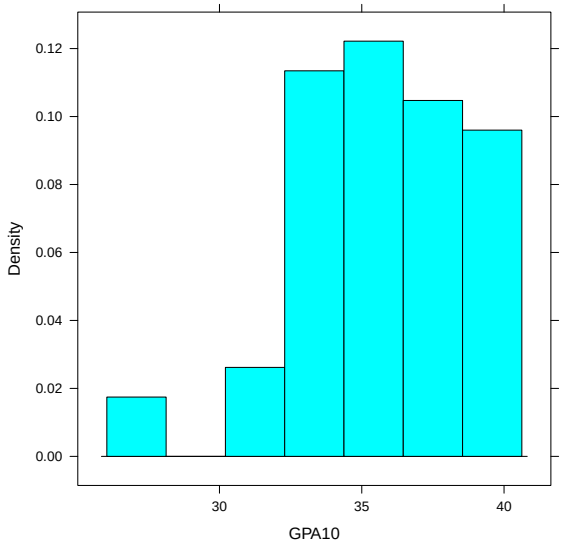
The p-value here of 0.000553 matches with the value of 0 we obtained using simulation-based methods. Thus, we have the same conclusion to reject the null hypothesis and say that we have evidence that GPA is a good predictor of medical school acceptance.

We can also create a confidence interval for plausible values of the coefficient on `GPA10` by using the `confint` function on our model:

```
> confint(medGPA10Logit)
                  2.5 %     97.5 %
(Intercept) -31.7132662 -9.3761026
GPA10         0.2696316  0.8965621
```

These results are somewhat similar to those seen with the simulation-based procedures, but are also a little different. This is likely due to there only being 55 observational units in our original sample. We can also see that the distribution of GPAs in our original sample is not symmetric. This may make us doubt whether the assumptions are met for theory-based procedures and lead us to go with the simulation-based methods as a more reliable predictor.

```
> histogram(~GPA10, data=medGPA10)
```



6.4 Exercises

Your solutions to these problems should be done in `RStudio` using an `RMarkdown` document and choosing the **Knit HTML** or **Knit PDF** option. Create a document that clearly labels each of the problems with their exercise number. Your document should include your `R` commands as well as the plots that were created by these commands. Make sure to also carefully answer the questions by discussing the output of the plots or the code wherever necessary.

6.1) Why does simple linear regression used in previous chapters not work well when the response is binary?

6.2) a) If the probability of an event occurring is 0.5, what are the odds?

b) If the probability of an event occurring is 0.9, what are the odds?

c) If the probability of an event occurring is 0.1, what are the odds?

6.3) a) If the odds of an event occurring is 2:1, what is the probability?

b) If the odds of an event occurring is 10:1, what is the probability?

c) If the odds of an event occurring is 1:4, what is the probability?

6.4) The `MedGPA.csv` file in my Public folder used in this chapter also contains information on the medical school admission test (MCAT) scores for the same sample of 55 students. Fit a logistic regression model to predict the *Acceptance* status using the $MCAT$ scores.

a) Write down the estimated versions of both the logit and probability forms for this model.

b) What would the estimated model say about the chance that a student with $MCAT = 40$ is accepted to medical school?

c) For approximately what $MCAT$ score would a student have roughly a 50-50 chance of being accepted to medical school?

6.5) In a study of 31 patients with esophageal cancer, it was found that in 18 of the patients the cancer had metastasized to the lymph nodes. Thus, an overall estimate of the probability of metastasis is $18/31 = 0.58$. A predictor variable measured on each patient is *Size* of the tumor (in cm). A fitted logistic regression model is

$$\log\left(\frac{\hat{\pi}}{1-\hat{\pi}}\right) = -2.086 + 0.5117 \cdot Size.$$

a) Use this model to estimate the odds of metastasis, $\pi/(1-\pi)$, if a patient's tumor size is 6 cm.

b) Use the model to predict the probability of metastasis if a patient's tumor size is 6 cm.

c) How much do the estimated odds change if the tumor size changes from 6 cm to 7 cm? Provide and interpret an odds ratio.

d) How much does the estimate of π change if the tumor size changes from 6 cm to 7 cm?

e) How large does a tumor need to be for the estimated probability of metastasis to be at least 0.80?

6.6) Some professors also enjoy playing golf. One such professor is an avid golfer. In a vain attempt to salvage something of value from all the hours he has wasted on the golf links, he has gathered data on his putting prowess. (For those of you that don't know anything about golf, a putt is an attempt to hit a ball using a golf club so that the ball rolls a few feet and falls into an expensive hole in the ground called the cup). He collects around a 100 different putts each from a variety of distances between 3 feet and 7 feet as he plays many rounds of golf. Comment on the conditions of independence and randomness in this setting.

6.7) Identify whether the conditions of randomness and independence are met in the following scenarios. After answering Yes or No, be sure to explain your reasoning. Some of these could go either way so your explanations are vital to you showing you understand the concepts.

a) During the 1970s, when women were entering the workforce in substantial numbers for the first time since World War II, many men were opposed to the trend. One study chose a random sample of men and asked them to agree or disagree with the statement "Women should stay in the home and let men run the country." A linear logistic regression relating the proportion of men who agreed to their years of education showed a strong relationship with a negative slope: The more time a man spent in school, the less likely he was to agree.

b) When Wierton Steel declared bankruptcy, hundreds of employees lost their jobs. After another company bought Wierton's assets, fewer than half of those same employees were rehired. A group of older employees sued, claiming age discrimination in the hiring decisions, and a logistic regression showed a strong relationship between age and whether a person was rehired.

c) Some professors also like to play the banjo. One of these professors tried to apply logistic regression to a bluegrass banjo "roll," an eight-note sequence with a fixed pattern. According to the logistic model, the predictor is the time in the sequence when the note is played and the response is whether the note is picked with the thumb.

6.8) The *Titanic* was a British luxury oceanliner that sank famously in the icy North Atlantic Ocean on its maiden voyage in April 1912. Of the approximately 2200 passengers on board, 1500 died. The high death rate was blamed largely on the inadequate supply of lifeboats, a result of the manufacturer's claim that the ship was "unsinkable." A partial dataset of the passenger list was compiled by Philip Hinde in his *Encyclopedia Titanica* and is given in the dataset `Titanic.csv` in my Public folder. We are interested in looking at the relationship between the variables `Survived` and `Age`.

a) Use a plot to explore whether there is a relationship between survival and the passenger's age. (*Hint:* A refresher of the different plots in sub-subsection 2.4.3.2 of these course notes will be helpful here.)

b) What can you conclude based on the plot alone?

c) Are the randomness and independence assumptions met here? Explain.

d) Assume that the conditions are met for the logistic model. Fit one to the survival and age variables. Give the logistic form and probability form of this fit.

e) Interpret the odds ratio for this logistic model.

f) Use simulation-based and theory-based tests to decide whether there is a statistically significant relationship between age and survival.

g) Give and interpret a 95% confidence interval for the odds ratio of survival. (Note that this says the odds ratio and NOT the log(odds).)

6.9) The variability of the GPAs of first-year students is an important area of study for many college administrators. The `FirstYearGPA.csv` file in my Public folder contains data on 219 randomly selected college students. One of the variables in the dataset is `FirstGen`, which indicates whether the student was a first-generation college attendee (1 if so, 0 if not). In this exercise, you will compare several ways of seeing if there is some association between GPA and $FirstGen$.

a) Use a two-sample simulation-based test to see if there is a significant difference in the average GPA between students who are and are not first-generation college attendees. (This will require you to take the WAY-BACK machine to near the beginning of the course). Report the p-value of the test and the direction of the relationship, if significant, using a confidence interval.

b) Use simple linear regression to predict GPA using $FirstGen$. ($FirstGen$ is a dummy variable in this case. We haven't done any examples explicitly like this but you should be able to think carefully about the interpretation of the fit.) Report a simulation-based p-value and discuss the conclusion of a test to see if $FirstGen$ is a significant predictor of GPA.

c) Do the comparison once more, this time using a logistic regression with GPA as the predictor and $FirstGen$ as the response. Compare the conclusion (and p-value) you would draw from this model to the results from parts (a) and (b).

6.10) A 1994 study collected data on the effects of air traffic on the behavior of the Pacific Brant (a small migratory goose). Each fall, nearly the entire population of 130,000 of this species uses the Izembek Lagoon in Alaska as a staging area, where it feeds and stores fat for its southerly migration. Because offshore drilling near this estuary had increased the necessity of air traffic, an impact study was timely. The data represent the flight response to helicopter "overflights" to see what the relationship between the proximity of a flight, both lateral and altitudinal, would be to the propensity of the Brant to flee the area. For this experiment, air traffic was restricted to helicopters because a previous study had ascertained that helicopters created more radical flight response than other aircraft.

The data are in the `FlightResponse.csv` file in my Public folder. Each case represents a flock of Brant that has been observed during one overflight in the study. Flocks were determined observationally as contiguous collections of Brants, flock sizes varying from 10 to 30,000 birds. For this study the variables we investigate are:

- *Altitude*: The experimentally determined altitude of the overflight by the helicopter. Units are in 100 meters.

- *Lateral*: The perpendicular or lateral distance (in 100 meters) between the aircraft and flock, as determined from studying area maps to the nearest 0.16 kilometer.
- *Flight*: This is a binary variable in which 0 represents an outcome where fewer than 10% of a flock flies away during the overflight and 1 represents an outcome where more than 10% of the flock flies away. This is the response variable of interest in this study.

a) Calculate a logistic regression model using *Flight* as the response variable and *Altitude* as the explanatory variable. Does this model confirm your suspicion about the existence and direction of a relationship between flight and altitude?

b) Report model estimates and interpret the estimated slope coefficient for odds. Also give the probability form of the model.

c) Test for the significance of the predictor in the model using randomization-based methods and theory-based methods. Compare and contrast the results of both.

d) Rework parts (a)-(c) using *Lateral* as the predictor and again using *Flight* as the response.

Chapter 7

Multiple Logistic Regression

In the last chapter, we found that GPA is a useful predictor of admission to medical school. What if you also know a student's MCAT score: Does that additional information lead to a better prediction? Or can you ace the MCAT and not worry about your GPA's effect on your chance of admission? Notice that each of these last two questions involves a pair of predictors and a binary response. Here's a second example of a similar structure: Does the size of campaign contributions (first predictor) affect the way U.S. Senators vote (binary response)? If so, is the relationship the same for Democrats and Republicans (second predictor)?

In this chapter, we consider models for datasets like these, fitting a linear predictor of the form $\beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k$ to the log odds for a binary response variable. As you will see, working with these models has a lot in common with things you have learned already. To help you take advantage of that overlap, we start with a preview that highlights the parallels. The hope is that seeing the big picture in advance will make it easier to learn the details.

7.1 Overview

In Chapter 4, we considered the simple linear regression model for a quantitative response based on a single quantitative predictor and, in Chapter 5, we extended this model to include multiple predictors. Having introduced logistic regression for a binary response with a single predictor in Chapter 6, we are now in a position to extend that model to allow for multiple predictors. Although the specific techniques differ from the ordinary multiple regression setting, the issues that we need to address in order to choose, fit, assess, and use a binary logistic model with multiple predictors should seem familiar.

Four Different Regression Models

Variables	**Predictor**	**Linear Regression**	**Logistic Regression**
One: X	$\beta_0 + \beta_1 X$	Response y	$\text{logit}(\pi) = log(\pi/(1-\pi))$
Several: $X_1, X_2, \ldots, X_k$	$\beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k$	Response y	$\text{logit}(\pi) = log(\pi/(1-\pi))$

The table above gives a quick comparison of simple and multiple, linear and logistic regression. Just as in Chapter 6 with simple logistic regression, there are two forms of the model, depending

on whether the left-hand side is the probability π or the log(odds) $= log(\pi/(1-\pi))$.

Definition 7.1 (Multiple Logistic Regression Model)**.** *The two forms of the multiple logistic regression model are just extensions of the simple logistic model:*

- *Logit form:*
$$log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k$$
- *Probability form:*
$$\pi = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_k X_k}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_k X_k}}$$

We have seen throughout the course that the predictors can be in various forms, including

- Individual quantitative variables such as GPA or $Weight$
- Functions (such as squares or log transformations) of quantitative variables
- Categorical factors coded as indicator/dummy variables

We can also look at interactions between different variables by looking at the multiplications of different variables and treating that multiplication as a new variable. We conclude our overview with a preview of how the four-step process we have discussed throughout these course notes will work for multiple logistic regression.

***Choosing* and *Fitting* the model**

Which predictors should you include? Just as with multiple linear regression, the answer depends on the data you have and on the goals of your analysis. Also, as with multiple linear regression, you can use hypothesis tests and confidence intervals to assess model utility. We will introduce in this chapter a way to check for the effectiveness of one logistic model compared to another in ways similar to the use of the adjusted R^2 value in Chapter 5. For each model that we consider, we will use `R` to get fitted coefficients.

***Assessing* the model**

1. To assess model utility, we will use the concept of **deviance** and compare the values of deviance from various fitted models to help decide which predictors to include. In particular, we can use a "drop-in deviance" test—the change in deviance—to compare a pair of models.
2. To prepare for inferential techniques (especially the theory-based, traditional ones), there are two sets of conditions to check, linearity of the logistic relationships and the validity of the probability model.
    - *Linearity.* A strong linear relationship is needed if the model is to fit well. For linear regression, linearity refers to the relationship between a predictor and the response. For logistic regression, linearity refers to the relationship between a predictor and the logit-transformed response. We check for linearity using empirical logit plots, whenever possible. Remember that this condition is not necessarily easy to check in many situations.

- *Probability model.* For tests and confidence intervals to be valid, the response values must be random and independent. Formally, the Bernoulli model must be reasonable. Informally, the outcomes must behave as if they were generated using a spinner, or unfair coin, with probabilities given by the logistic equation. To check this part of the model, you have to think carefully about the method used to produce the data.

***Using* the model**

Just as with multiple linear regression, we can test hypotheses and construct confidence intervals, provided the conditions for inference are satisfied. We can use hypothesis tests to find p-values to measure statistical significance. We can use theory-based methods in the assumptions are met or simulation-based methods to calculate these p-values based on the observed statistics. Confidence intervals can also be found to assess the extent of the effect of a certain variable on the model. In what follows, Section 7.2 introduces a variety of models in the context of campaign contributions and Senate votes by party and shows how to assess how well the models fit, and Section 7.4 summarizes and illustrates methods for inference.

7.2 Choosing, Fitting, Interpreting, and Assessing Logistic Models

Example 7.1. The Corporate Average Fuel Economy (CAFE) bill was proposed by Senators John McCain and John Kerry to improve the fuel economy of cars and light trucks sold in the United States. However, a critical vote on an amendment in March 2002 threatened to indefinitely postpone CAFE. The amendment charged the National Highway Traffic Safety Administration to develop a new standard, the effect being to put an indefinite hold the McCain-Kerry bill. It passed by a vote of 62-38.

A political question of interest is whether there is evidence of monetary influence on a senator's vote. Scott Preston, a professor of statistics at SUNY, Oswego, collected data on this vote, which includes our response variable, the vote of each senator (Yes or No), and as an explanatory variable, monetary contributions that each of the 100 senators received over his or her lifetime from car manufacturers.

Anyone with an interest in U.S. politics might naturally be led to ask the following questions with which we will proceed in answering:

- What is the effect of party affiliation on the vote in the CAFE amendment context?
- How different are the votes of Republican and Democratic senators and to what extent is the effect of contributions dependent on party affiliation?

Fitting a Multivariate Logistic Regression

The model in this case corresponds to the log odds of a Yes vote as the response and `Contribution` and `Democrat?` as the explanatory variables. We can denote that with the following equation:

$$log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 Contribution + \beta_2 Dem.$$

We will next read in the dataset and then get estimates for the coefficients of the model. (Note that this dataset has been modified slightly to create a dichotomous outcome for the `Party` variable. The modification comes with Senator James Jeffords of Vermont, who was a Republican until 2001, when he left the party to become an Independent and began caucusing with the Democrats. His `Party` has, thus, been changed to a `D` from an `I`.)

```
> cafeSen <- read.csv("/shared/ismayc@ripon.edu/CAFE.csv")
> head(cafeSen)
           Senator State Party Contribution LogContr Dem Vote
1 Murkowski, Frank    AK     R        19700 4.305351   0    1
2     Stevens, Ted    AK     R        13000 4.130334   0    1
3   Sessions, Jeff    AL     R         9500 4.000000   0    1
4  Shelby, Richard    AL     R        25000 4.406540   0    1
5  Hutchinson, Tim    AR     R         4900 3.732394   0    1
6 Lincoln, Blanche    AR     D         5500 3.778151   1    1
```

Note: Because some of the lifetime contributions are zero, the log transformation cannot be applied to those values. When, as here, all values are nonnegative but some are zero, the standard practice is to add 1 to all values before taking logs. Throughout the entire example, "LogContrib" and "log Contribution" refer to the natural log of (1 + Contr).

```
> cafeSen$LogContrib <- log(1+cafeSen$Contribution)
> head(cafeSen)
           Senator State Party Contribution LogContr Dem Vote LogContrib
1 Murkowski, Frank    AK     R        19700 4.305351   0    1   9.888425
2     Stevens, Ted    AK     R        13000 4.130334   0    1   9.472782
3   Sessions, Jeff    AL     R         9500 4.000000   0    1   9.159152
4  Shelby, Richard    AL     R        25000 4.406540   0    1  10.126671
5  Hutchinson, Tim    AR     R         4900 3.732394   0    1   8.497195
6 Lincoln, Blanche    AR     D         5500 3.778151   1    1   8.612685
> cafeModel <- glm(Vote ~ LogContrib + Dem, family=binomial, data=cafeSen)
> cafeModel
Call:  glm(formula = Vote ~ LogContrib + Dem, family = binomial, data = cafeSen)

Coefficients:
(Intercept)   LogContrib          Dem
    -2.5547       0.4833      -1.9012

Degrees of Freedom: 99 Total (i.e. Null);  97 Residual
Null Deviance:            132.8
Residual Deviance: 93.23         AIC: 99.23
```

The coefficient estimates give the following prediction equation:

$$log\left(\frac{\pi}{1-\pi}\right) = -2.55 + 0.48LogContrib - 1.90Dem.$$

The equivalent probability form is

$$\pi = \frac{e^{-2.55+0.48LogContrib-1.90Dem}}{1+e^{-2.55+0.48LogContrib-1.90Dem}}$$

For example, a Democratic senator with $50,000 in lifetime contributions from the car industry, that is, $\log(1+Contr) = 10.82$, would be predicted to have log odds of voting Yes on the CAFE amendment of

$$log\left(\frac{\hat{\pi}}{1-\hat{\pi}}\right) \approx -2.55 + 0.48(10.82) - 1.90(1) \approx 0.744$$

which gives a probability of about 2/3:

```
> exp(0.744)/(1+exp(0.744))
[1] 0.6778699
```

$$\hat{\pi} = \frac{e^{0.744}}{1+e^{0.744}} \approx 0.678.$$

A Republican senator with the same contribution would have

$$log\left(\frac{\hat{\pi}}{1-\hat{\pi}}\right) = -2.55 + 0.48(10.82) - 1.90(0) \approx 2.644$$

and

```
> exp(2.644)/(1+exp(2.644))
[1] 0.9336402
```

$$\hat{\pi} = \frac{e^{2.644}}{1+e^{2.644}} = 0.934.$$

Just as for logistic models with a single predictor, coefficient estimates translate to odds ratios. For example, exponentiating the logit form of the model gives

$$\frac{\hat{\pi}}{1-\hat{\pi}} = e^{-2.55+0.48log(1+Contr)-1.90Dem}.$$

We can see that each additional dollar of contribution multiplies the odds of voting Yes by a factor of $e^{0.48}$, or about 1.6. Alternatively, doubling a senator's contribution adds $\log(2) = 0.69$ to the log contribution, which multiplies the log odds of Yes by a factor of $e^{0.48*0.69} = e^{0.33}$, or about 1.4. Note that these calculations make sense only if we assume that the party doesn't change. Just as in ordinary multiple linear regression, the interpretation of the coefficient for one variable is conditional on all the other variables being kept constant.

Checking the Probability Model

Suppose you have chosen a scale that makes the relationships linear so that the linearity condition is met. What next? Typically, we want our analysis to head to head toward inference after we have fit our model. R spits out numbers at the click of a mouse. Why not go ahead?

The numbers themselves can't hurt you, but if you aren't careful, taking the numbers at face value definitely can hurt you. The methods of inference rely on a probability model for the data. Specifically, just as in Chapter 6, the Yes or No outcomes must behave according to the Bernoulli

(spinner) model. We, thus, need to check randomness and independence, by thinking about the process that created our data. The check of these conditions is essentially the same as in Chapter 6, regardless of how many predictors we have, because the main focus is on the response, not on the predictors.

Example 7.2. We now will think about checking the probability model for the CAFE data.

Randomness. The U.S. Senate is in no way a random sample, so we cannot appeal to sampling as a basis for regarding the outcome as random. Party affiliation and contributions are not randomly assigned, so we cannot appeal to the theory of experimentation. Do senators decide which way to vote using a Yes or No spinner? The answer is surely "No" (cynics to the contrary notwithstanding).

More to the point, how reasonable is it to think of the spinner model as an approximation to reality? This is a vastly more difficult question, because the right version to ask is *conditional*: *Given* a party affiliation and campaign amount, do senators' Yes or No votes behave like outcomes from a spinner? Data can't help us here because, for any given combination of party and contribution, the sample size is tiny, typically $n = 1$. Bottom line: Although there is no clear basis for saying that the spinner model does apply, there is also no clear basis for saying it does not apply.

Independence. If outcomes are not random, then independence is not an issue because independence has no meaning without probabilities. However, to the extent that randomness might be reasonable as an approximate model, it is important to think also about independence.

Without independence, the formulas for standard errors and the p-value calculations will be wrong. Here, as with randomness, there is a naive version of the question with a simple answer, and a subtler and harder version of the question. The simple version ignores conditioning and asks, "If you know one senator's vote, does that help you predict any other senator's vote?" A quick look at the votes state-by-state gives an equally quick answer of "Yes." Senators from the same state tend to vote the same way.

If you do not condition on party and contribution amount, votes are clearly *not* independent. What matters, however, is the *conditional* version of the question, which is the more subtle version: If two senators belong to the same party and receive the same campaign amounts, are their votes independent?

There are many ways that independence can fail, but most failures result from lurking variables—one or more shared features that are not part of the model. One way to check independence is to use the applied context to guess at a possible lurking variable, then compare predictions with and without that variable in the model.

For the CAFE data, one possible lurking variable is a senator's state. It is reasonable to think that senators from the same state tend to vote the same way more often than predicted by the independence condition. This prediction is one we can check: For each senator, we use our model to compute a fitted $P(Yes)$ given party and contribution level. We then use these fitted probabilities to compute the chance that both senators from a state vote the same way, *assuming that votes are independent.* This lets us compute the expected number of states where both senators vote the same way, if the independence condition holds. We can then compare the expected number with the observed number. If votes are in fact independent, observed and expected should be close. The table below shows the actual numbers.

	Together	**Split**	***Total***
Actual	38.0	12.0	50.0
If independent	31.1	18.9	50.0

Observed and expected are far enough apart to call into question the condition of independence. (This can be tested using a chi-square test and does produce a statistically significant result.) The evidence is all the stronger because the direction of the difference supports the idea that senators from the same state tend to vote the same way, even after you adjust for party and campaign contributions.

Where does this leave us in relation to the probability model? Agnostic at best. We have no clear basis to justify the probability part of the model, and some very substantial basis for thinking the model is incomplete. At the same time, we have no compelling basis for declaring the model to be so clearly wrong as to make tests and intervals also clearly wrong.

It is reasonable to go ahead with inference but to be prudent to regard the results as tentative guidelines only. This caveat aside, it is important to recognize that, even without formal inference, the logistic model still has great value. It fits well, and it provides a simple, accurate, and useful description of a meaningful pattern. In particular, the fitted model is consistent with the research hypothesis that campaign contributions and votes on the CAFE bill are related.

Checking conditions is the hard part of inference, as the last example illustrates. You really have to think. By comparison, getting the numbers for inference is often a walk in the park. Keep in mind, however, as you go through the next section, that what matters is not the numbers themselves, but what they do tell you, and what they cannot tell you, even if you want them to.

7.3 Inference for Multiple Logistic Regression

Inference for multiple logistic regression is in many ways quite similar to inference for multiple linear regression. For both kinds of regression, we have tests and confidence intervals for individual coefficients in the model. We have and will continue to use simulation-based and theory-based methods to check for the significance of the coefficients, which in turn checks for the signficance of predictors in our models.

In this section, we will also introduce the concept of **deviance**, which we will use as a way to compare the effectiveness of one logistic model to another. This is similar to what was done when we compared adjusted R^2 values in the multiple linear regression sense. We will also discuss the role that the interaction of predictors can have on models and check for significance of these interaction terms in our models.

7.3.1 Tests and Intervals for Individual Coefficients

7.3.1.1 Simulation-Based Methods

We begin our discussion of inference by using the same types of methods we have throughout these notes. We will focus on using randomization again but we could have used bootstrapping if we would have liked. Let's revisit the CAFE data testing whether log(Contributions) and political party belong in the model.

We proceed in a similar way to what was done in the simple logistic regression case. We simply shuffle the response variable over the different inputs of the explanatory variables and see where our original estimated coefficient falls on the simulated distributions of coefficients.

Recall that we can get the coefficients of our model fit via the following commands:

```
> cafeModel <- glm(Vote ~ LogContrib + Dem, family=binomial, data=cafeSen)
> cafeModel

Call:  glm(formula = Vote ~ LogContrib + Dem, family = binomial, data = cafeSen)

Coefficients:
(Intercept)   LogContrib          Dem
    -2.5547       0.4833      -1.9012

Degrees of Freedom: 99 Total (i.e. Null);  97 Residual
Null Deviance:            132.8
Residual Deviance: 93.23         AIC: 99.23
> LogContribCoef <- coef(cafeModel)[[2]]; LogContribCoef
[1] 0.4832521
> DemCoef <- coef(cafeModel)[[3]]; DemCoef
[1] -1.901202
```

We now want to see if both of these coefficients are statistically different from zero. In other words, we have a multiple binary logistic regression model with $k = 2$ predictors

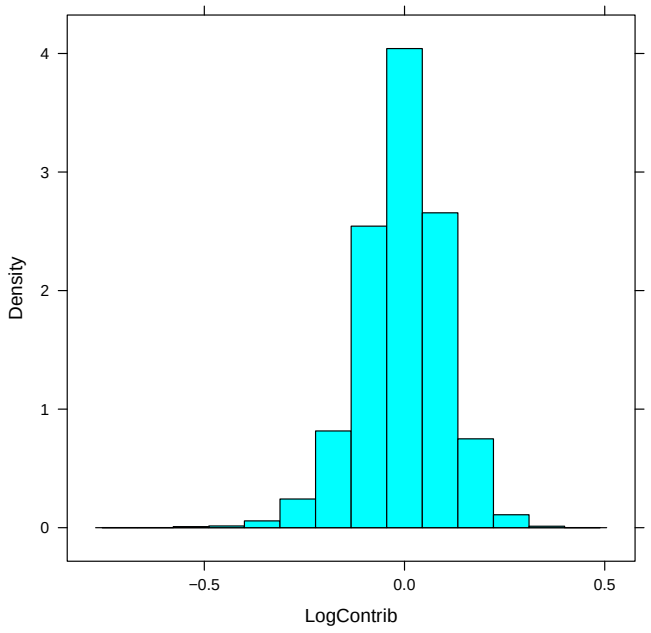
$$log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k.$$

We want to test for the contribution of the predictor X_i, given the other predictors in the model, using $H_0 : \beta_i = 0$ versus $H_0 : \beta_i \neq 0$.

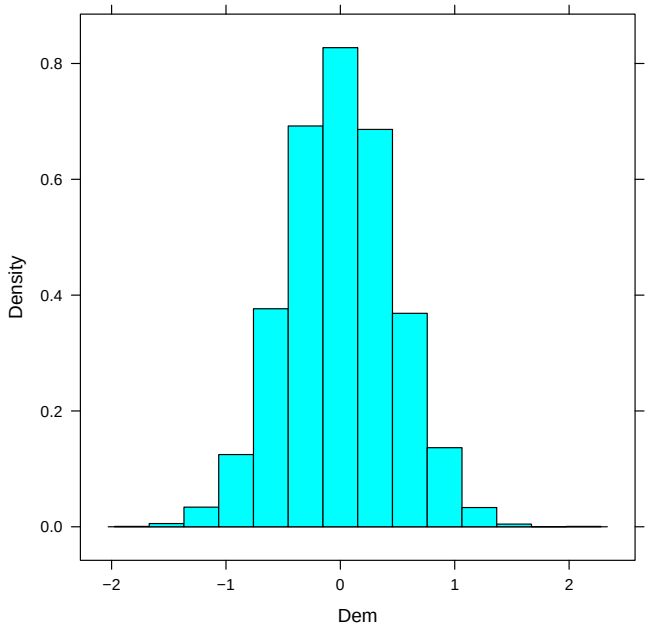
```
> cafeSim <- do(10000) * glm(shuffle(Vote) ~ LogContrib + Dem,
+                            family=binomial, data=cafeSen)
> head(cafeSim)
    Intercept  LogContrib          Dem
1 -0.09915188  0.09657783 -0.427524976
2  0.65825200 -0.02117411  0.022363894
3  0.07066605  0.04973697 -0.003389591
4  1.21540504 -0.03142108 -0.861632450
5  0.02979293  0.01839533  0.615577761
6 -0.80032342  0.13226870  0.341846783
```

We now look at the histograms for both of the simulated coefficents. Remember that the histograms are centered at zero since that is the hypothesized value of the coefficient.

```
> histogram(~LogContrib, center=0, data=cafeSim)
```




```
> histogram(~Dem, center=0, data=cafeSim)
```



After looking at the histograms, you should be able to give a good guess as to the size of the p-values based on how extreme the observed coefficients are in the distributions. We calculate the p-values for each of the tests below. Remember that these p-values are two-tailed.

```
> pvalLogContrib <- prop(~abs(LogContrib) >= LogContribCoef, data=cafeSim)
> pvalLogContrib
 TRUE
0.001
```

The p-value on `LogContrib` is extremely small at 0.001, which is clearly smaller than any reasonable value of a significance level α. This gives us evidence that `LogContrib` is an important predictor in the model.

```
> pvalDem <- prop(~abs(Dem) >= -DemCoef, data=cafeSim)
> pvalDem
 TRUE
2e-04
```

We see that this p-value is also small. We can, thus, say that both of the predictors belong in this multiple logistic regression model. More formally, to the extent that outcomes can be regarded as random (plausible) and independent (questionable at best), we have evidence to reject the hypothesis that there is no association between `Vote` and the two predictor variables.

We next compute confidence intervals for these coefficients using the standard deviation of the simulated distributions. Recall that we can check for zero being in the confidence interval or for one being in the confidence interval for the estimated odds ratio for a one-unit change in each predictor.

```
> sdLogContrib <- sd(~LogContrib, data=cafeSim); sdLogContrib
[1] 0.1049347
> sdDem <- sd(~Dem, data=cafeSim); sdDem
[1] 0.4689408
```

With our simulated distributions being bell-shaped, we can construct approximate 95% confidence intervals using the formula

$$\hat{\beta}_i \pm (2 \cdot \hat{SE}_{\beta_i})$$

where $\hat{SE}_{\beta_i}$ corresponds to the standard deviation of the simulated distribution for the coefficient in question.

```
> lowerLogContrib <- LogContribCoef - 2*sdLogContrib; lowerLogContrib
[1] 0.2733827
> upperLogContrib <- LogContribCoef + 2*sdLogContrib; upperLogContrib
[1] 0.6931214


> lowerDem <- DemCoef - 2*sdDem; lowerDem
[1] -2.839084
> upperDem <- DemCoef + 2*sdDem; upperDem
[1] -0.9633205
```

With zero not being in either confidence interval, we have further evidence that both of the predictors are significant in predicting `Vote`. Recall that these confidence intervals are on the logit scale though and are often much harder to interpret. If we transform our intervals to the odds scale, we can better understand what a one-unit increase means in the odds of success, which in this case means the odds of voting Yes to the measure.

```
> lowerOddsLC <- exp(lowerLogContrib); lowerOddsLC
[1] 1.314403
```

```
> upperOddsLC <- exp(upperLogContrib); upperOddsLC
[1] 1.999949
```

Thus, we are 95% confident that the odds of voting Yes on the measure increases from 1.3144 to 1.99995 based on a one-unit increase in `LogContrib`.

```
> lowerOddsDem <- exp(lowerDem); lowerOddsDem
[1] 0.05847924
> upperOddsDem <- exp(upperDem); upperOddsDem
[1] 0.3816236
```

We are 95% confident that the odds of voting Yes on the measure goes from 0.05848 to 0.38162 based on switching from Republican to Democrat in the model. With these values being less than one, this corresponds with the problem statement that Democrats are less likely to vote Yes on the measure than Republicans. If a Democrat ($Dem = 1$) and Republican ($Dem = 0$) have the same lifetime contribution, the interval, taken at face value, says that the odds of the Democrat voting Yes is between 0.05 and 0.45 times the odds of the Republican voting Yes.

7.3.1.2 Theory-Based Methods

Assuming that the assumptions for theory-based methods are met, we can also use normal-based theory to test for the significance of the predictors in the model. Remember that theory-based methods are often easier to conduct in `R` but it might not always be the case that the assumptions required to run the methods are met. It's often better to be on the safe side and use simulation based methods to check to see if they correspond with the theory-based results.

The theory-based methods for multiple logistic regression models use what is known as the Wald z-statistic, which is defined as

$$z = \frac{\hat{\beta}_i}{SE_{\beta_i}}$$

where the standard error of the coefficient here is based on a normal distribution. We can calculate these values based on the `summary()` function as we have used before. If we don't want to look at all of the output from `summary`, we can again use the `coef()` function to just look at the tests on the coefficients. We also use the `confint()` function to compute the theory-based confidence intervals based on the logit model.

```
> coef(summary(cafeModel))
             Estimate Std. Error   z value     Pr(>|z|)
(Intercept) -2.5547416  1.8651317 -1.369738 0.1707687117
LogContrib   0.4832521  0.1964246  2.460242 0.0138843498
Dem         -1.9012020  0.5594033 -3.398625 0.0006772548
> confint(cafeModel)
                2.5 %     97.5 %
(Intercept) -6.590346  0.6331929
LogContrib   0.157270  0.9136299
Dem         -3.063969 -0.8428995
```

We see that these confidence intervals deviate some from our simulated-based intervals but the overall conclusions are the same: the predictors are important in the model.

7.3.2 Model Comparing and Choosing

Recall that estimates in the ordinary linear regression setting were obtained by minimizing the sum of squared residuals. In logistic regression, we choose coefficients to minimize a different quantity (-2 log L) that behaves in much the same way. Comparing values of -2 log L for models with and without the predictors or combinations of the predictors and interaction terms serves as a basis for measuring the quality of each of the different model fits.

What is -2 log L? A detailed explanation of this term would take us out of the realm of the course but a bare-bones definition follows.

Definition 7.2 (Likelihood, Maximum Likelihood, and Deviance)**.** *The* **likelihood** *of the data, denoted L, is the probability of the data, regarded as a function of the unknown parameters with the data values fixed. (This is parallel to sums of squares being regarded as a function of the parameters with data values held fixed for linear regression.)*

The **method of maximum likelihood** *chooses parameter values to maximize L, or, equivalently, to minimize -2 log L, which is called the* **deviance***. The deviance behaves similarly to the residual sums of squares in linear regression.*

We can use the value of -2 log L for each of the different models we would like to fit to conduct a likelihood ratio test for the effectiveness of one model compared to another.

Definition 7.3 (Drop-in-Deviance Test (Nested Likelihood Ratio Test))**.** *A test for the overall effectiveness of the model*

$$H_0 : \beta_1 = \beta_2 = \cdots \beta_k = 0 \text{ versus } H_a : \text{ A least one } \beta_i \neq 0$$

uses the test statistic

$$G = -2log(L_0) - [-2log(L)]$$

where L_0 is the likelihood under the model $logit(\pi) = \beta_0$ and L is the likelihood under the larger model. The p-value for this test comes from the upper tail of a χ^2 distibution with k degrees of freedom.

The **nested likelihood ratio test (LRT)** *works the same way, but with more general hypotheses.*

$$H_0 : \text{ Reduced model versus } H_a : \text{ Full model}$$

The nested G-statistic is the difference in $-2log(L)$ between the two models. If the reduced model has k_1 predictors and the full model has $k_1 + k_2$ predictors, then the p-value comes from the upper tail of a χ^2-distribution with k_2 degrees of freedom.

Example 7.3. We now go back to the GPA, $MCAT$, and acceptance to medical school example.

```
> medGPA <- read.csv("/shared/ismayc@ripon.edu/MedGPA.csv")
> head(medGPA)
  Accept Acceptance Sex BCPM  GPA VR PS WS BS MCAT Apps
1      D          0   F 3.59 3.62 11  9  9  9   38    5
2      A          1   M 3.75 3.84 12 13  8 12   45    3
3      A          1   F 3.24 3.23  9 10  5  9   33   19
```

```
4       A           1    F 3.74 3.69 12 11  7 10   40    5
5       A           1    F 3.53 3.38  9 11  4 11   35   11
6       A           1    M 3.59 3.72 10  9  7 10   36    5
```

We will focus on two different models as follows. We look to answer the question, "If GPA is already in your model, can you get better predictions if you also include $MCAT$ in your model?" For this question, our two nested models are

$$H_0 : logit(\pi) = \beta_0 + \beta_1 GPA$$
$$H_a : logit(\pi) = \beta_0 + \beta_1 GPA + \beta_2 MCAT$$

Here, H_0 corresponds to the reduced model and H_a to the full model. To carry out the LRT, we need the values of deviance = -2 log (L) for both models. We compute the difference in values and refer that difference to a chi-square distribution with 1 degree of freedom. The $df = 1$ comes from subtracting the numbers of parameters in the two models: 3(full) - 2(reduced). Here is some output for the two models:

Reduced model:

```
> summary(glm(Acceptance ~ GPA, data=medGPA, family=binomial))
Call:
glm(formula = Acceptance ~ GPA, family = binomial, data = medGPA)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7805  -0.8522   0.4407   0.7819   2.0967

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -19.207      5.629  -3.412 0.000644 ***
GPA            5.454      1.579   3.454 0.000553 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 75.791  on 54  degrees of freedom
Residual deviance: 56.839  on 53  degrees of freedom
AIC: 60.839

Number of Fisher Scoring iterations: 4
```

Full model:

```
> summary(glm(Acceptance ~ GPA + MCAT, data=medGPA, family=binomial))
```

```
Call:
glm(formula = Acceptance ~ GPA + MCAT, family = binomial, data = medGPA)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7132  -0.8132   0.3136   0.7663   1.9933

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -22.3727     6.4538  -3.467 0.000527 ***
GPA           4.6765     1.6416   2.849 0.004389 **
MCAT          0.1645     0.1032   1.595 0.110786
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 75.791  on 54  degrees of freedom
Residual deviance: 54.014  on 52  degrees of freedom
AIC: 60.014

Number of Fisher Scoring iterations: 5
```

The value of -2 log (L) is called "Residual deviance" in the `R` output: 56.839 for the reduced model and 54.014 for hte full model. Subtracting gives a "drop-in-deviance" of 2.825. Referring this to a chi-square distribution with $df = 3 - 2 = 1$ gives the following p-value. Notice that we set the `lower.tail` option to `FALSE` so that we compute the area under the curve in the right-tail of the distribution.

```
> pchisq(56.839 - 54.014, df=1, lower.tail=FALSE)
[1] 0.09280689
```

The conclusion here is not clear-cut. The p-value is not below 0.05, but is close, at 0.09. If the goal is a formal test of H_0 versus H_a, we fail to reject H_0 (at the 0.05 level), but that "failure" could easily be a consequence of a small sample size. If the goal is to find a useful model, it may well make sense to include $MCAT$ in the model. As an additional check, we can use a nested LRT to check overall utility for the model that included both GPA and MCAT to the model that includes no predictors.

To assess the utility of the model $logit(\pi) = \beta_0 + \beta_1 GPA + \beta_2 MCAT$, we compare the residual deviance of 54.014 for that model with a residual deviance of 75.791 for the null model $logit(\pi) = \beta_0$. The drop-in-deviance is 75.791 - 54.014 = 21.777, with degrees of freedom = 2 because the full model has 3 coefficients and the null model has 1. We obtain the p-value as follows:

```
> pchisq(75.791 - 54.014, df=2, lower.tail=FALSE)
[1] 1.867173e-05
```

This p-value is tiny and we, thus, have evidence that the two predictor model is effective in predicting the log odds of acceptance to medical school.

We now discuss model selection with this medical school acceptance problem. We also introduce interaction terms in the model, how to add them into the model using R, and a more thorough use of the drop-in-deviance tests.

Example 7.4. One way to build a model is to start with the null model $logit(\pi) = \beta_0$ and add predictors one step at a time. As you will see, this method, though logical, is not easy to make automatic. For this example, using the data for medical school admissions, we start by using nested LRTs to choose one predictor from among GPA, $MCAT$, and FEM. FEM is created by converting the F/M in Sex to 1/0.

At each step after that, we use a new set of nested tests to find the most useful predictor from among those available, and decide whether to add it to the model. If at some point we include a quantitative predictor such as GPA, at the next step we include GPA^2, that is, the square of GPA, as a possible predictor; if at any point we have included a pair of terms, at the next step we include their interaction as a possible predictor.

```
> medGPA2 <- medGPA
> medGPA2$FEM <- ifelse(medGPA$Sex=="F", 1, 0)
> head(medGPA2)
  Accept Acceptance Sex BCPM  GPA VR PS WS BS MCAT Apps FEM
1      D          0   F 3.59 3.62 11  9  9  9   38    5   1
2      A          1   M 3.75 3.84 12 13  8 12   45    3   0
3      A          1   F 3.24 3.23  9 10  5  9   33   19   1
4      A          1   F 3.74 3.69 12 11  7 10   40    5   1
5      A          1   F 3.53 3.38  9 11  4 11   35   11   1
6      A          1   M 3.59 3.72 10  9  7 10   36    5   0
```

First predictor? Our null model is $logit(\pi) = \beta_0$. To choose a first predictor, we carry out three nested tests, one for each of GPA, $MCAT$, and FEM. We use methodology similar to what was done in the previous example to get the following results. Please ensure that you can get the values in the table using the appropriate commands in R.

		-2 log(L)		
Predictor	**Reduced**	**Full**	**Difference**	***p*-value**
GPA	75.791	56.839	18.952	0.0000134
MCAT	75.791	64.697	11.094	0.000866
FEM	75.791	73.594	2.197	0.138

The biggest drop-in-deviance is for GPA, with a tiny p-value, so we add that predictor to our model to get a new reduced model, $logit(\pi) = \beta_0 + \beta_1 GPA$.

*S*econd predictor? Now that GPA is in the model, we allow GPA^2 as a possible predictor. We have three new nested LRTs, one each for $MCAT$, FEM, and GPA^2. Remember that the code for adding the GPA^2 variable looks something like

```
> gpa2Model <- glm(Acceptance ~ GPA + I(GPA^2), family=binomial, data=medGPA2)
> summary(gpa2Model)
Call:
glm(formula = Acceptance ~ GPA + I(GPA^2), family = binomial,
    data = medGPA2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8363  -0.8020   0.3207   0.7830   1.9553

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   33.332     45.275   0.736    0.462
GPA          -24.752     26.366  -0.939    0.348
I(GPA^2)       4.325      3.832   1.128    0.259

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 75.791  on 54  degrees of freedom
Residual deviance: 55.800  on 52  degrees of freedom
AIC: 61.8

Number of Fisher Scoring iterations: 4
```

and we will be comparing this and the two other models to the new Reduced model given by

```
> gpaModel <- glm(Acceptance ~ GPA, family=binomial, data=medGPA2)
> summary(gpaModel)
Call:
glm(formula = Acceptance ~ GPA, family = binomial, data = medGPA2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7805  -0.8522   0.4407   0.7819   2.0967

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -19.207      5.629  -3.412 0.000644 ***
GPA            5.454      1.579   3.454 0.000553 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 75.791  on 54  degrees of freedom
Residual deviance: 56.839  on 53  degrees of freedom
AIC: 60.839

Number of Fisher Scoring iterations: 4
```

	Additional Predictor	-2 log(L) Reduced	-2 log(L) Full	Difference	p-value
GPA +	$MCAT$	56.839	54.014	2.825	0.090
	FEM	56.839	53.945	2.894	0.089
	GPA^2	56.839	55.800	1.039	0.308

Clearly, we don't want to add GPA^2 at this point. Even though the p-values for $MCAT$ and FEM are above 0.05, they are close enough to 0.05 to make it worth seeing what happens with one or both of them in the model. Which one to include is somewhat arbitrary, but because the $MCAT$ was designed to predict success in medical school, choosing to include $MCAT$ is reasonable on the basis of context. This choice gives a new reduced model, $logit(\pi) = \beta_0 + \beta_1 GPA + \beta_2 MCAT$.

Third predictor? With GPA and $MCAT$ in the model, we now expand our list of possible predictors to include both quadratic terms GPA^2 and $MCAT^2$ and the interaction $GPA \cdot MCAT$, along with FEM.

Note: We add the interaction in R as one might suspect using the following code.

```
> medAccModel <- glm(Acceptance ~ GPA + MCAT + I(GPA*MCAT), family=binomial,
+                    data=medGPA2)
> summary(medAccModel)
Call:
glm(formula = Acceptance ~ GPA + MCAT + I(GPA * MCAT), family = binomial,
    data = medGPA2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7790  -0.8048   0.1937   0.7694   1.9347

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   17.0632    34.1200   0.500    0.617
GPA           -6.6350    10.0837  -0.658    0.511
MCAT          -0.9359     0.9737  -0.961    0.336
I(GPA * MCAT)  0.3154     0.2864   1.101    0.271

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 75.791  on 54  degrees of freedom
Residual deviance: 53.186  on 51  degrees of freedom
```

```
AIC: 61.186

Number of Fisher Scoring iterations: 5
```

Remember that we are now comparing values to the model with predictors GPA and $MCAT$:

```
> gpaMCATModel <- glm(Acceptance ~ GPA + MCAT, family=binomial, data=medGPA2)
> summary(gpaMCATModel)
Call:
glm(formula = Acceptance ~ GPA + MCAT, family = binomial, data = medGPA2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7132  -0.8132   0.3136   0.7663   1.9933

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -22.3727     6.4538  -3.467 0.000527 ***
GPA           4.6765     1.6416   2.849 0.004389 **
MCAT          0.1645     0.1032   1.595 0.110786
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 75.791  on 54  degrees of freedom
Residual deviance: 54.014  on 52  degrees of freedom
AIC: 60.014

Number of Fisher Scoring iterations: 5
```

	Additional		***-2 log(L)***		
	Predictor	**Reduced**	**Full**	**Difference**	p**-value**
GPA + MCAT +	GPA^2	54.014	53.235	0.779	0.337
	$MCAT^2$	54.014	54.005	0.009	0.925
	$GPA \cdot MCAT$	54.014	53.186	0.828	0.363
	FEM	54.014	50.786	3.228	0.072

The drop-in-deviance tests reveal a mild surprise: the p-value for FEM is lower with $MCAT$ in the model than it was with $MCAT$ *not* in the model. None of the other possible predictors are worth including, but adding FEM to the model seems reasonable. As a check, we can also use a nested LRT to compare the resulting model $logit(\pi) = \beta_0 + \beta_1 GPA + \beta_2 MCAT + \beta_3 FEM$ with the null model $logit(\pi) = \beta_0$. The drop-in-deviance is 75.791 - 50.786 = 25.005. The null model has 1 coefficient and the new full model has 4, so we refer 25.005 to a chi-square with 3 degrees

of freedom to get a p-value, which turns out to be 0.0000015. This implies the full model is a significant improvement over the null model.

Fourth predictor? With all three of GPA, $MCAT$, and FEM in the reduced model, we have five possibilities for an additional predictor:

	Additional	***-2 log(L)***			
	Predictor	**Reduced**	**Full**	**Difference**	p**-value**
GPA + MCAT + FEM +	GPA^2	50.786	49.580	1.207	0.272
	$MCAT^2$	50.786	50.781	0.005	0.939
	$GPA \cdot MCAT$	50.786	50.226	0.560	0.454
	$FEM \cdot GPA$	50.786	50.702	0.084	0.771
	$FEM \cdot MCAT$	50.786	48.849	1.937	0.164

An argument could be made that $FEM \cdot MCAT$ is worth including in the model. For example, the data suggest that female applicants may have a greater chance of acceptance than do male with the same GPA and $MCAT$ score. If you want to pursue this, you might choose to explore models with interaction terms of FEM with other predictors. On the other hand, the p-value for $FEM \cdot MCAT$ is quite far from 0.05, and for some purposes it makes sense not to include the term in the model, and instead to stop with the three-predictor model. That is what we choose to do here. Thus, we have selected our model to be

$$logit(\pi) = \beta_0 + \beta_1 GPA + \beta_2 MCAT + \beta_3 FEM$$

using nested likelihood ratio tests and a process of adding terms as given above.

7.4 Exercises

Your solutions to these problems should be done in `RStudio` using an `RMarkdown` document and choosing the **Knit HTML** or **Knit PDF** option. Create a document that clearly labels each of the problems with their exercise number. Your document should include your `R` commands as well as the plots that were created by these commands. Make sure to also carefully answer the questions by discussing the output of the plots or the code wherever necessary.

7.1) Biologists took several measurements of the heads of blue jays. Among the variables measured were $BillLength$, $BillWidth$, and $BillDepth$ (where $BillDepth$ is the distance between the upper surface of the upper bill and the lower surface of the lower bill, measured at the nostril). All measurements were in millimeters. The data are in my Public folder in the file `BlueJays.csv`. We want to study the relationship between sex (coded as M/F in the variable `KnownSex` and as 1/0 in the variable `Sex`) and measurements of the blue jays.

a) Make parallel boxplots of $BillLength$ by $KnownSex$, $BillWidth$ by $KnownSex$, and $BillDepth$ by $KnownSex$. Which of these three predictors has the weakest relationship with $KnownSex$? Which has the strongest relationship?

b) Fit a multiple logistic regression model on Sex depending on $BillLength$, $BillWidth$, and $BillDepth$.

c) Report the logit form and the probability form of the model.

7.2) Use the `BlueJays.csv` file to answer the following questions. We want to study the relationship between sex and different measurements.

a) Make parallel boxplots of $Head$ by $KnownSex$ and $Mass$ by $KnownSex$. Which of these two predictors has the strongest relationship?

b) Fit a simple logistic regression model on Sex depending on $Mass$.

c) Report the logit form and the probability form of the model.

d) Fit a multiple logistic regression model of Sex depending on $Mass$ and $Head$.

e) Report the logit form and the probability form of the multiple logistic model.

7.3) In the last chapter, we considered data on the passengers who survived and those who died when the oceanliner *Titanic* sank on its maiden voyage in 1912.

a) Fit a multiple logistic model using Age and $SexCode$ as the two predictors and $Survived$ as the response. Write down both the logit and probability forms for the fitted model.

b) According to the fitted model, estimate the probability and odds that an 18-year-old man would survive the $Titanic$ sinking.

c) Repeat the calculations for an 18-year-old woman and find the odds ratio compared to a man of the same age.

d) Redo parts (b) and (c) for a man and woman of age 50.

e) What happens to the odds ratio (female to male of the same age) when the age increases in the *Titanic* data? Will this always be the case?

7.4) Refer to Exercise 7.1 b). Follow a procedure similar to that done in the notes in Section 7.3 by adding terms related to $BillLength$, $BillWidth$, and $BillDepth$ to determine the "best" fitting model for predicting the odds of Sex using nested LRTs.

7.5) Refer to Exercise 7.3a). Follow a procedure similar to that done in the notes in Section 7.3 by adding terms related to Age and $SexCode$ to determine the "best" fitting model for predicting the odds of $Survived$ using nested LRTs.

Chapter 8

Review Problems

Your solutions to these problems should be done in `RStudio` using an `RMarkdown` document and choosing the **Knit HTML** or **Knit PDF** option. Create a document that clearly labels each of the problems with their exercise number. Your document should include your `R` commands as well as the plots that were created by these commands. Make sure to also carefully answer the questions by discussing the output of the plots or the code wherever necessary.

The problems are designed to help you review the major ideas we have covered in the course. They will also serve as your Quiz #3 grade.

8.1) What factors can help to predict how long a Major League Baseball game will last? Data was collected from http://www.baseball-reference.com for 15 games played on August 26, 2008, and stored in the file on my Public folder named `BaseballTimes.csv`. The *Time* is recorded in minutes. *Runs* and *Pitchers* are totals for both teams combined. *Margin* is the difference between the winner's and loser's scores.

a) First, analysze the distribution of the response variable (*Time* in minutes) alone. Use a graphical display as well as descriptive statistics. Describe the distribution. Also, identify the outlier (which game is it?) and suggest a possible explanation for it.

b) Examine scatterplots to investigate which of the quantitative predictor variables appears to be the best single predictor of time. Comment on what the scatterplots reveal.

c) Choose the one predictor variable that fits the data alone best. (What measure do we use to determine this?) Determine the regression equation for predicting time based on that predictor. Also, interpret the slope coefficient of this equation.

d) Analyze appropriate residual plots and comment on what they reveal about whether the conditions for inference appear to be met here.

e) Which game has the largest residual (in absolute value) for the model that you selected? Is this the same game that you identified as an outlier based on your analysis of the time variable alone?

f) Repeat the entire analysis from the parts above, with the outlier omitted.

g) Comment on the extent to which omitting the outlier changed the analysis and your conclusions.

8.2) Use the file `MetabolicRate.csv` in my Public folder to examine the linear relationship between the log (base 10) of metabolic rate and log (base 10) of body size for a sample of caterpillars.

a) Fit a least squares regression line for predicting $LogMrate$ from $LogBodySize$. What is the equation of the regression line?

b) Is the slope parameter statistically different from zero? Use simulation-based and theory-based methods.

c) Construct a 95% confidence interval for the slope parameter using simulation-based and theory-based methods.

8.3) In 1987, the federal government set the speed limit on interstate highways at 65 mph in most areas of the United States. In 1995, federal restrictions were eliminated so that states assumed control of setting speed limits on interstate highways. The datafile `Speed.csv` in my Public folder contains the variables $FatalityRate$, the number of interstate fatalities per 100 million vehicle-miles of travel, $Year$, and an indicator variable $StateControl$ that is 1 for the years 1995-2007 and zero for earlier years in the period 1987-1994.

a) Fit a regression of fatality on year. Interpret the slope coefficient.

b) Examine a residual plot. What is remarkable about this plot?

c) Fit the multiple regression of fatality rate on year, state control, and the interaction between year and state control. Use simulation-based methods to determine if there is a significant change in the relationship between fatality rate and year starting in 1995.

d) What are the fitted equations relating fatality rate to year before and after 1995?

8.4) Data that arose from a study of 51 patients treated for a form of Leukemia is in the file `Leukemia.csv` in my Public folder. The first six variables in that dataset all measure pretreatment variables: Age, $Smear$, $Infil$, $Index$, $Blasts$, and $Temp$. Fit a multiple logistic regression model using all six variables to predict $Resp$, which is 1 if a patient responded to treatment and 0 otherwise.

a) Based on values from a summary of your model, which of the six pretreatment variables appear to add to the predictive power of the model, given that other variables are in the model?

b) Specifically, interpret the relationship (if any) between Age and $Resp$ and also between $Temp$ and $Resp$ indicated in the multiple model.

c) If a predictor variable is nonsignificant in the fitted model here, might it still be possible that it should be included in a final model? Explain why or why not.

d) Use a procedure similar to that done on the last two exercises of the last chapter to determine the best fitting model. You need not use any squared variables but you should check for two-way interactions between pairs of variables.