W. D. BRINDA

DATA ANALYSIS WITH R: THE BIG PICTURE

Contents

Part I

Introductory Material

1 Overview

Before diving into the details of the material, we are going to define a few basic terms and outline the process of data analysis. At the end of the chapter, we will look at a concept map of what this book will cover. I learn best when I understand clearly how each topic fits into the big picture. To that end, this book will repeatedly refer back to the concept map that we present here.

1.1 What is Data?

"Data" is the plural form of the word "datum"; a "datum" is defined as simply "a piece of information." However, we will deal with a more specific definition in this book. We will only discuss analysis of data that can be arranged into a data frame. A *data frame*[1] is a table in which all of the values in each row are measurements of different aspects of the same object, while all of the values in each column are measurements of the same aspect of the different objects. That was quite a mouthful; it's much easier to understand by example. Let's say there are 5 students in your class, and you record the height and gender of each student. This data can be arranged into a table that might, for instance, look as follows.

[1] My use of the term "data frame" is borrowed from the R programming language, as you will learn in the next chapter.

Height (cm)	Gender
174	Female
171	Male
169	Female
174	Female
183	Male

Table 1.1: Five students' heights and genders.

Each row corresponds to a particular student in the class. The first column gives the students' heights, while the second column gives the students' genders. That means the table is a data frame.[2] Therefore, from the standpoint of this book, we are almost ready to agree that this is a dataset we can analyze. We just need to check that the columns are of the right type.

[2] Often a data frame will have an additional column on the far left, giving a name or number that identifies each row.

The rows of a data frame are called *observations*, while the columns are called *variables*. We will deal with two different types of variables in this book, and *understanding these two variable types is vital*. If the variable tells you which category each observation belongs to, then

we call it *categorical*. For instance, there are only a small number of gender categories, and the second column of Table 1.1 tells us which one describes each student. On the other hand, if the variable consists of numerical "measurements," then we call it *quantitative*. The heights reported in the first column of our table fit this description.

However, if a column consists of numbers, that doesn't necessarily mean that the variable is quantitative! Sometimes numbers are being used to identify categories, such as in zip codes. The zip code 35758 isn't "bigger" than the zip code 29475 in any important way. The numbers simply correspond to different locations. Furthermore, if the numbers really are meaningfully numeric, but there are only a few of them, you may still want to treat them as categorical. For instance, if you data frame has a variable called "year" than only takes the values 1980 and 1990, then you probably want to call it categorical. However, if it has 1980, 1981, 1982, ..., 1990, then you probably want to call it quantitative.

Conversely, if a column consists of words, that doesn't necessarily mean that you have to treat the variable as categorical! For example, let's say the survey question "What is your view of the candidate?" allows three possible responses: Unfavorable, Neutral, or Favorable. Data on this variable technically fits the definition of categorical, but it also has a very natural ordering to it[3] that isn't captured by categorical data analysis techniques. In practice, one may want to treat this variable as quantitative by, for instance, rewriting Unfavorable as -1, Neutral as 0, and Favorable as 1. Variables of this in-between type are often called *ordinal*, and you can choose to treat them as either categorical or quantitative.

[3] Unfavorable < Neutral < Favorable.

Although any pieces of information could be called "data," in this book we will learn how to analyze data with a specific structure:

1. it is organized into a data frame
2. each variable of the data frame can be considered either categorical or quantitative

This may seem limiting, but it actually covers a lot of ground, as you will see from the wide range of datasets analyzed in this book. Even advanced machine learning tasks that seem to lie outside of this paradigm are often converted to problems that fit this pattern. A mastery of the techniques and concepts in this simplified case is essential to being an effective data analyst.

1.2 What is Data Anaysis?

Once you've got a data frame in which you have identified each variable as either categorical or quantitative, then you're ready to "analyze" it. But, like "data," "data analysis" is another nebulous term that we should narrow down a bit to simplify things for ourselves. We will break the process of data analysis up into two distinct stages: Description and Inference.

1.2.1 *The Description Stage*

In the *description* stage, you simply want to understand the objects represented in the dataset. For instance, in Table 1.1 the "objects" represented are the five students in the class. To help us understand the objects, we use two tools: plots and statistics.

- A *plot* is any picture that represents aspects of the data. A good plot might provide a clear and intuitive understanding of the data, or it might reveal an unexpected fact that you wouldn't have noticed by just looking at the data frame.
- A *statistic* is any value that is calculated from a dataset. At this stage, our interest is in statistics that will summarize aspects of the data for us in useful ways.[4]

[4] Statistics also play a central role in inference, as you will learn in Chapter 3.

Looking at plots and thinking about statistics will likely improve your understanding of the objects in the dataset, and, in doing so, hopefully it will help you address the questions that motivated the data analysis (if there were any).

But which plots should you create, and which statistics should you calculate? That depends on two things:

- First, the set of possible plots and statistics depends on the nature of the dataset. In particular, the types of variables (categorical and quantitative) determine what plots and statistics make sense, as you will see in the coming chapters.[5]
- Secondly, it depends on the purpose of the data analysis. You don't need to make every possible plot and calculate every possible statistic. Instead, think about questions are you trying to answer?[6] Let these questions guide your choices; make the plots and calculate the statistics that you think will be helpful. In general, however, it's also a good idea to make a variety of plots for yourself along the way, because you never know what you're going to learn.[7]

[5] The size of the dataset also matters in some cases. If the number of observations or the number of variables is very large, some techniques may be computationally infeasible. However, the topic of "big data" is beyond the scope of this book.

[6] Sometimes there is no specific question in mind, and the data analysis is simply *exploratory*.

[7] If you're writing up a report summarizing your results, you probably don't want to include every plot, as many of them will be uninteresting or redundant. You shouldn't overwhelm your reader; instead, just get to the point.

This book will introduce a handful of plots and statistics that are often useful in common cases. But you never know what peculiar datasets and questions you'll face in "real world" data analysis. Ultimately, a data analyst must be creative.

Let's look at one simple example of how the description stage of data analysis might help you address a real-world problem.

Example 1.2.1. My computer's hard drive is running low on space. I have 10,000 files, each of which is a `DOC`, a `JPG`, or an `MP3`. I have saved the filetype and size (in megabytes) of each file as a `CSV` (comma-separated values) file on my website. Below, the data is read into R as a data frame, and the first six rows are displayed.[8]

[8] Throughout this book, any R code I use will be on display. You may want to ignore the code for now if it is distracting you from the main concepts of this chapter.

```
# Read in the data from the web
x <- read.csv("http://www.stat.yale.edu/~wdb22/Files.csv")
```

```
# Display the first six rows of the data frame
head(x)

##   type size
## 1  JPG 0.38
## 2  DOC 0.04
## 3  MP3 0.31
## 4  JPG 0.16
## 5  JPG 0.55
## 6  DOC 0.29
```

Let's take a look at how many files I have of each type.

```
# Number of files of each type
table(x$type)

##
##  DOC  JPG  MP3
## 4000 4990 1010
```

My computer has many times more `JPG`s and `DOC`s than it has `MP3`s. But that's not really getting at the question of space. Instead, let's add up the file sizes of all our `DOC`s, `JPG`s, and `MP3`s separately.[9]

[9] In practice, I would also create side-by-side boxplots (see Chapter 6) for this data, but we'll keep things simple at this point.

```
# Find the total amount of space used for each type
totals <- aggregate(x$size, list(type=x$type), sum)
names(totals)[2] <- "total"
totals

##   type    total
## 1  DOC   809.88
## 2  JPG  1240.29
## 3  MP3 10443.54
```

Aha! Now we see that the vast majority of the space on my hard drive is being used by `MP3`s. Let's turn these numbers into a pie chart (Figure 1.1) to give ourselves a visual understanding of how much more space is being used by `MP3`s than by other files.

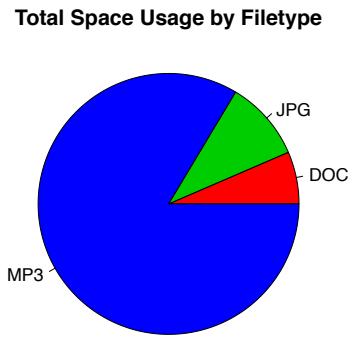


Figure 1.1: The pie chart shows the relative proportion of space being used up by each filetype. We see that `MP3`s are taking up the vast majority.

```
pie(totals$total, totals$type, col=2:4,
    main="Total Space Usage by Filetype")
```

We've learned that my storage space problem is basically due to `MP3`s. This data analysis didn't *solve* my problem, but it tells me where I should direct my attention. There is little point in, for instance, trying to figure out how to compress my `JPG` files, but there might be great gain if I could compress my `MP3` files.

In this example, we were trying to better understand only the set of files that were represented in the dataset. Therefore, once we had finished the description stage, we were finished with our data analysis. Often, however, we are also interested in pursuing the more

ambitious aim of drawing conclusions about objects that were not represented in the dataset, as we will discuss next.

1.2.2 *The Inference Stage*

The *inference* stage only occurs if you also want to draw conclusions about a larger population that your data was sampled from. In Example 1.2.1, the dataset represented all the files on my computer. Because my question was only about those files, I didn't need to speculate about any files that weren't in the dataset. Often, however, we want to generalize our conclusions from a dataset to a larger population. A common example is opinion polls. How is it that pollsters can feel confident in their claims about an entire country after asking their questions to only a few hundred people?

Let's revisit Example 1.2.1, but this time pretend that we didn't have all the files in our dataset. Imagine that acquiring filetypes and sizes is hard work. Instead of gathering that data on all the files, we will only gather the data on a sample of the files. In particular, we will take a *random sample*, a subset of the population in which each file was equally likely to be included. We will use a *sample size* of 100. This random sample can be easily simulated in R.

```
# Take a random sample of size 100 from the full data set
set.seed(1)
y <- x[sample(1:nrow(x), 100), ]
head(y)

##      type size
## 2656  DOC 0.37
## 3721  JPG 0.25
## 5728  JPG 0.10
## 9080  JPG 0.12
## 2017  JPG 0.26
## 8980  JPG 0.16
```

Next, we will look at the total amount of space used by the different filetypes in our sample.

```
# Find the total amount of space used for each type
totals <- aggregate(y$size, list(type=y$type), sum)
names(totals)[2] <- "total"
totals

##   type  total
## 1  DOC   5.91
## 2  JPG  13.71
## 3  MP3 198.77
```

As with the full data set, the sample has MP3s taking up the vast majority of the space. The give us a visual impression, we make another pie chart. As seen in Figure 1.2, it looks strikingly similar to the pie chart for the full data set (Figure 1.1).

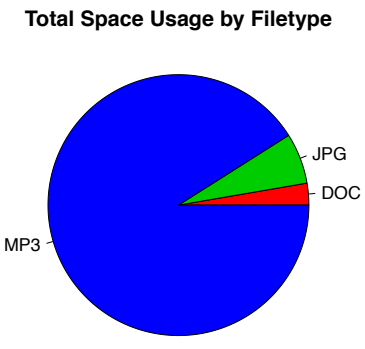


Figure 1.2: The pie chart shows the relative proportion of space being used up by each filetype in our random sample taken from the full dataset.

```
pie(totals$total, totals$type, col=2:4,
    main="Total Space Usage by Filetype")
```

Our analysis of this random sample of 100 files leads us to the same basic conclusion that our analysis of the full 10,000 files did.

This example demonstrates a general phenomenon and the key insight underlying inference: *a random sample tends to resemble the population that it was drawn from*. Furthermore, *the larger the sample size, the stronger the resemblance tends to be*. In the inference stage of data analysis, we make probabilistic statements about the population based on statistics calculated from the sample. Understanding this process in detail requires some knowledge of probability; our main explanation of probability and inference will come in Chapter 3.

1.3 The Mix of Variable Types

Let's recap the data analysis process. Given a data frame,[10] we classify each variable (column) as either categorical or quantitative, if possible.[11] The first stage of analyzing this data is to simply describe the observations (rows) in the data frame with the assistance of plots and statistics. Plots are graphical displays that help you see aspects of the data, while statistics are values that summarize aspects of the data. If you also want to draw conclusions about a larger population than was represented in the data frame, then you proceed to the inference stage. In this stage, you make probabilistic statements about the population based on statistics calculated from the sample. Inference is justified by the fact that a random sample tends to resemble the population that it was drawn from.

In the description stage, the plots and statistics available depend on the types of variables in your data frame. In Part II, this book will present some of the plots and statistics that I have found most useful for analyzing each of the cases listed below. Inference also depends on the types of variables, and in Part III, we go through each case again, this time presenting some of the inference techniques that I have found most useful. The cases are broken up as follows.

1. Categorical variables (Description: Chapter 4; Inference: Chapter 7)
   - one categorical variable
   - two categorical variables
2. Quantitative variables (Description: Chapter 5; Inference: Chapter 8)
   - one quantitative variable
   - two quantitative variables[12]
   - three quantitative variables
3. Both categorical and quantitative variables together (Description: Chapter 6; Inference: Chapter 9)

[10] Recall that if the data is not given to you as a data frame, you may still be able to rewrite it in that form.

[11] If some of your variables aren't easily classified as categorical or quantitative (e.g. an essay), then you can leave those variables out, and still perform a data analysis on the remaining variables, if there are any.

[12] This section will also include an extended section on linear regression, a common and powerful technique that can be used on an arbitrary number of variables with arbitrary types!

- one categorical variable and one quantitative variable
- two categorical variables and one quantitative variable
- one categorical variable and two quantitative variables

Why did I stop with these cases? Because these are the cases in which it is possible to make easily interpretable plots. If you try to include any more variables than this in a single plot, you're usually pushing it. I don't want to discourage you from being creative, but you should keep in mind that there are limits to humans' pattern-detection abilities. You may come up with a clever way to pack four quantitative and three categorical variables onto a single plot, but that doesn't mean that the people who see your plot are going to be able to make any sense of it.

You often just want to work with some subset of your variables at a time. For instance, assume you want to analyze a data frame comprising two categorical variables called X and Y and one quantitative variable called Z. We see that there is a section of this book ("Two categorical variables and one quantitative variable") telling us how to analyze all three of these variables together. But we could also look at any subset of the variables. Here are the seven possibilities:

- X, Y, and Z all together (two categorical variables and one quantitative variable)
- X and Y together (two categorical variables)
- X and Z together (one categorical variable and one quantitative variable)
- Y and Z together (one categorical variable and one quantitative variable)
- X by itself (one categorical variable)
- Y by itself (one categorical variable)
- Z by itself (one quantitative variable)

Each of these subsets is addressed by a section of the book and could be analyzed accordingly. Should you look into them all? Don't hesitate to make any plot that might be interesting,[13] but there are often so many possibilities that you don't want to pursue them all. Focus on the ones that make the most sense based on the purpose of your data analysis.

In a sense, this book gives you step-by-step instructions for analyzing a dataset.

1. Check that the data is structured as a data frame; if it isn't, try to rewrite it as one.
2. Classify each variable as either categorical or quantitative; ignore any variables that are neither.
3. Decide which subsets of the variables you want to analyze; each subset must fit a pattern from the above list.

[13] Often a plot will reveal that some of the data values don't make any sense such as, for example, if the variable is supposed to be measuring a distance and you find that you have negative values. This is one reason to make more plots than you strictly need; they make it more likely that you will detect nonsensical or suspicious data. When you find strange things in a dataset, you might try asking the people who collected the data if they can explain it.

4. Description stage: For each desired subset, find the section in Part II discussing that subset of variables. Adapt the sections' example code to create plots and calculate statistics for your data; think carefully about the plots and statistics to better understand your data.

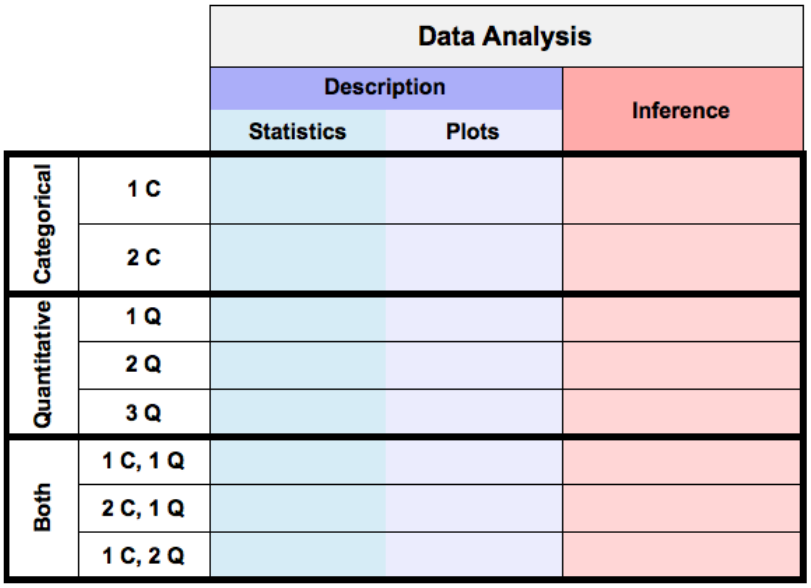
5. Inference stage: If you also want to draw conclusions about a larger population, then, for each desired subset, find the section in Part III discussing that subset of variables. Adapt the sections' examples to make probabilistic statements about the population your observations were sampled from.

However, data analysis is rarely so straight-forward in the real world. This book is intended to make you comfortable with the principles behind data analysis and with R programming. It is a starting point, but keep studying because there are a lot of useful techniques out there. And be ready to think creatively, because you may face data analysis tasks that nobody else has had to think about before!

1.4 *The Big Picture*

The main concepts and topics of this book are organized into a graphic below. We will fill in more details as we go. I find that when learning a new topic, it is always helpful to clearly understand how it's related to the other topics, that is, how it fits into the *big picture*. Each chapter in Parts II and III will start by reminding you of the big picture and pinpointing which piece of it they will be covering. Those chapters end with another view of the picture, this time filled in with the main points you learned in that chapter.

Figure 1.3: The "big picture" of topics this book will cover.



Next up, Chapter 2 introduces you to R programming. First, it will help you get set up to run R on your computer, so that you

can replicate any of the code snippets you find in this book's data analysis examples. Then it covers the very basics of the programming language, so that you can begin to make sense of the book's code snippets. However, most of your understanding of R will happen gradually as you go through the book's examples.

After that, Chapter 3 discusses the relationship between probability and inference. There you will see simple examples to help you get a clear understanding of how inference works. The chapter contains a bit of mathematics which I hope will not bog you down. If you'd prefer, you can skip that chapter for now and go through Part II covering the description stage of data analysis. But you should return to Chapter 3 before moving on to Part III covering inference.

Finally, Part IV wraps up with a couple additional thoughts. First, Chapter 10 describes briefly my process for writing up the results of a data analysis as a report. Then, Chapter 11 offers practical tips and intuition that I call "statistical reasoning." Here I give my opinions on how to give yourself the best chance of getting your data analyses right, as well as how to make sense of endless stream of data that bombards us in the modern world.

2 *R Basics*

R Basics

3 Probability and Inference Basics

The magic of statistics happens in the inference stage of data analysis. From a sample of data points, we will be able to make probabilistic statements about a whole population. But you have to understand some concepts from probability theory before you can understand inference.

3.1 Probability

When you flip a coin, you don't know which side it is going to land on. The unpredictable nature of phenomena such as this is often called *randomness*, and the behavior of unpredictable phenomena can be modeled by the mathematical concept of *probability*. But probability is also a concept that you're intuitively familiar with from your everyday experience and language. In this section, we'll build on that intuition by introducing the concept of a *random variable*.

A *random variable* is an "unknown" quantity that has a probability distribution. The *probability distribution* (or just *distribution*) tells you the probability that the unknown quantity is in any inteval. To make this more concrete, let's think about a specific example in the context of coin-flipping.

3.1.1 Binomial Distributions

Assume I am going to flip a fair coin[1] once. Let the random variable X represent the number of times that the coin lands showing heads. Then $P(X = 0) = 1/2$ and $P(X = 1) = 1/2$.

[1] A *fair* coin has an equal probability of landing on either heads or tails.

Next, assume I am going to flip a fair coin twice. Again, let the random variable X represent the number of times that the coin lands showing heads. There are four possibile outcomes, each with probability $1/4$ of occurring: HH, HT, TH, TT. One of the outcomes results in $X = 0$, two outcomes result in $X = 1$, and one outcome results in $X = 2$. This means that $P(X = 0) = 1/4$, $P(X = 1) = 1/4 + 1/4 = 1/2$, and $P(X = 2) = 1/4$.

What if the coin isn't necessarily fair? Assume, more generally, that we have a coin that has probability p of landing heads on each flip. Let the random variable X represent the number of times that the coin lands showing heads. Then $P(X = 0) = 1 - p$ and $P(X = 1) = p$. This is called the Bernoulli distribution with parameter p.

Again, let's think about flipping the coin twice. Let the random variable X represent the number of times that the coin lands showing heads. There are four possibile outcomes: HH, HT, TH, TT. This time, however, we can't assume that the outcomes aren't equally probable. To find the probability of each outcome, we can draw a tree.

Along each branch, write the probability of moving from the left endpoint to the right endpoint. In coin-flipping, we assume that each flip is *independent* of all the others. That means that your probability of getting heads doesn't depend on what has happened on past flips. We can also say that the flips are *identically distributed* because they each have the exact same probability of showing heads. These two conditions often show up together in probability and statistics, so we have an abbreviation for them: *iid* stands for "independent and identically distributed."

To find the probability of following any particular path, multiply the numbers along that path.

- $P(X = 0) = P(TT) = (1-p)^2$
- $P(X = 1) = P(HT) + P(TH) = p(1-p) + (1-p)p = 2p(1-p)$
- $P(X = 2) = P(HH) = p^2$

Let's generalize this scenario to an arbitrary number of coin-flips. First, notice that the probability of any path only depends on the number of heads and tails it has. If n is the total number of coin-flips, then the probability of any particular outcome with k heads (and $n-k$ tails) is exactly $p^k(1-p)^{n-k}$. But to find the probability that $X = k$, we need to take the sum of the probabilities of all the paths that have k heads. Because all these paths have the same probability, we simply need to multiply that probability by the number of paths with k heads. The notation $\binom{n}{k}$ is used to denote the number of coin-flip sequences with k heads in n flips.[2] Therefore, if k is an integer between 0 and n, then[3]

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

It can be shown that the number of coin-flip sequences with k heads in n flips is equal to a simple expression in terms of factorials.

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Thus, given any n, p, and k, you could use this formula (and a calculator) to find the probability of flipping k heads. Thankfully, R makes it even easier than that with the built-in `dbinom` function. For instance, to get the probability of one heads in two flips of a fair coin ($p = .5$), which we calculated eariler, we can use the `dbinom` function.

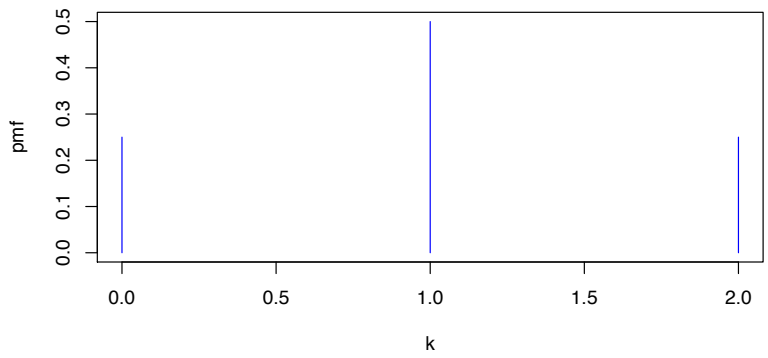
[2] Equivalently, this is the number of subsets of size k that exist within a set of size n. When you see the symbol, you read it aloud as "n choose k."

[3] For any other numbers, the probability is zero.

```
dbinom(1, 2, .5)

## [1] 0.5
```

This is a value from the *probability mass function* (pmf), which gives the probability that $X = k$ for each k. Let's see what the whole pmf looks like.

```
drawPMF <- function(n, p=.5) {
  # This function draws a pmf for the Binomial
  # distribution with the given n and p.
  k <- 0:n
  pmf <- sapply(k, dbinom, size=n, prob=p)
  plot(k, pmf, col=4, type="h", ylim=c(0, max(pmf)))
}

drawPMF(2)
```



The height of each bar above each value tells you the probability that X takes that value. To find the probability that X is in some set A, you simply need to sum up the heights of the bars above the numbers in A. The total sum of the heights has to be one.

As another example, let's increase the sample size to 20 and draw the pmf again.

```
drawPMF(20)
```

For any n (total number of coin flips) and p (probability that each flip will land heads), we get a distribution defining the probability that X (the number of heads) takes each value from 0 to n. We call these *Binomial* distributions; any (n, p) pair defines the Binomial(n, p) distribution.

In addition to the pmf, another useful function for probability distributions is the *cumulative distribution function* (cdf). For any probability distribution, the cdf F is defined by

$$F(t) := P(X \leq t).$$

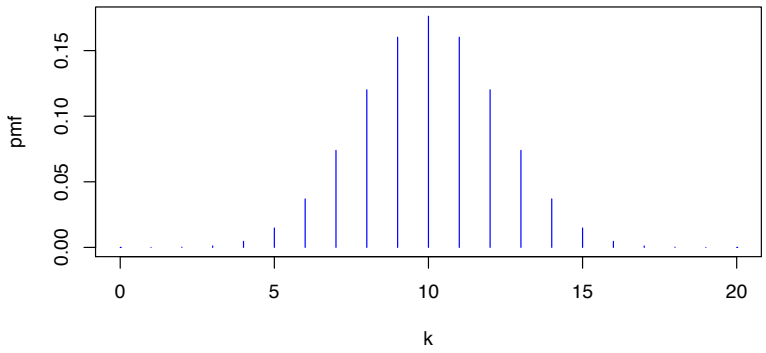


Figure 3.1: The probability mass function of the Binomial(20, .5) distribution.

Visualizing this in terms of the pmf, $F(t)$ is equal to the sum of the heights of the bars from $-\infty$ to t.

Let's see a specific example of a cdf. Recall the scenario of two fair coin tosses from earlier. We found that $P(X = 0) = 1/4$, $P(X = 1) = 1/4 + 1/4 = 1/2$, and $P(X = 2) = 1/4$. The cdf in this case is thus

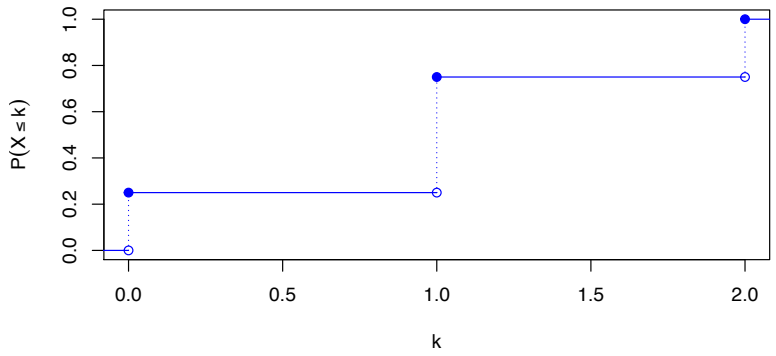
$$F(t) = P(X \leq t) = \begin{cases} 0 & t < 0 \\ 1/4 & 0 \leq t < 1 \\ 3/4 & 1 \leq t < 2 \\ 1 & \text{otherwise} \end{cases}$$

See Figure 3.2 for a drawing of the cdf.

```
drawCDF <- function(n, p=.5) {
  # This function draws a cdf for the Binomial
  # distribution with the given n and p.
  k <- 0:n
  cdf <- sapply(k, pbinom, size=n, prob=p)
  lower <- c(0, cdf[1:n])
  plot(k, lower, col=4, ylim=c(0, max(cdf)),
       ylab=expression(P(X <= k)))
  points(k, cdf, col=4, pch=19)
  for(i in 0:(n-1)) {
    lines(c(i, i), c(lower[i+1], cdf[i+1]), lty=3, col=4)
    lines(c(i, i+1), c(cdf[i+1], lower[i+2]), col=4)
  }
  lines(c(n, n), c(lower[n+1], cdf[n+1]), lty=3, col=4)
  lines(c(-1, 0), c(0, 0), col=4)
  lines(c(n, n+1), c(1, 1), col=4)
}

drawCDF(2)
```

For Binomial distributions, you can find $F(t)$ by adding up the

Figure 3.2: The cdf of the Binomial(2, .5) distribution.



probabilities for all the integers from zero to t.

$$F(t) = P(X \leq t) = \begin{cases} \sum_{i=0}^{\lfloor t \rfloor} \binom{n}{i} p^i (1-p)^{n-i} & t \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Again, R saves us a lot of work with a built-in function `pbinom`. For instance, to find the probability of one or fewer heads in two flips of a fair coin ($p = .5$), use the `pbinom` function as follows.

```
pbinom(1, 2, .5)

## [1] 0.75
```

The cdf evaluated at t tells you the probability that X is less than or equal to t, but it can also be used to find the probability that X is greater than t. This is easy to understand by picturing the pmf. $P(X > t)$ is equal to the sum of the heights of the bars to the right of t. Because the total sum is one, this probability is just 1 minus the sum of the heights of the bars from $-\infty$ to t. That is,

$$\begin{aligned} P(X > t) &= 1 - P(X \leq t) \\ &= 1 - F(t) \end{aligned}$$

The cdf can also be used to find the probability that X is in any given interval. Assume you want to find $P(a < X \leq b)$. That is equal to the sum of the heights of the pmf bars that are between a and b (including the endpoint b but not a). We can start with $F(b)$, the sum of the bars up to b then simply subtract $F(a)$, the sum of the bars up to a to get the sum of the bars just in our desired interval $(a, b]$.

$$\begin{aligned} P(a < X \leq b) &= P(X \leq b) - P(X \leq a) \\ &= F(b) - F(a) \end{aligned}$$

Let's work through some specific example questions for a Binomial random variable. Suppose you have a coin with heads probability of .4, and you plan to flip it 100 times. What is the probability that you get exactly 40 heads? At most 35 heads (i.e. $P(X \leq 35)$)?

At least 45 heads (i.e. $P(X > 44)$)? Between 30 and 50 heads (i.e. $P(29 < X \leq 50)$)?

```
dbinom(40, 100, .4)

## [1] 0.08121914

pbinom(35, 100, .4)

## [1] 0.1794694

1 - pbinom(44, 100, .4)

## [1] 0.1789016

pbinom(50, 100, .4) - pbinom(29, 100, .4)

## [1] 0.968463
```

If X is a Binomial(n, p) random variable, then we can write it as

$$X = X_1 + X_2 + \ldots + X_n$$

where the X_i are independent Bernoulli(p) random variables.[4] Each X_i represents the outcome of the ith coin-flip. Because of this representation, we can say that X is a sum of iid random variables, which is a common scenario in inference.

[4] Recall that Bernoulli(p) means $P(X_i = 0) = 1 - p$ and $P(X_i = 1) = p$.

Each Binomial distribution is a *discrete distribution*, meaning that the set of possible values for X is *countable*.[5] But it is also possible to talk about distributions over uncountable sets, such as the entire real line. For data analysis, the most important example of such a distribution is the standard *Normal* distribution.[6]

[5] "Countable" means that it's possible to make a list of them.

[6] Another common term for Normal is "Gaussian."

3.1.2 Normal Distributions

Whereas a discrete distribution has a probability mass function (pmf), a *continuous distribution* such as the standard Normal has a *probability density function* (pdf, or just *density*). And whereas a pmf tells you the probability that $X = k$, a pdf is used to find the probability that X is in a given interval. In particular, if X has a pdf, then $P(X \in [a, b])$ is equal to the area under the pdf in the interval from a to b. The total area under the pdf is equal to one.

The standard Normal is the distribution defined by the pmf

$$f(t) = \frac{1}{\sqrt{2\pi}} e^{-t^2/2}$$

Figure 3.3 shows this function over the interval from -3 to 3, although it's positive over the entire real line.

```
# Draw the standard normal density curve
grid <- seq(-3, 3, length.out=100)
plot(grid, sapply(grid, dnorm), type="l", col=4,
     xlab="x", ylab="density", main="Standard Normal Density")
```

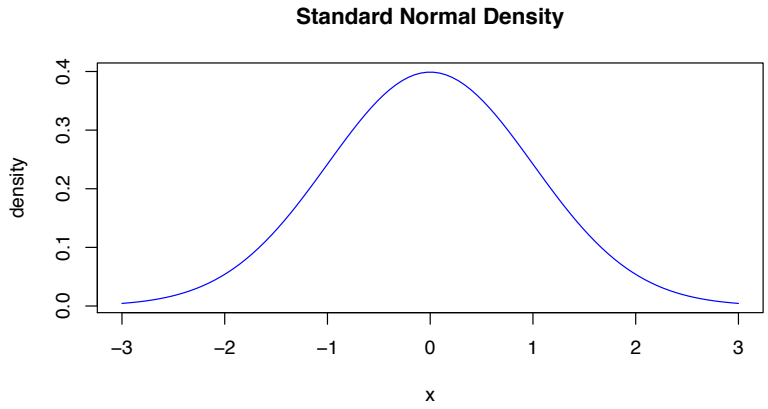


Figure 3.3: The pdf that defines the standard Normal distribution.

Let's see a simple example of what the pdf tells you. If X is standard Normal, then the probability that X is between 0 and 2 is equal to the area under the pdf shaded in Figure 3.4.

```
# Draw the standard normal density curve
grid <- seq(-3, 3, length.out=100)
plot(grid, sapply(grid, dnorm), type="l", col=4,
     xlab="x", ylab="density", main="Standard Normal Density")
# Shade the area under the curve from 0 to 2
grid <- seq(0, 2, length.out=100)
points(grid, sapply(grid, dnorm), type="h", col=3)
```

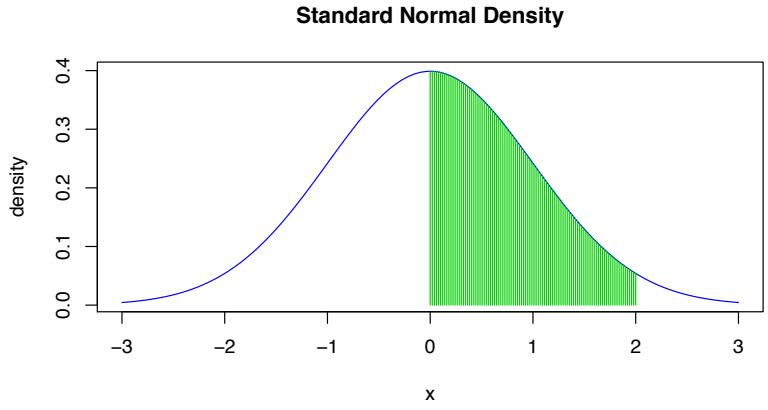


Figure 3.4: The standard Normal pdf with the region from 0 to 2 shaded.

In general, the probability that a random variable's value is between a and b is the area under its pdf from a to b. If you're familiar with calculus, you will realize that this is equal to the definite integral. In the standard Normal case, for instance, this is

$$P(X \in [a, b]) = \int_a^b \frac{1}{\sqrt{2\pi}} e^{-t^2/2} \, \mathrm{d}t.$$

Thankfully, we can use the cdf and built-in R functions to avoid having to calculate this integral by hand. The cdf $F(t)$ tells you the

amount of area under the pdf from $-\infty$ to t. The same equation we derived for discrete distributions also holds for continuous distributions.[7]

[7] In the case of continuous distributions, it doesn't actually matter whether you're trying to find the probability of the open or the closed interval. The difference is a single point, and there's no area under a single point. So $P(X \in [a,b]) = P(X \in (a,b))$.

$$P(X \in (a,b]) = F(b) - F(a)$$

Thus, we can repeat the same tricks that we used in the discrete case above to find probabilities of intervals. To find the area under the pdf from 0 to 2, for example, we can take the amount of area up to 2 and subtract the amount of area up to 0. In R, the Normal cdf is `pnorm`, so for a standard Normal X, $P(X \in [0,2])$ equals

```
pnorm(2) - pnorm(0)

## [1] 0.4772499
```

The bell curve shape actually defines a two-dimensional family of Normal distribution, parameterized by a mean and a variance.[8] The standard Normal distribution is simply the Normal distribution with mean equal to zero and variance equal to one.

[8] The *variance* quantifies how spread out a distribution is. It is defined in the next section.

3.1.3 Expected Values

The *expected value* (also known as the *expectation* or the *mean*) of a random variable is the weighted average of its possible values, weighted by the probabilities of those values. If X is a discrete random variable that can take any value in some set K, this can be expressed as[9]

[9] For the expected value of a continuous random variable, the sum is replaced with an integral, and the pmf is replaced with a pdf.

$$E(X) := \sum_{k \in K} kP(X = k)$$

As an example, let's find the expected value of X if it has a Bernoulli(p) distribution. Recall $P(X = 0) = 1 - p$ and $P(X = 1) = p$. Then

$$\begin{aligned} E(X) &:= \sum_{k \in \{0,1\}} kP(X = k) \\ &= (0)P(X = 0) + (1)P(X = 1) \\ &= (0)(1 - p) + (1)(p) \\ &= p \end{aligned}$$

If X and Y are random variables, and a and b are constants, then[10]

[10] This fact holds as long as the two expectations aren't ∞ and $-\infty$.

$$E(aX + bY) := aE(X) + bE(Y)$$

We can use this fact to find the expectated value of a Binomial(n, p) random variable X. Recall that X can be expressed as a sum of n independent Bernoulli(p) random variables.

$$\begin{aligned} E(X) &= E(X_1 + X_2 + \ldots + X_n) \\ &= E(X_1) + E(X_2) + \ldots + E(X_n) \\ &= p + p + \ldots + p \\ &= np \end{aligned}$$

Any function of X, such as X^2 or e^X is also a random variable. It's expectation can be written in terms of the probability distribution of X.

$$E[f(X)] = \sum_{k \in K} f(k)P(X = k)$$

The *variance* of X is the expectation of the squared deviation of X from its expected value. For simplicity, we'll use the symbol μ to denote $E(X)$.

$$\begin{aligned} \text{Var}(X) &:= E[(X - \mu)^2] \\ &= \sum_{k \in K} (k - \mu)^2 P(X = k) \end{aligned}$$

The *standard deviation* is defined to be the square root of the variance.

Let's find the variance of a Bernoulli(p) random variable X. We just found that its expected value is p. So

$$\begin{aligned} \text{Var}(X) &:= E[(X - \mu)^2] \\ &= \sum_{k \in \{0,1\}} (k - p)^2 P(X = k) \\ &= (0 - p)^2(1 - p) + (1 - p)^2(p) \\ &= p^2(1 - p) + p(1 - p)^2 \\ &= p(1 - p)[p + (1 - p)] \\ &= p(1 - p) \end{aligned}$$

It is easy to show from the definition of variance that for any constant a and random variable X, $\text{Var}(aX) = a^2\text{Var}(X)$ and $\text{Var}(X + a) = \text{Var}(X)$. Another important fact about variances is that if X and Y are *independent* random variables, then

$$\text{Var}(X + Y) := \text{Var}(X) + \text{Var}(Y)$$

We can apply this fact to find the variance of a Binomial(n, p) random variable X, again by expressing it as a sum of n independent Bernoulli(p) random variables.

$$\begin{aligned} \text{Var}(X) &: = \text{Var}(X_1 + X_2 + \ldots + X_n) \\ &= \text{Var}(X_1) + \text{Var}(X_2) + \ldots + \text{Var}(X_n) \\ &= p(1 - p) + p(1 - p) + \ldots + p(1 - p) \\ &= np(1 - p) \end{aligned}$$

3.1.4 *Asymptotics*

In Chapter 1, we pointed out a key insight that makes inference possible: a random sample tends to resemble the population it was drawn from, and the larger the sample, the stronger the resemblance tends to be. Here we clarify how that fact works for means.

Assume $X_1, \ldots, X_n$ are iid random variables, each with an expected value[11] μ and a finite variance σ^2. Often a statistic of interest is the *sample mean* $\bar{X} := \frac{1}{n}(X_1 + \ldots + X_n)$.

[11] Because each X_i has the same distribution, they must all have the same expected value and variance.

In such cases, the expected value of $\bar{X}$ is exactly equal to the expected value of the random variables it's averaging.

$$\begin{aligned} E(\bar{X}) &= E(\frac{1}{n}(X_1 + \ldots + X_n)) \\ &= \frac{1}{n}[E(X_1) + \ldots + E(X_n)] \\ &= \frac{1}{n}[\mu + \ldots + \mu] \\ &= \frac{1}{n}[n\mu] \\ &= \mu \end{aligned}$$

Furthermore, the variance of $\bar{X}$ is

$$\begin{aligned} \text{Var}(\bar{X}) &= \text{Var}(\frac{1}{n}(X_1 + \ldots + X_n)) \\ &= \left(\frac{1}{n}\right)^2 [\text{Var}(X_1) + \ldots + \text{Var}(X_n)] \\ &= \frac{1}{n^2}[\sigma^2 + \ldots + \sigma^2] \\ &= \frac{1}{n^2}[n\sigma^2] \\ &= \frac{\sigma^2}{n}. \end{aligned}$$

The variance gets smaller as the number of random variables being averaged increases. In other words, the distribution of the sample mean $\bar{X}$ becomes increasingly *concentrated* around μ as the sample size increases. A more formal statement of this is called the *Law of Large Numbers* (LLN).

The LLN doesn't tell you anything about the shape of $\bar{X}$'s distribution. However, the *Central Limit Theorem* (CLT) says that the distribution of $\bar{X}$ increasingly resembles a Normal distribution as the sample size gets larger. In particular, it resembles the Normal distribution with mean μ and variance σ^2/n (i.e. the mean and variance that we just derived for $\bar{X}$). It follows that a particular *linear transformation* of $\bar{X}$ has approximately a *standard* Normal distribution: $\sqrt{n}\frac{\bar{X}-\mu}{\sigma}$. It is easy to show that this random variable has mean zero and variance one.

$$\begin{aligned} E(\sqrt{n}\frac{\bar{X}-\mu}{\sigma}) &= \frac{\sqrt{n}}{\sigma}[E(\bar{X}) - E(\mu)] \\ &= \frac{\sqrt{n}}{\sigma}[E(\bar{X}) - \mu] \\ &= \frac{\sqrt{n}}{\sigma}[\mu - \mu] \\ &= 0 \end{aligned}$$

$$\begin{aligned}\text{Var}(\sqrt{n}\frac{\bar{X}-\mu}{\sigma}) &= \frac{n}{\sigma^2}\text{Var}(\bar{X}-\mu)\\ &= \frac{n}{\sigma^2}\text{Var}(\bar{X})\\ &= \frac{n}{\sigma^2}\left(\frac{\sigma^2}{n}\right)\\ &= 1\end{aligned}$$

You may have noticed in Figure 3.1 that the Binomial(20, .5) pmf resembles a bell curve. Figure 3.5 shows that pmf again, this time superimposed in front of the Normal pdf with matching mean ($\mu = np = 10$) and variance ($\sigma^2 = np(1-p) = 5$).

```
n <- 20; p <- .5
grid <- seq(0, n, length.out=100)
plot(grid, sapply(grid, dnorm, mean=n*p, sd=sqrt(n*p*(1-p))),
     type="l", col=4)
k <- 0:n
pmf <- sapply(k, dbinom, size=n, prob=p)
points(k, pmf, col=3, type="h", ylim=c(0, max(pmf)))
```

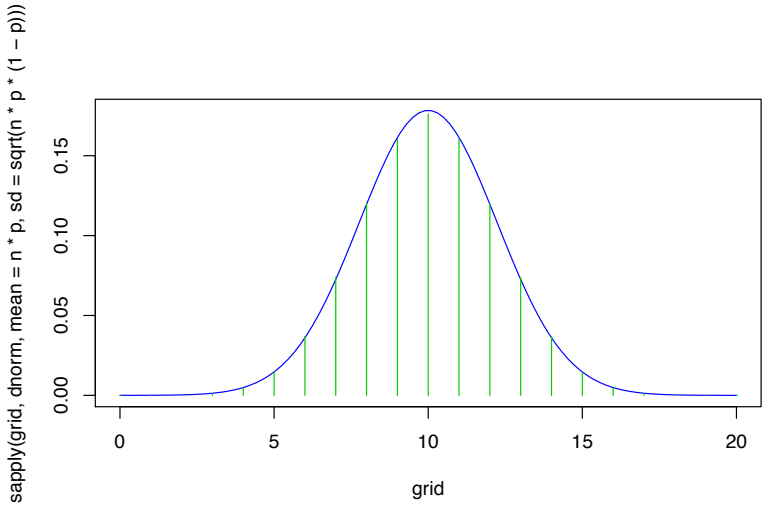


Figure 3.5: The Normal(10, 5) pdf in blue with the Binomial(20, .5) pmf superimposed in green.

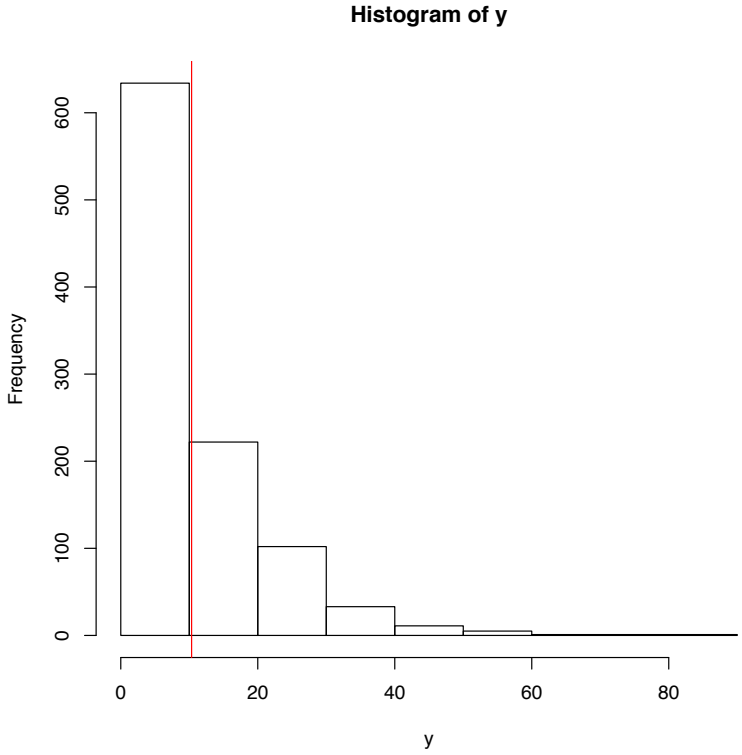
This is a perfect example of the CLT at work. Why? Because a Binomial(20, .5) random variable is simply a sum of 20 iid Bernoulli(p) random variables. And the sum is proportional to the sample mean, so the distributions of the sum and sample mean have the same shape, which resembles a bell curve in this case.[12]

As stated above, the CLT doesn't tell you how large you need your sample size to be before the sample mean is well-approximated by a Normal distribution. That depends on the distribution of the X_i random variables that go into it, but many data analysts consider Normal approximations reliable for sample sizes of at least 30.

Let's see the CLT in action one more time using a simple simulation. Consider the set of `MP3`s from the familiar computer files dataset. There are 1010 `MP3` files, and their mean is about 10.3 MB.

[12] The resemblance is close that it is common to approximate Binomial distributions using a Normal distribution even at moderate sample sizes.

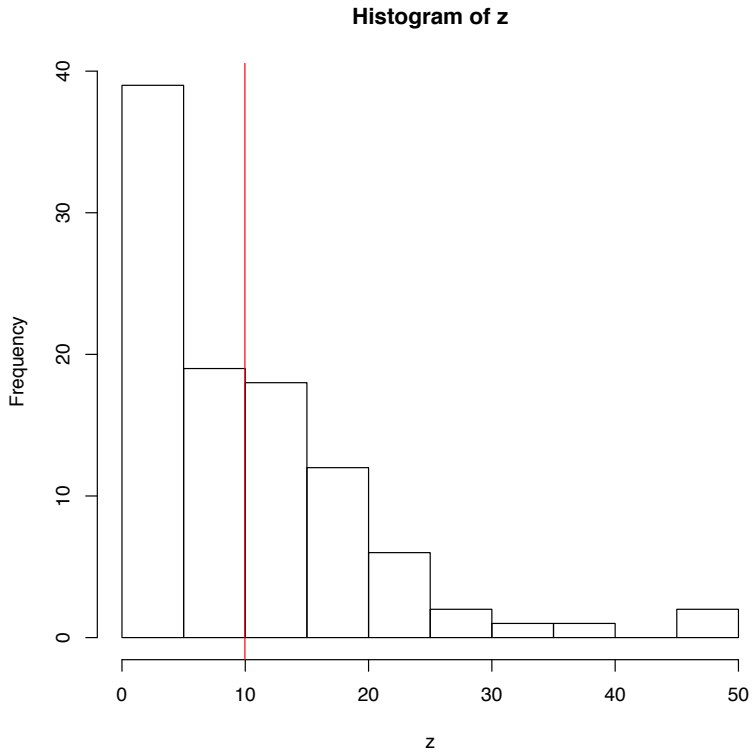
```
x <- read.csv("http://www.stat.yale.edu/~wdb22/Files.csv")
head(x)

##   type size
## 1  JPG 0.38
## 2  DOC 0.04
## 3  MP3 0.31
## 4  JPG 0.16
## 5  JPG 0.55
## 6  DOC 0.29

y <- x$size[x$type=="MP3"]
length(y)

## [1] 1010

hist(y)
M <- mean(y)
M

## [1] 10.34014

abline(v=M, col=2)
```



The histogram has a right-skewed shape. Imagine that I wanted to know the mean of this population, but (for some reason) its not possible for me to look at all of the file sizes. Instead I can only take a random sample of 100 of the files.[13] Then it's natural to use the sample mean to *estimate* the population mean. Let's try this.

[13] For the purposes of this example, this random sample should be done *with replacement*, meaning that once a file has been selected it can still get selected again. Othwerwise, the files sizes would not be independent.

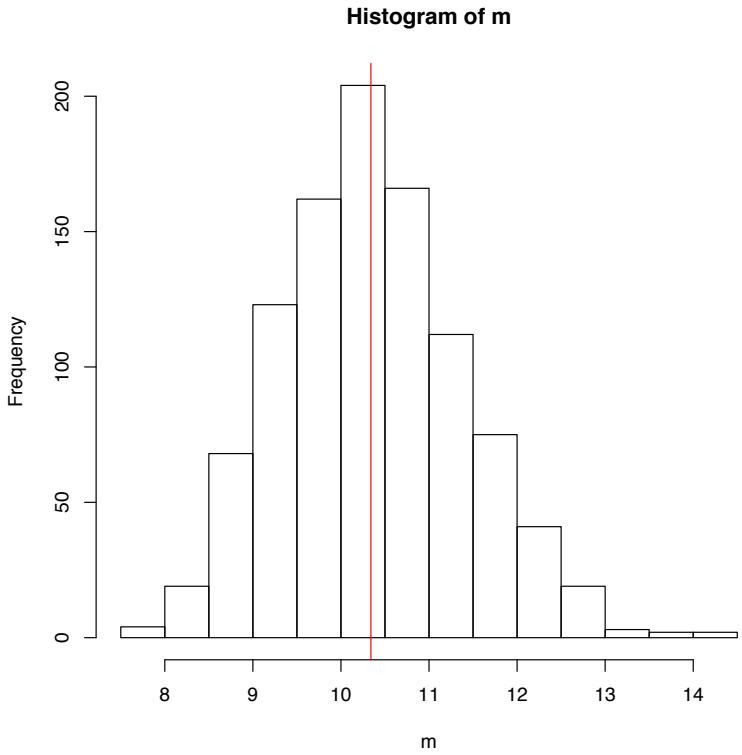
```
n <- 100
set.seed(1)
z <- sample(y, size=n, replace=T)
hist(z)
m <- mean(z)
m

## [1] 9.97

abline(v=m, col=2)
```



The sample mean is 9.97 which is pretty close to the true population mean. But let's think about the distribution of this sample mean. According to the CLT and our sample-size-30 rule of thumb, we might think that the sample mean should be approximately Normally distributed. What does this mean? It means that if we were to repeat the process of sampling 100 files over and over, the frequencies with which the sample mean takes different values should have a bell-curve shape. Let's try it out: repeat the process 1000 times and draw a histogram of the 1000 sample means that we calculate.

```
set.seed(1)
N <- 1000
m <- rep(NA, N)
for (i in 1:N) {
  z <- sample(y, size=n, replace=T)
  m[i] <- mean(z)
}
```

```
hist(m)
abline(v=M, col=2)
```



As expected, the histogram of sample means is bell-shaped.[14]

[14] I hope this example also demonstrates to you the value of *simulation*, which is one of the most powerful tools available to modern statisticians and data analysts.

The Normal distribution will play a key role in our inference techniques. Remarkably, many real-world processes result in bell-shaped data. Human heights are an example. In other words, a person's adult height is basically a draw from some Normal distribution. Why would this be? The Central Limit Theorem provides the key to understanding this phenomenon. It says that the sample mean of an iid sample has a distribution that increasingly resembles a Normal distribution as the sample size increases. If the sample mean has a bell-curve shape, then so does the sample sum. Your adult height is the *sum total* of a large number of your genes and factors of your environment, each of which typically has a small effect. These effects aren't iid, of course, but actually generalizations of the CLT have been proven that don't strictly need iid random variables. In a variety of cases, the distribution of a sum of random variables resembles a normal distribution. Thus real-world quantities that are well-modeled as a sum of random variables tend to have bell-shaped histograms.

3.2 *Inference*

In our discussion of probability, we have assumed that we know a probability distribution and then answered questions about what the data drawn from that distribution will look like. Usually, in real

life, we need to go the other direction: all we have is the data, and we want to think about what the distribution is.

Inference is the process of deriving statements about a probability distribution based on data drawn from that distribution. There are *three main inference tasks* that we will study.

1. *Estimation* means picking a "best" guess.
2. *Hypothesis testing* means deciding whether or not you should reject a proposition.
3. A *confidence interval* is a "best" set of guesses along with a quantification of your uncertainty.

3.2.1 Estimation

A statistic that is used as a guess for an unknown constant quantity is called an *estimator*. The most common example is the sample mean $\bar{X}$ of a n iid sample; it is often used as an estimator for the true mean μ of the population. As you saw in Section 3.1.4, the expected value of $\bar{X}$ is equal to μ; thus we say that $\bar{X}$ is an *unbiased* estimator[15] for μ. In fact, X_1 is also an unbiased estimator for μ (because μ is defined to be the expectation of the X_i). So instead of using the mean of the whole sample, you could just use the first observation to estimate μ. But recall from Section 3.1.4 that the variance if $\bar{X}$ is $1/n$ times the variance of X_1. That means the distribution of $\bar{X}$ is more concentrated around μ than X_1 is. In other words, $\bar{X}$ is probably going to be closer to μ than X_1 is, so $\bar{X}$ is a better estimator.

[15] Don't get too hung up on this property. Unbiasedness is not a particularly important criterion for deciding among possible estimators.

Often, you want to estimate the distribution of the observations. In such cases, you usually confine your search to a *parametric family* of distribution, such as the Normal distributions, and select the distribution among this family that *fits* the data best, according to some criteria. Selecting a distribution essentially means selecting values of the parameters that define the family.

3.2.2 Hypothesis Testing

The logic behind hypothesis testing is similar to that behind *proof by contradiction*. In a proof by contradiction, you assume a proposition is true, then show that logical contradictions follow from that assumption, thereby revealing that the assumption can't be true. In hypothesis testing, you assume a proposition is true, then if the data clashes with that assumption, you might conclude that the assumption is unlikely to be true after all.

The proposition that you assume is called the *null hypothesis*. How can you tell whether the "data clashes with the assumption"? You need to determine a statistic whose distribution you would know (or at least approximately know) if the null hypothesis were true. Then you calculate that *test statistic* from your dataset. If the test statistic is far away from where it "should" be, then that is taken as evidence against the null hypothesis. Formally, we calculate the probability

that the test statistic would be at least as extreme as the value we observed; this probability is called the *significance probability* (or the p-value). A low significance probability indicates that the dataset would be unlikely to look the way it does if the null hypothesis were in fact true. More precisely, letting p be the significance probability that you calculated, you can say "if the null hypothesis were true, the probability that the data would be at least as unusual as what I've observed is p."

Sometimes a threshold for the significance probability is specified ahead of time (often .05), that is used to decide whether or not the null hypothesis should be *rejected*; this threshold is called the *level* of the test. If the significance probability is below the threshold, then you reject the null hypothesis (you have a "statistically significant result"); otherwise you fail to reject the null hypothesis.[16]

[16] Just because you failed to reject the null hypothesis doesn't necessarily mean that you should think the null hypothesis is actually true.

Let's think through a simple example. Imagine that you want to decide whether or not to believe that a coin is fair. You flip the coin 20 times and get $H, H, T, H, T, H, H, T, T, T, H, H, T, H, H, T, H, H, H, H$. The null hypothesis is $p = .5$. Assuming the null hypothesis is true, the distribution of the number of heads is Binomial(20, .5); we'll use the number of heads X as our test statistic. There are 13 heads in this dataset, so the significance probability is the probability that a Binomial(20, .5) random variable would take a value at least as extreme as 13. Because the Binomial(20, .5) pdf is symmetric about its mean of 10, any value from 13 to 20 or from 0 to 7 would count as being at least as extreme as our observed test statistic value of 13. The significance probability is equal to the sum of the probabilities of these values, highlighted in Figure 3.6.

```
drawPMF(20)
k <- c(0:7, 13:20)
pmf <- sapply(k, dbinom, size=20, prob=.5)
points(k, pmf, col=2, type="h", lwd=2, ylim=c(0, max(pmf)))
```

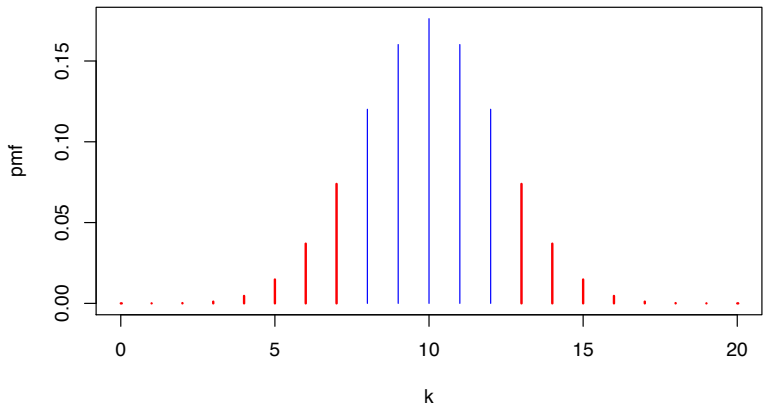


Figure 3.6: The significance probability is the sum of the heights of the red bars.

The significance probability is $P(X \leq 7) + P(X \geq 13)$. First, observe that[17]

$$\begin{aligned} P(X \geq 13) &= P(X > 12) \\ &= 1 - P(X \leq 12). \end{aligned}$$

[17] Or observe by symmetry that $P(X \leq 7) = P(X \geq 13)$.

We can use the cdf to calculate this in R.

```
pbinom(7, 20, .5) + (1 - pbinom(12, 20, .5))

## [1] 0.263176
```

The significance probability is about .26, meaning that the observed number of heads isn't all that unlikely from 20 flips of a fair coin. If we want a threshold of .05, this test fails to reject the null hypothesis that the coin is fair.

Of course, a 95% hypothesis test will not always give you the right answer. That's too much to ask! But when the null hypothesis is true, the test has only a $1/20$ chance of giving you the wrong answer (i.e. rejecting the null hypothesis). This type of mistake is called a *false positive*.[18]

[18] The *level* of a hypothesis test dictates what its false positive rate will be. The *false negative* rate is important too, but it is beyond the scope of this book.

3.2.3 Confidence Intervals

Instead of selecting a *single value* to guess an unknown constant quantity (i.e. estimation), you could select a *set of numbers* that hopefully contains the true value. You first need to determine how likely you want it to be that your set contains the true value. Typically, people want a 95% confidence interval, a set of numbers that has a .95 probability of including the true value. The more assurance you want, the larger your set of guesses must be.

You'll see your first example of finding a confidence interval in Chapter 8.

3.3 Conclusion

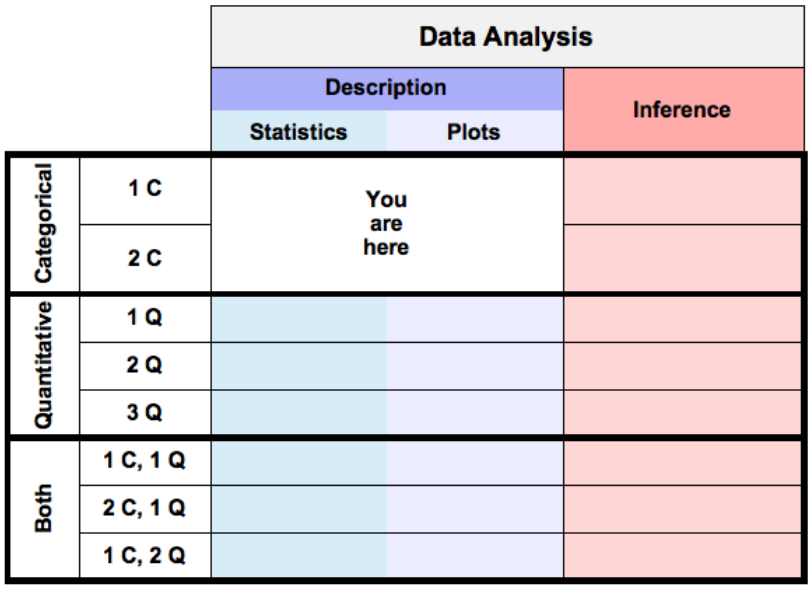
This chapter started with some basic definitions from probability theory, particularly *random variables* and *probability distributions*. The field of probability is about describing how a data (modeled by random variables) will behave based on their distribution. The field of statistical inference looks at things in the other direction: start with known data and try to describe the distribution that generated the data. We listed and defined three common inference tasks: estimation, hypothesis testing, and confidence intervals. Chapters 7 through 9 will present a number of common instances of these tasks.

Part II

Description

4 Description of Categorical Data

We begin our survey of descriptive data analysis by considering how to treat categorical variables. Recall that categorical variables give a category assignment to each observation in the dataset. In the descriptive stage of data analysis, we want to better understand the observations in our data set by calculating statistics and creating plots.



First, we will see how to analyze one categorical variable on its own. Then, we will see how to analyze two categorical variables together. In each case, you will learn what statistics can be calculated to summarize the data as well as some of the plots can be made to display it.

4.1 One Categorical Variable

In Chapter 1, we looked at a dataset of computer files (Example 1.2.1).

```
x <- read.csv("http://www.stat.yale.edu/~wdb22/Files.csv")
head(x)

##   type size
```

```
## 1  JPG 0.38
## 2  DOC 0.04
## 3  MP3 0.31
## 4  JPG 0.16
## 5  JPG 0.55
## 6  DOC 0.29
```

The data frame is made up of one categorical variable (`type`) and one quantitative variable (`size`). The `type` variable will be our example as we consider what statistics and plots can be used for a single categorical variable.

4.1.1 *Statistics*

The most obvious statistics to summarize a categorical variable are the counts of the number of observations within each category, which can be calculated using the `table` command.

```
tab <- table(x$type)
tab

##
##  DOC  JPG  MP3
## 4000 4990 1010
```

We will call this set of counts a *one-way frequency table*. If you are also interested in the proportion of observations belonging to each category, then you can divide each of these counts by the total number of observations to get the *one-way relative frequency table*.

```
tab/sum(tab)

##
##   DOC   JPG   MP3
## 0.400 0.499 0.101
```

If you were to draw one file at random from the set of files who are recorded in the frequency table, the relative frequency table gives the *probabilities* that the file is of each type. For this reason, this table is also called the *empirical distribution* on the categories.[1]

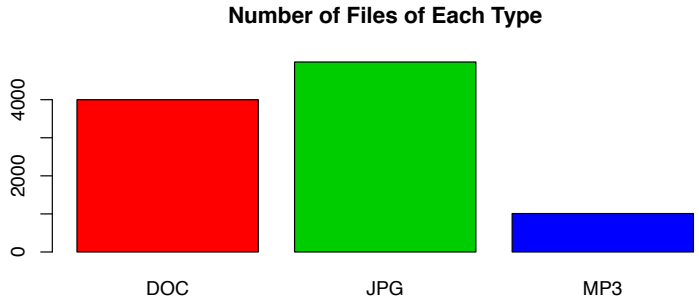
[1] Note that in the context of inference, we often talk about the underlying *distribution* that the data was drawn from. Don't confus concepts. In fact, at the description stage, we don't even need to think about the data being drawn from some underlying distribution; the data that we have is all we care about.

4.1.2 *Plots*

The *bar chart* is a simple and intuitive plot for displaying the count of each category, which is the same information as a one-way frequency table.[2] In fact, the R command `barplot` wants the data to already be in table form.

[2] In fact, it's useful for displaying any data that can be arranged into a one-way table, as you will see in Section 6.1.2.

```
barplot(tab, col=2:4,
            main="Number of Files of Each Type")
```



The height of the bar above each category tells you the count.

A *pie chart*, on the other hand, conveys the relative proportions of the different categories, the same information that a two-way relative frequency table tells. The command `pie` also takes one-way tables as input.

```
pie(tab, col=2:4,
    main="Proportion of Files of Each Type")
```

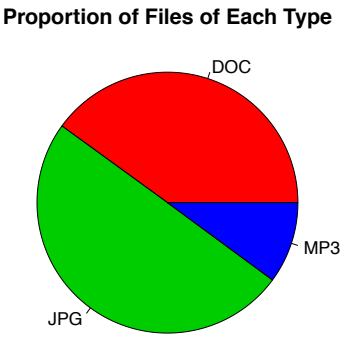


Figure 4.1: The pie chart shows the relative proportion of files that belong to each filetype.

The proportion of the circle's area that has a category's color is equal to the proportion of observations belonging to that category.

One-way frequency tables tell you the counts, while bar charts display them visually. One-way relative frequency tables tell you the proportions, while pie charts display them visually. These statistics and plots are especially straight-forward, so we won't discuss them further. With only one categorical variable, there's really not much to do. Things will get more interesting when we add more variables.

4.2 Two Categorical Variables

What if we have more than one categorical variable? We could simply make a table and barchart, as described above, for each of our variables, one at a time. This would help us understand the individual variables. However, when you plot more than one variable at a time, it can enable you to understand how the variables are *related* to each other, which is often much more interesting.

The computer files dataset only had one categorical variables, so we need to move on to another example dataset. This time, we will use `survey`, which comes with the built-in R package `MASS`.

```
library(MASS)
help(survey)
head(survey)
```

```
##      Sex Wr.Hnd NW.Hnd W.Hnd    Fold Pulse    Clap Exer Smoke Height
## 1 Female   18.5   18.0 Right  R on L    92    Left Some Never 173.00
## 2   Male   19.5   20.5  Left  R on L   104    Left None Regul 177.80
## 3   Male   18.0   13.3 Right  L on R    87 Neither None Occas     NA
## 4   Male   18.8   18.9 Right  R on L    NA Neither None Never 160.00
```

```
## 5   Male   20.0   20.0 Right Neither    35   Right Some Never 165.00
## 6 Female   18.0   17.7 Right  L on R    64   Right Some Never 172.72
##        M.I    Age
## 1   Metric 18.250
## 2 Imperial 17.583
## 3     <NA> 16.917
## 4   Metric 20.333
## 5   Metric 23.667
## 6 Imperial 21.000
```

We will look at the two variables `Exer` and `Smoke` which contain the students' responses to how often they exercise and how often they smoke.

4.2.1 *Statistics*

Again, the most obvious statistic to calculate with two categorical variables is a count. This time we will *count the number of observations that belong to each possible combination of categories.*[3]

[3] Or rather, R will count them.

```
table(survey$Exer, survey$Smoke)

##
##        Heavy Never Occas Regul
##   Freq     7    87    12     9
##   None     1    18     3     1
##   Some     3    84     4     7
```

Let's copy these variables into a new R object in case we want to modify them. Then if we make any mistakes or change our minds about anything, we can easily start over with the original data still safe and sound in the "survey" object.

```
y <- survey[, c("Exer", "Smoke")]
head(y)

##   Exer Smoke
## 1 Some Never
## 2 None Regul
## 3 None Occas
## 4 None Never
## 5 Some Never
## 6 Some Never

table(y)

##       Smoke
## Exer   Heavy Never Occas Regul
##   Freq     7    87    12     9
##   None     1    18     3     1
##   Some     3    84     4     7
```

The variable `Smoke` has four categories, some of which contain only a few observations. For simplicity, it might make sense to recategorize the values as simply "Never smokes" if they responded that they never smoke and "Smokes" otherwise.

```
levels(y$Smoke) <- c("Smokes", "Never smokes", "Smokes", "Smokes")
table(y)

##       Smoke
## Exer   Smokes Never smokes
##   Freq     28           87
##   None      5           18
##   Some     14           84
```

Notice that the categories of `Exer` have a natural ordering,[4], but the table is listing them in a different order. The `factor` command allows us to change the order of the categories.

[4] None < Some < Freq

```
# Change the order of Exer from (Freq, None, Some)
# to (None, Some, Freq)
y$Exer <- factor(y$Exer, levels(y$Exer)[c(2, 3, 1)])
# And reverse the order of Smoke
y$Smoke <- factor(y$Smoke, levels(y$Smoke)[2:1])
tab <- table(y)
tab

##       Smoke
## Exer   Never smokes Smokes
##   None           18      5
##   Some           84     14
##   Freq           87     28
```

We will call this set of statistics a *two-way frequency table*. If we divide by the total number of observations counted, we get a *two-way relative frequency table*.

```
# Divide the table by the sum of the counts
# and round the output to two decimal places.
joint <- round(tab/sum(tab), 2)
joint

##       Smoke
## Exer   Never smokes Smokes
##   None         0.08   0.02
##   Some         0.36   0.06
##   Freq         0.37   0.12
```

We could also call this table the *empirical joint distribution* of the two variables.

In this context, the individual variables' counts and empirical distributions are known as *marginal frequencies* and the *empirical marginal*

distribution to distinguish them from the joint frequencies and empirical joint distribution. To find the marginal frequency of `Smoke`, you can use the table command as in 4.1.1 or you can take the sums along the columns of the two-way frequency table.[5]

[5] Actually, if you try this with `Exer` instead, you will find that the marginals do not match! This is because the variable `Smoke` is missing a value; that is, one respondent did not tell the surveyors whether or not he smoked. R ignored that row when computing the two-way table for `Exer` and `Smoke` together, but it included that observation when computing the one-way table for `Exer`.

```
# Make a one-way table for Smoke
table(y$Smoke)

##
## Never smokes       Smokes
##          189           47

# Find the sums of the two-way table's columns
marginalFreq <- apply(tab, 2, sum)
marginalFreq

## Never smokes       Smokes
##          189           47
```

Recall that empirical marginal distributions can be found by dividing the marginal frequencies by the total count. They can also be found by simply summing the joint distribution along the direction of interest.

```
# Divide the marginal frequencies by the total count
round(marginalFreq/sum(tab), 2)

## Never smokes       Smokes
##          0.8           0.2

# Sum the joint distribution values along the columns
apply(joint, 2, sum)

## Never smokes       Smokes
##         0.81          0.20
```

In the first case, we took the sums then divided each one by the total count. In the second case, we divided each number by the total count, then took sums. These are mathematically equivalent.[6]

[6] But you may get slightly different numbers due to rounding.

Finally, empirical conditional distributions are wonderful for elucidating the relationships between variables. The *conditional distribution* tells you the probabilities distribution that one variable must have, given that you know the value of another variable.[7] For instance, imagine I were to select a student at random and ask him whether or not he smokes. If I find that he smokes, then what is the probability that he exercises frequently? Well, because I know that he smokes, I know that he is represented in the right-hand column of the two-way table, the smokers. Because he was randomly sampled, he is equally likely to be any of the smokers. Therefore, the probability that he is exercises frequently is just the number of smokers who exercise frequently divided by the total number of smokers, which is $28/(5+14+28) \approx 0.60$. That isn't the same as the overall probabil-

[7] This is discussed in Chapter 3.

ity of exercising frequently if we hadn't learned that the person was a smoker, which is approximately[8] $0.37 + 0.12 = 0.49$. In R, the conditional distribution of `Exer` given that `Smoke` takes the value `Smokes` can be found by dividing each entry of the `Smoke` column by that column's sum.[9] The `prop.table` command takes a two-way table and provides conditional distributions; the second argument determines which variable to condition on.

[8] This is the sum across the bottom row of the joint distribution.

[9] This can be done using either the two-way frequency table or the two-way relative frequency table.

```
# The conditional distributions of Smoke given Exer
prop.table(tab, 1)

##       Smoke
## Exer   Never smokes    Smokes
##   None    0.7826087 0.2173913
##   Some    0.8571429 0.1428571
##   Freq    0.7565217 0.2434783

# The conditional distributions of Exer given Smoke
prop.table(tab, 2)

##       Smoke
## Exer   Never smokes    Smokes
##   None    0.0952381 0.1063830
##   Some    0.4444444 0.2978723
##   Freq    0.4603175 0.5957447
```

4.2.2 *Plots*

One way to visually display the information in a two-way frequency table is a plot we will call *multiple bar charts*. It's exactly what it sounds like! You've already seen bar charts, so this simple extension will be easy for you to interpret.

```
barplot(tab, beside=TRUE, legend=rownames(tab), col=2:4,
        main="Students' Smoking and Exercise Habits")
```

Notice that you could have made a bar chart for each of the different categories of `Exer` and used colors for the `Smoke` variable instead. This can be achieved by using the *transpose* of the original table `tab`.[10]

[10] In mathematics, transposing a matrix means switching the rows with the columns.

```
barplot(t(tab), beside=TRUE, legend.text=TRUE, col=2:3,
        main="Students' Smoking and Exercise Habits",
                args.legend=list(x="topleft"))
```

In practice, which variable should you put where? If you're thinking about one of the variables "explaining" the other variable, then each of the categories of the explaining variable should get its own bar charts, while the explained variable is color-coded.[11] For instance, if we think that perhaps the health effects of smoking could make it hard to exercise, then we would prefer Figure 4.3

[11] If you don't have any explanation in mind, then you may want to make both plots for yourself and choose the one that looks best or makes the most sense.

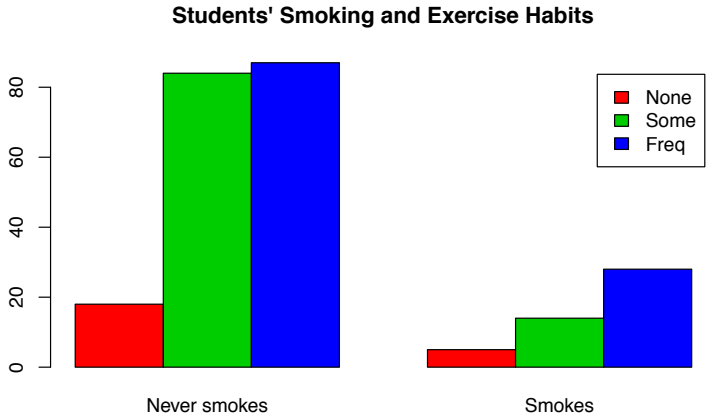


Figure 4.2: An example of multiple bar charts.

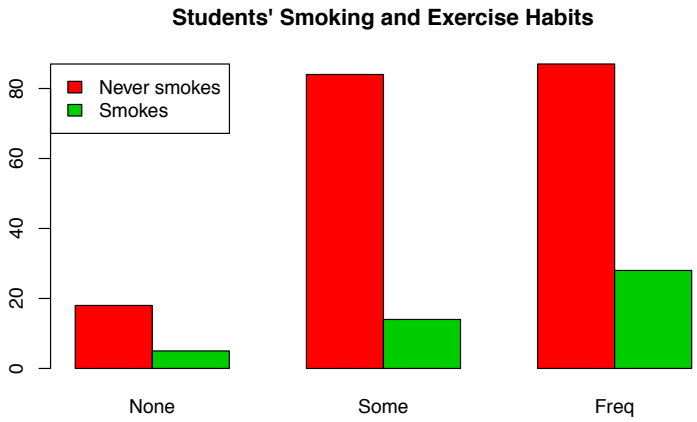
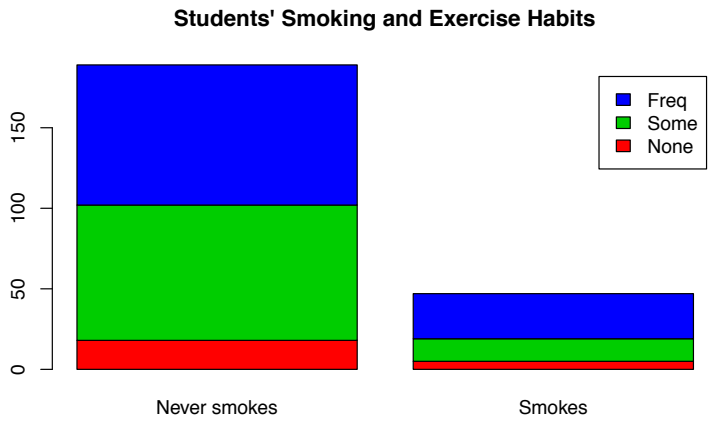


Figure 4.3: Another plot of multiple bar charts, this time reversing the roles of the variables.

A second type of plot for two-way frequency tables is *stacked bar charts*. It uses the same `barplot` command but without setting the `beside` parameter to `TRUE`. This time it's harder to see the specific counts within every combination of categories, but you can now see the total counts easily (the marginal frequencies) for the variable labeled on the horizontal axis.

```
barplot(tab, legend.text=TRUE, col=2:4,
        main="Students' Smoking and Exercise Habits")
```



Again, you get to choose which variable plays which role, and the same rule applies: each category of the explaining variable should get its own stacked bar.

If you are more interested in displaying and comparing the two-way relative frequencies, an excellent choice is the *mosaicplot*. It is analogous to the pie chart in that the amount of area corresponding to each combination of categories is proportional to the count.

```
mosaicplot(tab, main="Students' Smoking and Exercise Habits",
           color=3:4, cex=.9)
```

The plot shows you the empirical joint distribution but treats the two variables differently. It lets you see at a glance the empirical marginal distribution of the variable whose categories are listed along the top: the probability of each category is proportional to the width of its column. And looking within a column, you can easily see the empirical conditional distributions that result from conditioning on those categories. For instance, in Figure 4.4, the `Freq` column has more blue than the `Some` column. This means that a randomly selected student who exercises frequently is less likely to smoke than a randomly selected student who exercises some.

Obviously, the roles can be switched.

```
mosaicplot(t(tab), main="Students' Smoking and Exercise Habits",
           color=2:4, cex=.9)
```

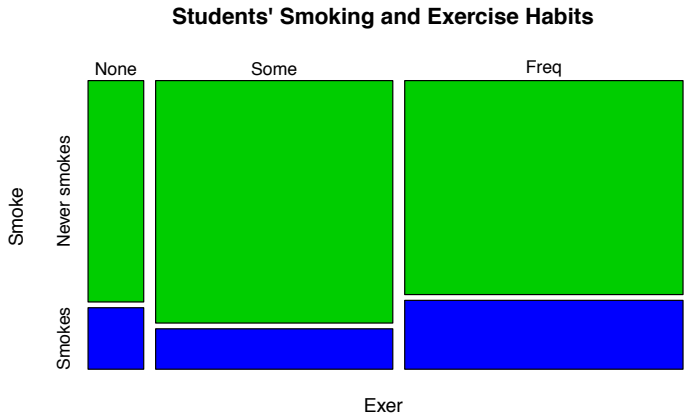
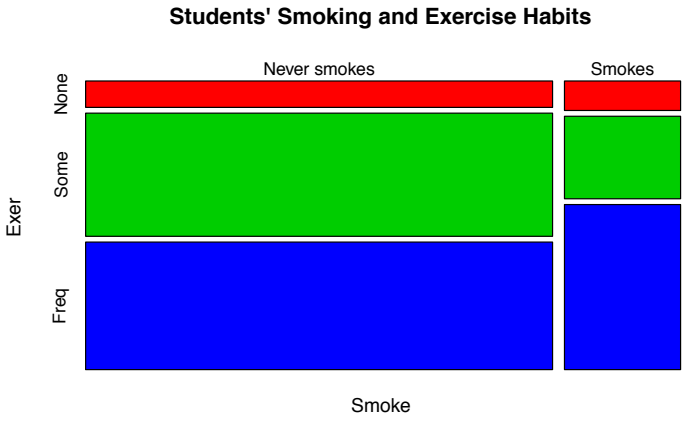


Figure 4.4: In a mosiacplot, the area of each rectangle represents the proportion of observations that belong to each combination of categories. We can see that most of the subjects in the survey are non-smokers who exercise some or frequently.

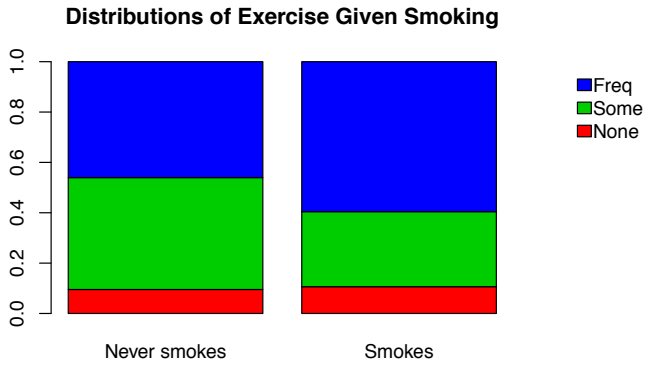


As before, you want to be able to easily see the conditional distributions that are formed by conditioning on the explaining variable.

If all you care about are the conditional distributions, then the mosaicplot's varying sizes of the columns may be distracting. In that case, a good technique is to draw a *rescaled stacked bar chart*, which is like the stacked bar chart except that the bars have been rescaled to have the same total height.[12]

[12] There reason this code looks a little complicated is that the legend is placed beside the plot. Without the extra instructions, the legend would obscure the boxplots.

```
ptab <- prop.table(tab, 2)
par(mar=c(5, 5, 4, 10))
barplot(ptab, legend.text=TRUE, col=2:4,
        args.legend=list(x=3.2, y=1, bty="n", x.intersp=.1),
                main="Distributions of Exercise Given Smoking")
par(mar=c(5.1, 4.1, 4.1, 2.1))
```



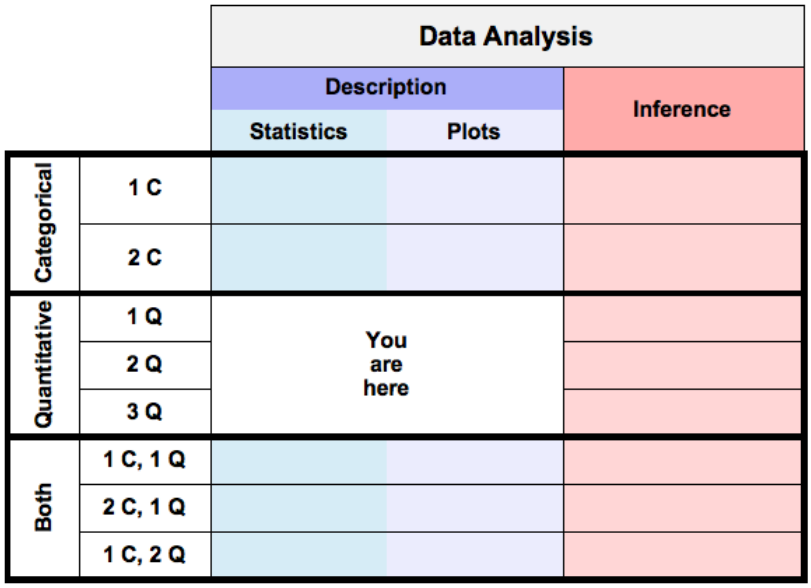
4.3 *Conclusion*

In this chapter, we have learned some of the statistics and plots that can be useful when dealing with categorical variables. In fact, for each statistic we discussed, we also pointed out at least one plot that displays the same information visually. The table below summarizes these statistics and plots and their relationships; a dotted line from a statistic to a plot signifies that the plot intuitively displays that statistic. This table fills in some of the details of our big picture (Figure 1.3).

	Description	
	Statistics	**Plots**
One Categorical Variable	one-way frequency table	bar chart
	one-way relative frequency table	pie chart
Two Categorical Variables	two-way frequency table	multiple bar charts stacked bar charts
	two-way relative frequency table	mosaicplot
	conditional distributions	rescaled stacked bar charts

5 Description of Quantitative Data

We now turn to quantitative variables, those in which each observation has a numerical measurement. As we continue our survey of the descriptive stage of data analysis, recall how this chapter fits into the big picture.



This time, we will consider the descriptive analysis of one, two, or three quantitative variables; in each case, we will describe some of the most common and useful statistics and plots.

5.1 One Quantitative Variable

In this chapter, we will use student grades as our example dataset. In a statistics class that I taught, there were five homework assignments, four quizzes, and one final exam. The data[1] is posted on the web.

[1] This isn't the real data; I generated these numbers using R.

```
# Read in the data from the web
x <- read.csv("http://www.stat.yale.edu/~wdb22/Grades.csv")

# How many students are there?
# How many grades were given to each student?
dim(x)
```

```
## [1] 15 10

# Display the first six rows of the data frame
head(x)

##   Exam HW1 HW2 HW3 HW4 HW5 Quiz1 Quiz2 Quiz3 Quiz4
## 1   12  25  52  83  25   7    23    56    59    12
## 2   21  52  12  11  91  42    12    90    47    21
## 3   98  68  96  42  69  94    39    84    71    98
## 4   71  14  76  58  62   5    44    12    60    71
## 5   36  70  13  99  12  30    97    75    14    36
## 6   35  96  94  12  61  18    43    86    65    35
```

Now, let's see what tools are available for the description stage, starting with a look at the exam scores.

5.1.1 *Statistics*

There are only fifteen observations in the data frame, so let's go ahead and take a look at all the values in our variable of interest, `Exam`.

```
x$Exam

##  [1] 12 21 98 71 36 35 21 33 75 49 15 27  2 38 92
```

You can get a better sense of the data by sorting the numbers.

```
sort(x$Exam)

##  [1]  2 12 15 21 21 27 33 35 36 38 49 71 75 92 98
```

There are two main aspects of a quantitative variable that are often summarized: location[2] (where the numbers are) and spread (how far apart the numbers are from each other).

[2] Location is also known as "center" in some books.

The most common statistic summarizing the location of the data is the *mean*, the sum of the values divided by the number of observations. It's usually what people mean when they say "average." The mean is nice because it is intuitive and familiar, but one drawback is that it is not *robust*. One unusual data point can make the mean go completely off-target. For instance, pretend you have a dataset purporting to give five people's heights in meters.

```
heights <- c(1.8, 1.6, 1.6, 1.7, 190)
mean(heights)

## [1] 39.34
```

Obviously something is wrong with the data, because nobody is 190 meters tall! Perhaps that height was recorded in centimeters by mistake. Regardless, it only took one bad value to give you a bad mean. And this is a real concern; in practice, wild data values are all too frequent.

On the other hand, consider the *median*, the middle value of the sorted numbers.[3] In this case, the wild number doesn't hurt us.

[3] If there are an even number of observations, then you take the mean of the two middle values.

```
median(heights)

## [1] 1.7
```

The median is sometimes preferred as it is more robust. In some cases it is also more representative of what the observations actually look like. As an example, imagine you want to summarize the incomes in a small town of 100 people; 99 of them make 25 thousand dollars per year while the other one makes 100 million. The mean income is about a million dollars, but reporting that might give a very misleading impression of the situation. There is actually nobody in the town who is well-described by the mean. On the other hand, the median income of 25 thousand dollars does accurately describe 99 of the 100 people.

The *quartiles* extend the idea of the median to devise additional statistics that give other details about the variable's location. The median is a number that splits the sorted data values in half. You can also split those halves in half again. A number that splits the lower half is called the *first quartile*, while a number that splits the upper half is the *third quartile*.[4] If about a quarter of the data lies below the first quartile, and a quarter lies above the third quartile, that means the "middle half" the data lies between these quartiles. So knowing the median and the quartiles gives you a pretty good summary of the variable's location.[5]

The location statistics we have discussed are calculated by the `summary` command.

[4] I won't describe precisely how to calculate these quartiles because there is some disagreement about it. It's fine to just use whatever quartiles R tells you.

[5] The 1/4, 1/2, and 3/4 data values are intuitive, but sometimes you might be interested in splitting the data into other proportions. The more general notion is called a *quantile* or a *percentile* if you're using percentages.

```
summary(x$Exam)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00   21.00   35.00   41.67   60.00   98.00
```

Next, we will cover two statistics that are often used to summarize the "spread" of a quantitative variable. When the mean is reported to summarize location, typically an estimate of *standard deviation* is reported to summarize spread.[6] In R, the `sd` command gives us this quantity; to calculate it manually, take the sum of the squared deviations from the sample mean, divide by the number of observations minus one, then take the square root of the result. If the variables values are $\mathbf{x} = (x_1, \ldots, x_n)$ and $\bar{x}$ represents the sample mean, then the estimated standard deviation is

[6] The quantity calculated by R is an estimate of the population's standard deviation, assuming the data is randomly sampled. Even if we don't care about inference, this quantity still gives us an indication of spread.

$$\hat{\sigma} := \sqrt{\frac{1}{n-1}\sum(x_i - \bar{x})^2}$$

```
# The estimated standard deviation
sd(x$Exam)

## [1] 29.43678

# Calculating the same quantity manually
n <- length(x$Exam)
m <- mean(x$Exam)
SS <- sum((x$Exam-m)^2)
sqrt(SS/(n-1))

## [1] 29.43678
```

The other common statistic for spread is the *interquartile range* (or IQR), which is the third quartile minus the first quartile. It tells you how wide of an interval you would need to cover the middle half of the data.[7]

[7] Don't confuse this with the *range*, which is the interval from the minimum data point to the maximum data point.

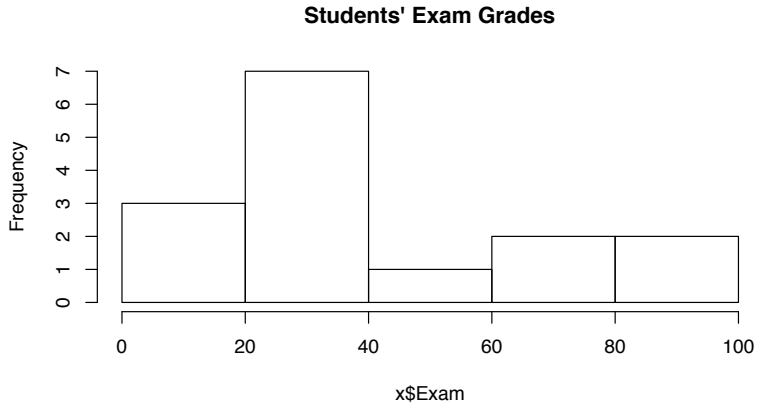
```
s <- summary(x$Exam)
names(s)

## [1] "Min."    "1st Qu." "Median"  "Mean"    "3rd Qu." "Max."

IQR <- as.numeric(s["3rd Qu."] - s["1st Qu."])
IQR

## [1] 39
```

Ultimately, the statistics that you calculate will depend on the purpose of your data analysis. You may even want to "make up" a statistic that we haven't talked about here if it's relevant to your questions. Remember: be creative!

5.1.2 *Plots*

A *histogram* is a simple plot to help you visualize one quantitative variable. It divides the range into "bins" and places a bar above each bin with height equal to the number of data points in that subinterval.

```
hist(x$Exam, main="Students' Exam Grades")
```

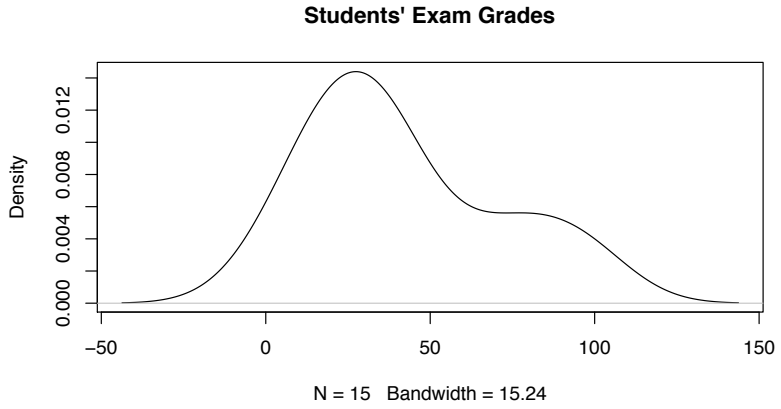


This tell us that there were three students with scores between 30 and 40, three students with scores between 40 and 50, six students with scores between 50 and 60, and so on. If you really want to, you can tell R how to split the range up into bins,[8] but I've found that R's default behavior is typically fine.

[8] Use `help(hist)` for details.

Similar to the histogram is the *density plot*. It is a "smoothed-out" curve that resembles the histogram and is normalized to have a total area under the curve equal to 1.

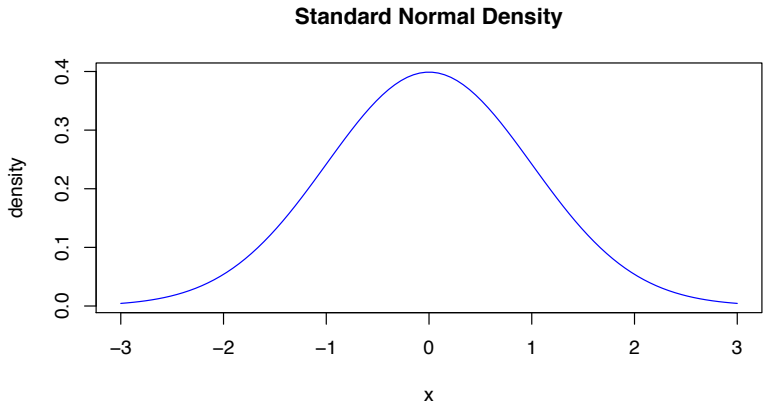
```
plot(density(x$Exam), main="Students' Exam Grades")
```



One common feature to look for in histograms is a *bell curve* shape. The "bell curve" refers to the shape of the Normal distribution's probability density function, shown in Figure 5.1. In this idealized case, two thirds of the data points are within one standard deviation of the mean, 95 percent of the data points are within two standard deviations of the mean, and 99 percent of the data points are within three standard deviations of the mean. These quantities (67%, 95%, 99%) are good to remember.

```
# Draw the standard normal density curve
grid <- seq(-3, 3, length.out=100)
plot(grid, sapply(grid, dnorm), type="l", col=4,
     xlab="x", ylab="density", main="Standard Normal Density")
```

Figure 5.1: The standard Normal distribution's probability density function on the interval from -3 to 3.

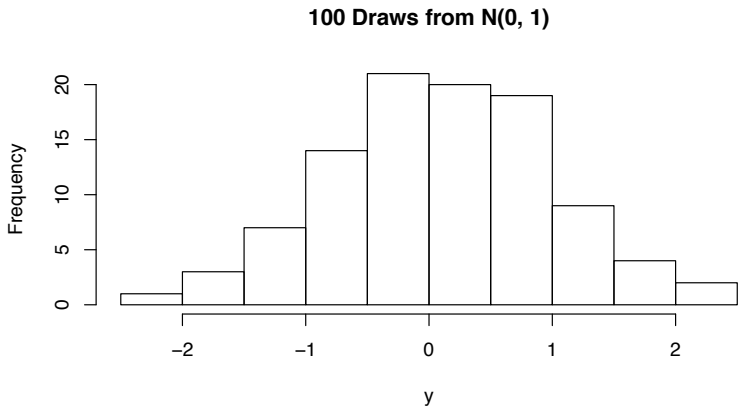


Often the histograms and density plots of real data will have this same basic shape, with the bulk of the data points together near the center and a roughly symmetric *tails* extending outward to either side in roughly a $(67\%, 95\%, 99\%)$ distribution.[9] Figure 5.2 gives an idealized example showing 100 simulated draws from an actual standard normal distribution.

[9] Chapter 3 discusses why many real-world random quantities tend to be like the Normal.

```
set.seed(1)
y <- rnorm(100)
hist(y, main="100 Draws from N(0, 1)")
```

Figure 5.2: Histogram of 100 independent draws from the Standard Normal distribution.



Recognizing the bell curve shape will return to play a major role when we cover inference for quantitative variables. For the description stage, realize that identifying data as bell-shaped is an important part of your summary.

Another common shape for histograms is having most of the data points close together and a heavy tail in one direction.[10] Often you find this property with data that has a natural cutoff on one side but no bound on the other side, such as variables that can only take positive values. Income is a classic example, as shown in Figure 5.3.

[10] The sample is said to be *skewed* in the direction of the heavy tail.

```
library(np)

## Nonparametric Kernel Methods for Mixed Datatypes (version
0.60-2)
## [vignette("np_faq",package="np") provides answers to frequently
asked questions]

data(wage1)
dim(wage1)

## [1] 526  24

names(wage1)

##  [1] "wage"     "educ"     "exper"    "tenure"   "nonwhite" "female"
##  [7] "married"  "numdep"   "smsa"     "northcen" "south"    "west"
## [13] "construc" "ndurman"  "trcommpu" "trade"    "services" "profserv"
## [19] "profocc"  "clerocc"  "servocc"  "lwage"    "expersq"  "tenursq"

hist(wage1$wage)
```

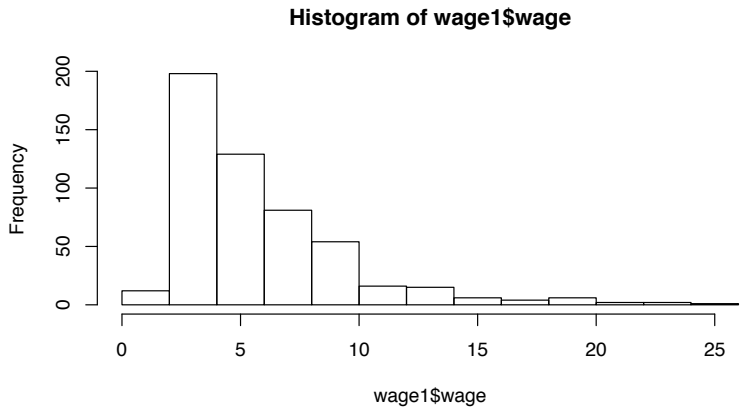


Figure 5.3: The histogram shows 526 observations of hourly wage sampled randomly from the 1976 U.S. Current Population Survey. This data does not resemble a bell-shaped curve; it has a heavy tail on the right side.

Often we can take a simple transformation (such as a logarithm or a square root) of each of the data points to produce data values that are closer to a bell curve. We will take the natural logarithm transformation of the incomes and see what happens.

```
log.wage <- log(wage1$wage)
hist(log.wage)
summary(log.wage)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.6349  1.2030  1.5370  1.6230  1.9290  3.2180
```

Why would we want to do that? What's so good about bell curves? For now, just think of them as being easy to summarize. If I say that the log incomes have a bell curve shape and I report the `summary` to you, that gives you a pretty good idea of the shape of the data,

Figure 5.4: The histogram shows the logarithms of the 526 incomes. It more closely resembles a bell curve than the original histogram did.

even without seeing Figure 5.4. It would be much harder for me to concisely convey the general shape of the original income data in Figure 5.3. The more important reason to make this transformation will become clear when we discuss inference (Part III), where bell curves will play a central role.

Finally, several of the statistics we've described are visible on a *boxplot*, which is demonstrated in Figure 5.5. The box represents the middle half of the data: the top side is the third quartile, the bottom side is the first quartile, and the horizontal line in-between is the median.[11] The upper and lower vertical lines extending from the box reach as far as the most extreme data value that is within 1.5∗IQR of the box. Data points beyond that distance from the box are considered *outliers* and plotted separately.[12]

[11] It follows that the vertical length of the box is the IQR.

[12] In general, there's no single hard definition of "outlier." It's more of a subjective term for any data point that seems far from the bulk of the data.

```
# Recall the "summary" command from earlier
summary(x$Exam)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00   21.00   35.00   41.67   60.00   98.00

# Compare the summary output to the boxplot
boxplot(x$Exam, main="Students' Exam Scores")
```

Figure 5.5 has no outliers, but a second example Figure 5.6 has three.

```
set.seed(1)
y <- rexp(30)
boxplot(y, main="30 Draws from Exp(1)")
```

Every data analyst will regularly find outliers in the course of his or her work, and such points often require special attention. In some cases, there may have been a mistake on the part of the person recording the data. Other times, they are particularly interesting cases that are relevant to the question you're trying to answer. Depending on how far out the outliers are, you sometimes want to

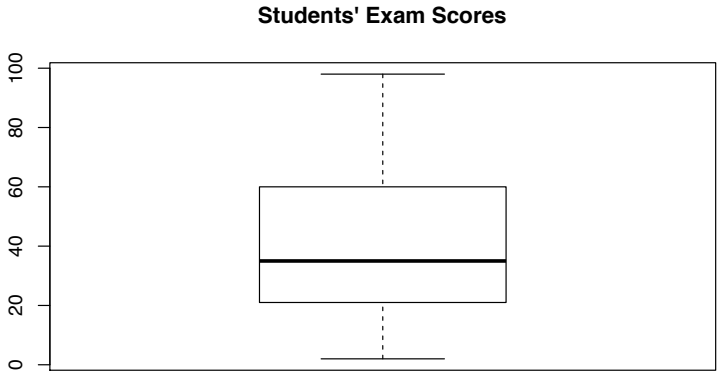


Figure 5.5: The boxplot displays a summary of the shape and location of the student's exam scores. The middle half of students scored in the 20 to 60 range, but overall the range of grades spans almost the entire interval from 0 to 100.

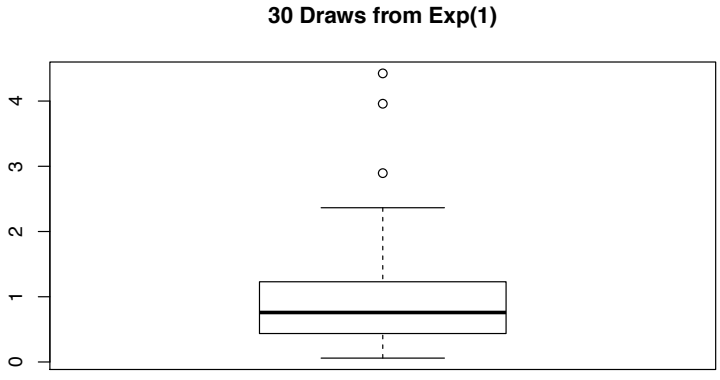


Figure 5.6: A boxplot of thirty numbers randomly generated by R according to the standard exponential distribution.

exclude them from the analysis of the rest of the data. Otherwise, the outliers can have a disproportionate effect on your summary.[13]

[13] Recall our discussion of the mean's lack of robustness.

5.2 Two Quantitative Variables

Continuing with the grades data, let's pose a specific problem. Imagine that my goal is to assign letter grades to the students in a sensible way. This is an example of a data analysis task in which I am only interested in the observations at hand. I have no desire to speculate about any individuals who weren't in the dataset, so there would be no inference stage to this data analysis.

The students were told that the lowest quiz score would be dropped, and that the final exam counts as three quizzes. Using those two rules, we can calculate a homework average and a quiz/exam average for each student.[14]

[14] The operations to calculate these averages are included for the sake of completion, but don't worry about understanding them for now. Keep your focus on understanding the main ideas.

```
# Calculate each student's homework average and quiz/exam average
x$HW.avg <- apply(x[, 2:6], 1, mean)
x$Quiz.avg <- (apply(x[, 7:10], 1, sum) - apply(x[, 7:10], 1, min))/3
x$Quiz.avg <- (x$Quiz.avg + x$Exam)/2

# We have created two additional variables in the data frame
head(x)
```

```
##   Exam HW1 HW2 HW3 HW4 HW5 Quiz1 Quiz2 Quiz3 Quiz4 HW.avg Quiz.avg
## 1   12  25  52  83  25   7    23    56    59    12   38.4 29.00000
## 2   21  52  12  11  91  42    12    90    47    21   41.6 36.83333
## 3   98  68  96  42  69  94    39    84    71    98   73.8 91.16667
## 4   71  14  76  58  62   5    44    12    60    71   43.0 64.66667
## 5   36  70  13  99  12  30    97    75    14    36   44.8 52.66667
## 6   35  96  94  12  61  18    43    86    65    35   56.2 49.83333
```

5.2.1 Statistics

We've already covered the common statistics for characterizing a single quantitative variable. Now that we're looking at two quantitative variables together, we'll discuss correlation and the (closely related) least-squares line, both of which help us understand the relationship between the variables. The *correlation* formula looks a little complicated. Let $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} := (y_1, \ldots, y_n)$ denote two variables (columns of our data frame). And define $\bar{x}$ and $\bar{y}$ to be the means of the respective vectors.

$$\text{cor}(\mathbf{x}, \mathbf{y}) := \frac{\sum[(x_i - \bar{x})(y_i - \bar{y})]}{\sqrt{\sum(x_i - \bar{x})^2}\sqrt{\sum(y_i - \bar{y})^2}}$$

Of course, in practice you let R do this calculation for you.

```
cor(x$HW.avg, x$Quiz.avg)

## [1] 0.6962957
```

The correlation indicates the linear[15] relationship between the variables; it always takes a value between -1 (exact negative linear relationship) and 1 (exact positive linear relationship). This point becomes clearer in the context of the least-squares line. The *least-squares line* is the line ($ax + b$ for some a and b, which are known as *parameters*) that is closest to the data points in a specific sense.[16] In particular, it minimizes the sum of the squared *residuals*, where the residuals are the differences between the y-values and the fit, in this case

$$r_i := y_i - (ax_i + b).$$

Let us use $\hat{a}$ and $\hat{b}$ to denote the slope and intercept (the two parameter values) of the least-squares line (i.e. the line minimizing $\sum r_i^2$). It is easy to show that

$$\hat{a} = \text{cor}(\mathbf{x}, \mathbf{y}) \frac{\hat{\sigma}_y}{\hat{\sigma}_x} \qquad \text{and} \qquad \hat{b} = \bar{y} - \hat{a}\bar{x}$$

where $\hat{\sigma}_x$ and $\hat{\sigma}_y$ represent the estimated standard deviations of $\mathbf{x}$ and $\mathbf{y}$ as described in Section 5.1.1. Think of the line as a way of predicting a value for y_i once you know x_i; call the prediction $\hat{y}_i$.

$$\hat{y}_i := \hat{a}x_i + \hat{b}$$

Plugging in $\hat{a}$ and $\hat{b}$ and rearranging, this equation can also be written as

$$\frac{\hat{y}_i - \bar{y}}{\hat{\sigma}_y} = \text{cor}(\mathbf{x}, \mathbf{y}) \frac{x_i - \bar{x}}{\hat{\sigma}_x}$$

The quantity $\frac{x_i - \bar{x}}{\hat{\sigma}_x}$ represents the number of standard deviations above the mean the ith observation's x-value is. The correlation tells you what to multiply this by to predict the number of standard deviations above the mean that the y-value will be. For instance, in our grades example, we found that the correlation between homework average and quiz average was about 0.7. Therefore, if you learn that Jack's homework average was two standard deviations above the mean, then the least-squares line predicts that Jack's quiz average is 1.4 standard deviations above the mean. The `lm` command[17] in R tells you the coefficients of the least-squares line.

```
lm(x$Quiz.avg ~ x$HW.avg)

##
## Call:
## lm(formula = x$Quiz.avg ~ x$HW.avg)
##
## Coefficients:
## (Intercept)     x$HW.avg
##     -4.2967       0.9955
```

[15] Technically, the term is *affine linear* when the line is allowed to have a non-zero y-intercept.

[16] This will make more sense once you've seen scatterplots in the next section.

[17] "lm" stands for "linear model."

We'll return to this command when we cover inference for quantitative variables in Chapter 8.

When finding a least-squares line[18], you have to choose one variable to play the role of the "y" variable (called the *response variable*) and another to play the role of the "x" variable (called the *explanatory variable*). Typically, if one variable intuitively seems more like it, in part, explains the behavior of the other variable, then this explaining variable is typically used as the explanatory variable.[19] In our example of homework averages and quiz averages, one might think that working hard on homework improves a student's quiz scores; thus homework average seems like a somewhat more natural choice to be the explanatory variable.

A final statistic that we will discuss is called R^2. It is defined to be 1 minus the ratio of the the variance of the residuals over the variance of the orginal response variable $\mathbf{y}$.[20]

$$R^2 := 1 - \frac{(1/n)\sum(r_i - \bar{r})^2}{(1/n)\sum(y_i - \bar{y})^2}$$

where n is, of course, the sample size, and $\bar{r}$ is, of course, the mean of the residuals.[21] You can think of it as the proportion of "variation" remaining in the data after *fitting*. In the case of a straight line fit for two variables, R^2 turns out to be equal to the squared correlation.

```
cor(x$HW.avg, x$Quiz.avg)^2

## [1] 0.4848277
```

We've discussed statistics that summarize the linear relationship between two quantitative variables. But the relationship between the variables may not be well-captured by a line. Let's see what plots are available to help us figure this out.

5.2.2 Plots

The *scatterplot* is simple but incredibly useful; it draws each (x_i, y_i) pair (each observation's x and y values) as a point on the two-dimensional plane.

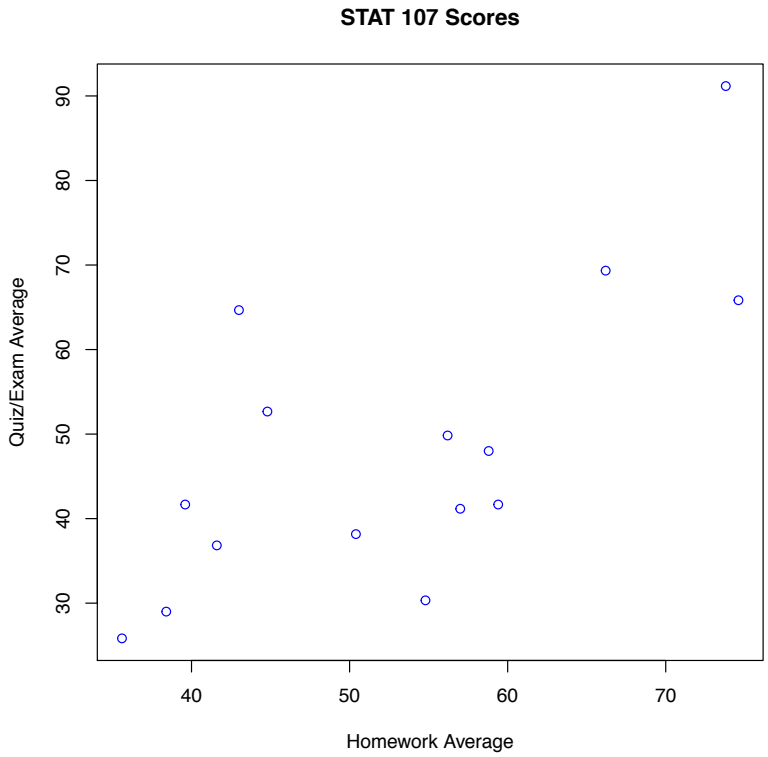
```
# Scatterplot of students' averages
plot(x$HW.avg, x$Quiz.avg, col=4, xlab="Homework Average",
     ylab="Quiz/Exam Average", main="STAT 107 Scores")
```

[18] This process is also known as *simple linear regression*.

[19] However, sometimes you are planning to get future measurements of one variable and you want to use them to predict the value of the other variable. In that case, the variable that you will get future data on should be the explanatory variable.

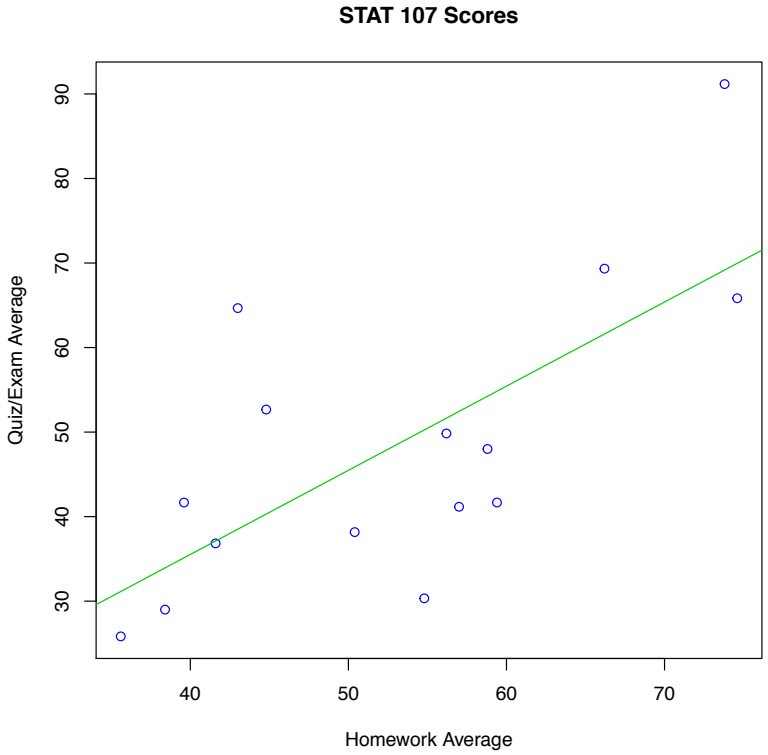
[20] Variance was defined in Chapter 3. It is the average squared deviation from the mean.

[21] The $1/n$ cancels out, so you could just write this as a ratio of the sums of squared deviations. Also, in the least-squares procedure discussed in this chapter, it turns out that the mean of the residuals will be exactly zero.



We see immediately that students with higher homework averages also tend to have higher quiz averages. And in fact, a straight line would summarize the relationship between these variables fairly well. In fact, let's see the scatterplot again, this time with the least-squares line drawn in.

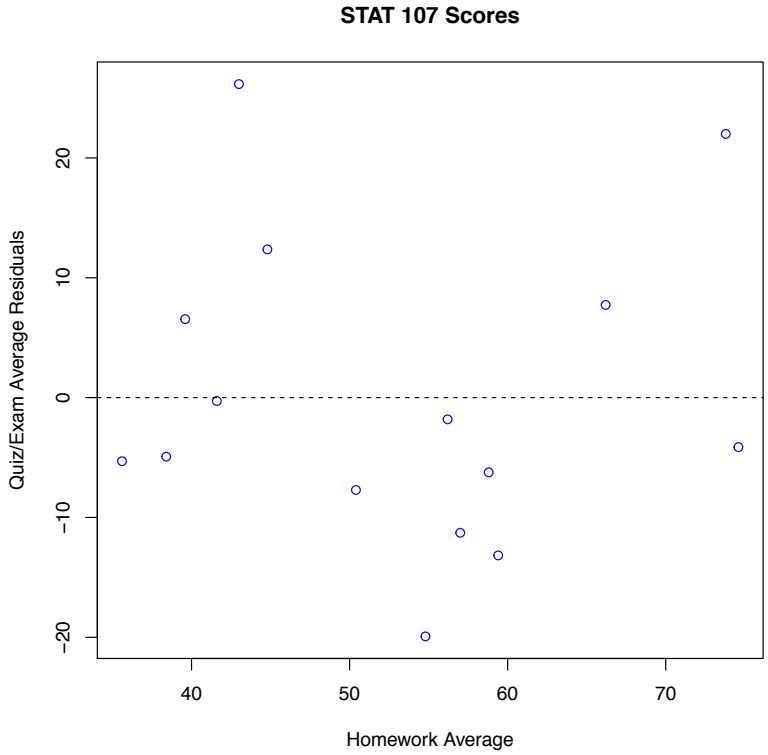
```
plot(x$HW.avg, x$Quiz.avg, col=4, xlab="Homework Average",
     ylab="Quiz/Exam Average", main="STAT 107 Scores")
fit <- lm(x$Quiz.avg ~ x$HW.avg)
abline(fit$coefficients, col=3)
```



Next, let's look at a *residual plot* where we do a scatterplot with the residuals in place of the y-values.[22]

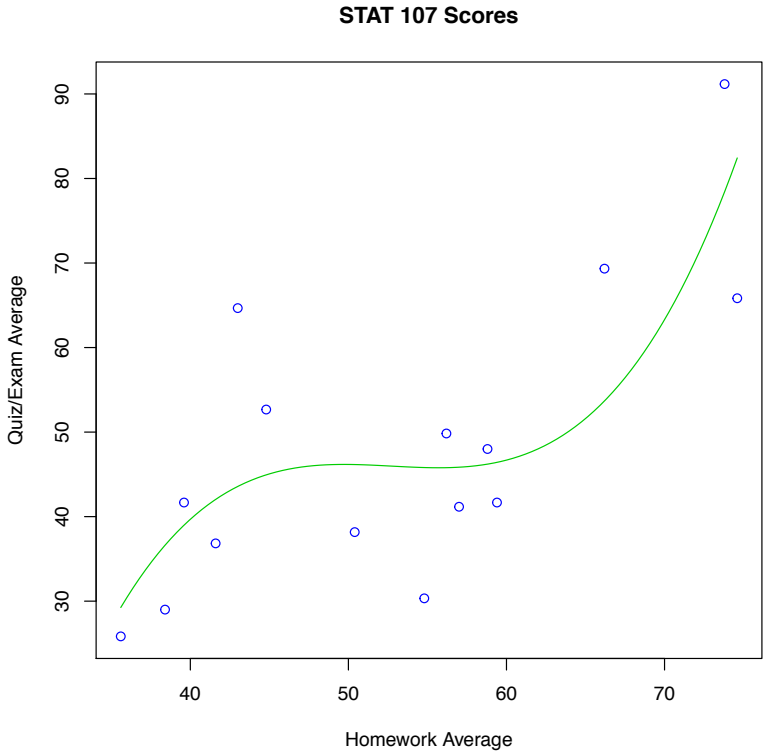
[22] For many authors, a "residual plot" puts the predicted values (the $\hat{y}$-values) on the horizontal axis. I prefer to put the original explanatory variables on the horizontal axis instead. However, when there are multiple explanatory variables (as we'll see in the upcoming section), it probably makes more sense to use the predicted value on the horizontal axis.

```
plot(x$HW.avg, fit$residuals, col=4, xlab="Homework Average",
     ylab="Quiz/Exam Average Residuals", main="STAT 107 Scores")
abline(h=0, lty=2)
```



Residual plots help you determine if there are any interesting patterns remaining in your data after you have done a fit. If the data looks like random fluctuations from zero, then there is no more "pattern" left to squeeze out of the data. In our case, there does seem to be an up-down-up pattern remaining. It looks a bit like some sort of cubic curve, a curve with an equation of the form $y = ax^3 + bx^2 + cx + d$ for some (a, b, c, d). In fact, we can find the least-squares cubic curve with the exact same `lm` command. Let's go back and try a cubic fit instead.

```
fit2 <- lm(x$Quiz.avg ~ x$HW.avg + I(x$HW.avg^2) + I(x$HW.avg^3))
fit2$coefficients

##   (Intercept)      x$HW.avg I(x$HW.avg^2) I(x$HW.avg^3)
## -4.875135e+02  3.056349e+01 -5.817587e-01  3.679261e-03

grid <- seq(min(x$HW.avg), max(x$HW.avg), length.out=100)
curve <- fit2$coefficients[1] + fit2$coefficients[2]*grid +
  fit2$coefficients[3]*grid^2 + fit2$coefficients[4]*grid^3
plot(x$HW.avg, x$Quiz.avg, col=4, xlab="Homework Average",
     ylab="Quiz/Exam Average", main="STAT 107 Scores")
lines(grid, curve, col=3)
```



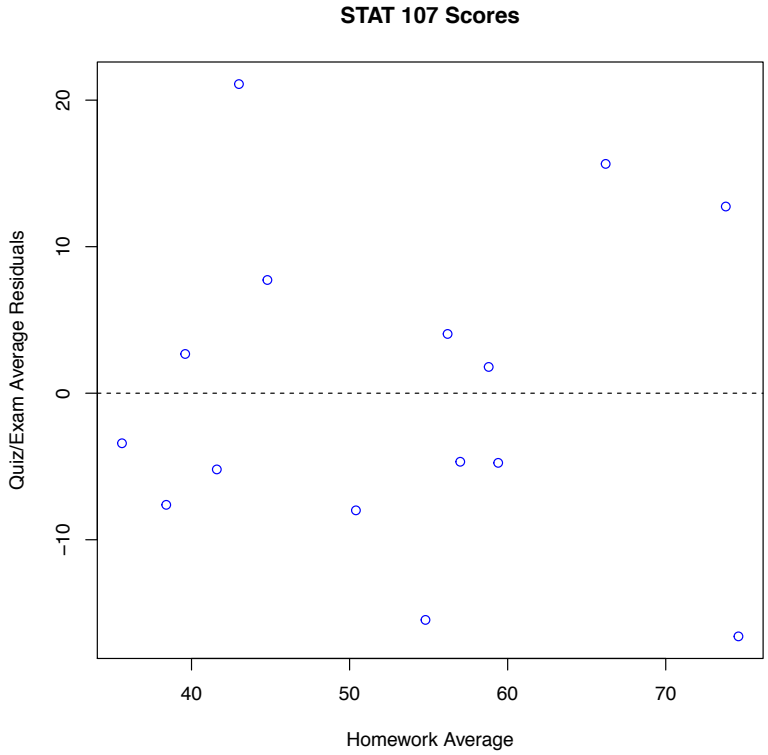
This cubic curve follows the pattern of the data better. Rounding each coefficient to two significant digits, the equation for the curve summarizing the relationship is (letting y represent quiz/exam average and x represent homework average)

$$y = -490 + 31x - 0.58x^2 + 0.0037x^3$$

The resulting residual plot that looks a lot more like random noise.[23]

[23] Another option when the relationship isn't linear is to transform one or both of the variables (e.g. take the natural logarithm of all the y-values) to get a plot in which the points do seem to have a more linear relationship. If you can find a simple transformation resulting in a linear relatinship, then you can calculate the ordinary least-squares line for relating the transformed variables.

```
plot(x$HW.avg, fit2$residuals, col=4, xlab="Homework Average",
     ylab="Quiz/Exam Average Residuals", main="STAT 107 Scores")
abline(h=0, lty=2)
```

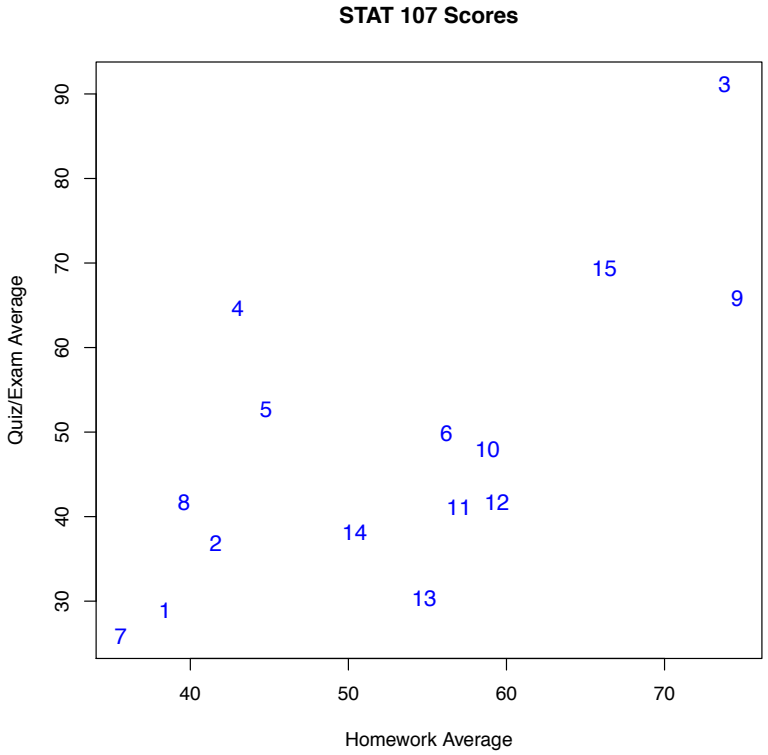


Be careful here! The point is to summarize the data, not to draw a best-fitting curve you can think of. Otherwise, we could have just drawn a complicated curve that wiggles around and goes through all the data points! But that wouldn't *summarize* the data well, because it isn't simplifying anything![24]

Most of the analysis above was only done to demonstrate the process; it isn't actually relevant to my goal. Recall that my purpose is to assign grades in a way that I think is sensible. Let's redraw the original scatterplot, this time drawing the observation numbers in place of dots so that we can identify the points.

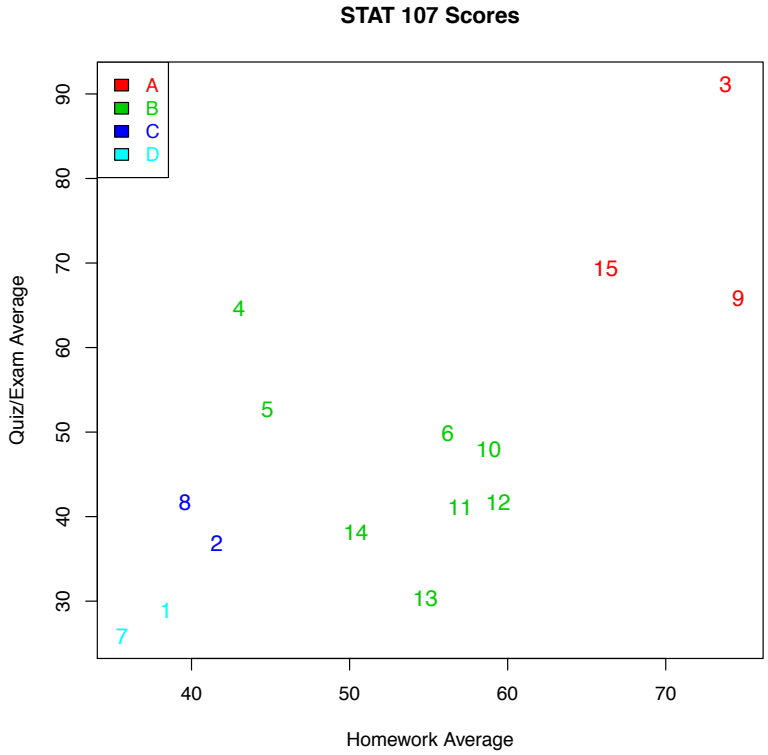
[24] This trade-off between fit and simplicity is considered by some to be one of the central unifying concepts of statistics.

```
# Scatterplot of students' averages with labels
plot(x$HW.avg, x$Quiz.avg, type="n", xlab="Homework Average",
     ylab="Quiz/Exam Average", main="STAT 107 Scores")
text(x$HW.avg, x$Quiz.avg, labels=rownames(x), col=4, cex=1.2)
```



Points in the top right are the best; points in the bottom left are the worst. Any pair of students who did about the same (points close to each other) should get the same grade. Let's come up with sensible groupings based on that idea. The three points in the upper right seem to stand out from the rest; those will be the A's. The rest of my grade assignments can be seen in the code below.

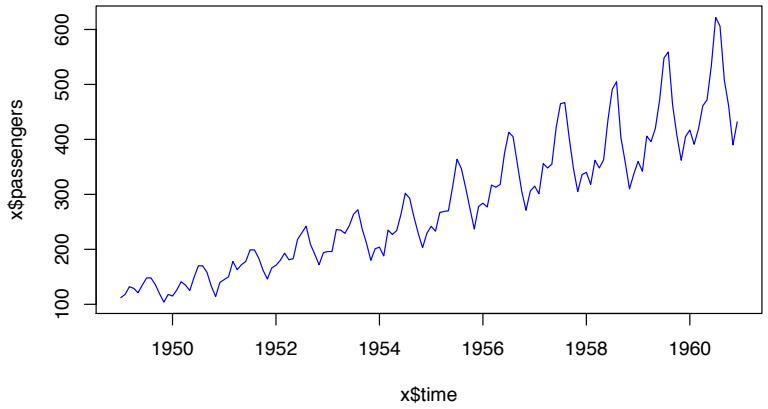
```
# Assign grades based on sensible grouping
A <- c(3, 9, 15)
B <- c(4, 5, 6, 10, 11, 12, 13, 14)
C <- c(2, 8)
D <- c(1, 7)

plot(x$HW.avg, x$Quiz.avg, type="n", xlab="Homework Average",
     ylab="Quiz/Exam Average", main="STAT 107 Scores")
text(x$HW.avg[A], x$Quiz.avg[A], labels=rownames(x)[A], col=2, cex=1.2)
text(x$HW.avg[B], x$Quiz.avg[B], labels=rownames(x)[B], col=3, cex=1.2)
text(x$HW.avg[C], x$Quiz.avg[C], labels=rownames(x)[C], col=4, cex=1.2)
text(x$HW.avg[D], x$Quiz.avg[D], labels=rownames(x)[D], col=5, cex=1.2)
legend("topleft", legend=c("A", "B", "C", "D"),
       text.col=2:5, fill=2:5)
```



Job done.

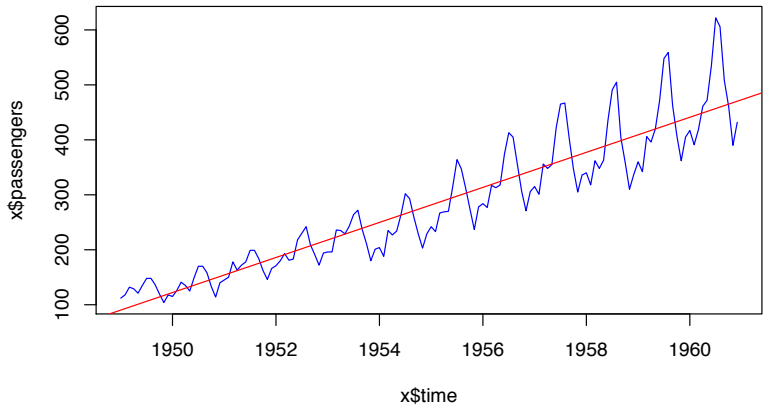
Example 5.2.1. Often you want to draw a scatterplot with consecutive data points connected by line segments, especially when the explanatory variable is time. For example, the plot below shows the number of airline passengers each month from the beginning of 1949 to the end of 1960.

```
help(AirPassengers)
head(AirPassengers)

## [1] 112 118 132 129 121 135

t <- seq(1949, 1960+11/12, by=1/12)
x <- data.frame(time=t, passengers=AirPassengers)
plot(x$time, x$passengers, type="l", col=4)
```



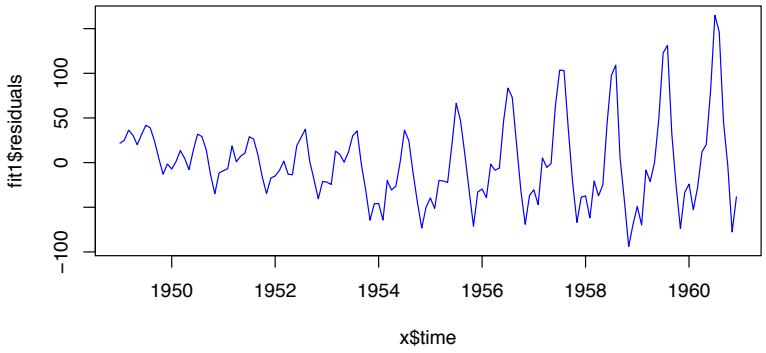
Clearly there is a certain regularity to the picture. The number of passengers is generally increasing, but it also seems to have a periodic pattern by month. Let's try to capture the increasing trend first. The least-squares line looks pretty good.

```
plot(x$time, x$passengers, type="l", col=4)
fit1 <- lm(passengers ~ time, data=x)
abline(fit1, col=2)
```



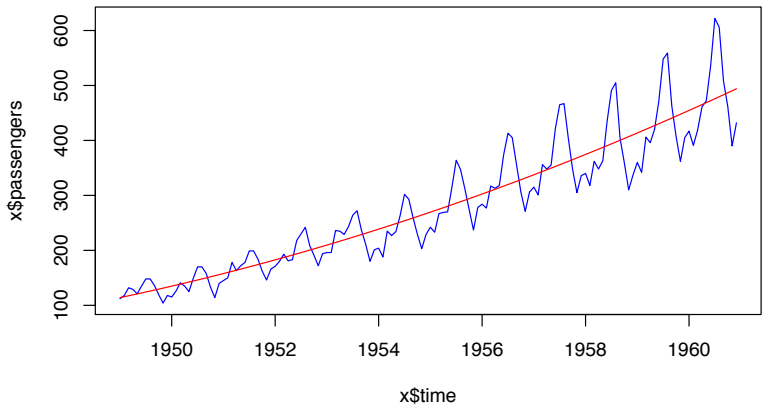
But when we look at the residuals, there's a down-up curvature left over.

```
plot(x$time, fit1$residuals, type="l", col=4)
```



A quadratic fit, however, seems to capture the trend nicely.

```
plot(x$time, x$passengers, type="l", col=4)
fit1 <- lm(passengers ~ time + I(time^2), data=x)
grid <- seq(min(x$time), max(x$time), length.out=100)
lines(grid, fit1$coef[1] + fit1$coef[2]*grid + fit1$coef[3]*grid^2, col=2)
```




```
plot(x$time, fit1$residuals, type="l", col=4)
```

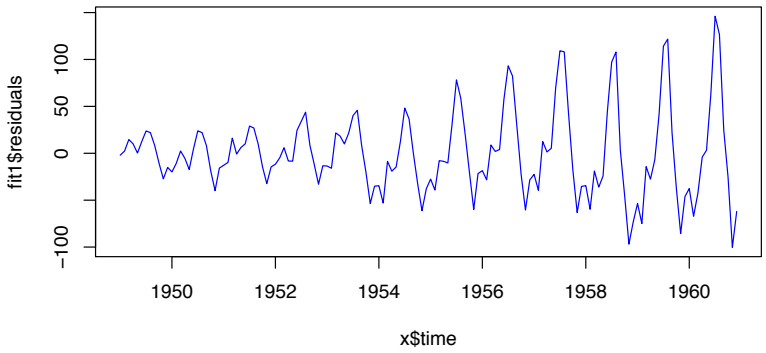
Next, we will pursue and *iterative fitting* strategy. Iterative fitting means repeatedly fitting the residuals.

$$
\begin{aligned}
y &= \text{fit}_1 + \text{residuals}_1 \\
\text{residuals}_1 &= \text{fit}_2 + \text{residuals}_2 \\
&\vdots \\
\text{residuals}_{k-1} &= \text{fit}_k + \text{residuals}_k
\end{aligned}
$$

Putting the equations together, the overall result of iterative fitting is

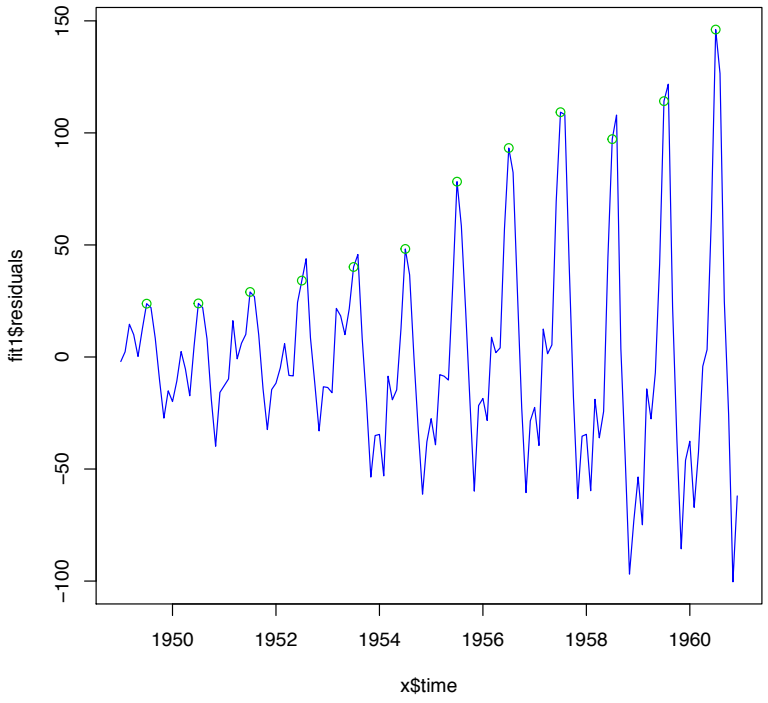
$$
y = \underbrace{\sum_{i=1}^{k} \text{fit}_i}_{\text{overall fit}} + \underbrace{\text{residuals}_k}_{\text{overall residuals}}
$$

Figure 5.7: Residuals from the least-squares quadratic fit.



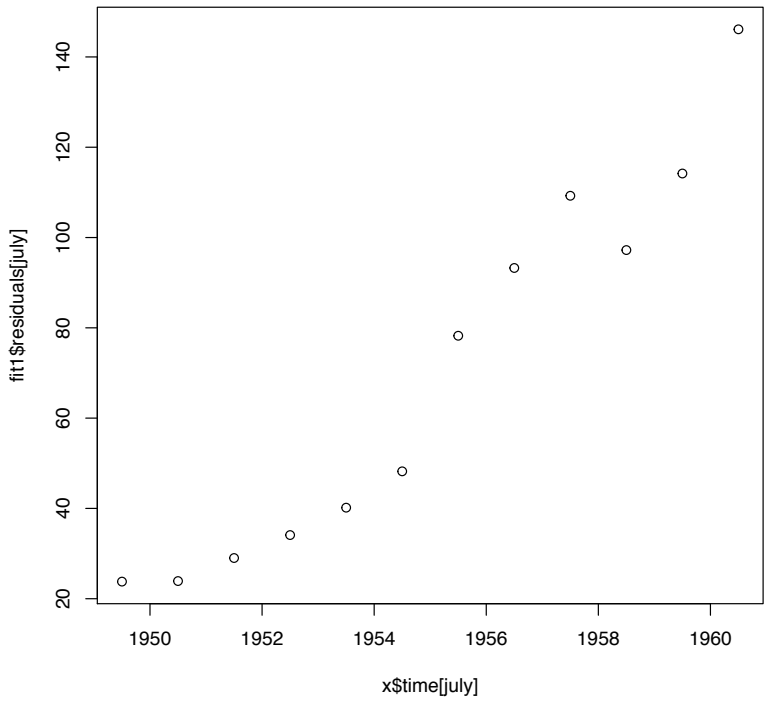
So we want to fit the residuals shown in Figure 5.7. There is clearly periodic behavior, but the scale of these fluctuations is also increasing over time. Let's try to figure out the nature of the scale increase first. After a bit of experimentation, you will find that the peaks are usually happening in the month of July.

```
july <- seq(7, 144, by=12)
plot(x$time, fit1$residuals, type="l", col=4)
points(x$time[july], fit1$residuals[july], col=3)
```



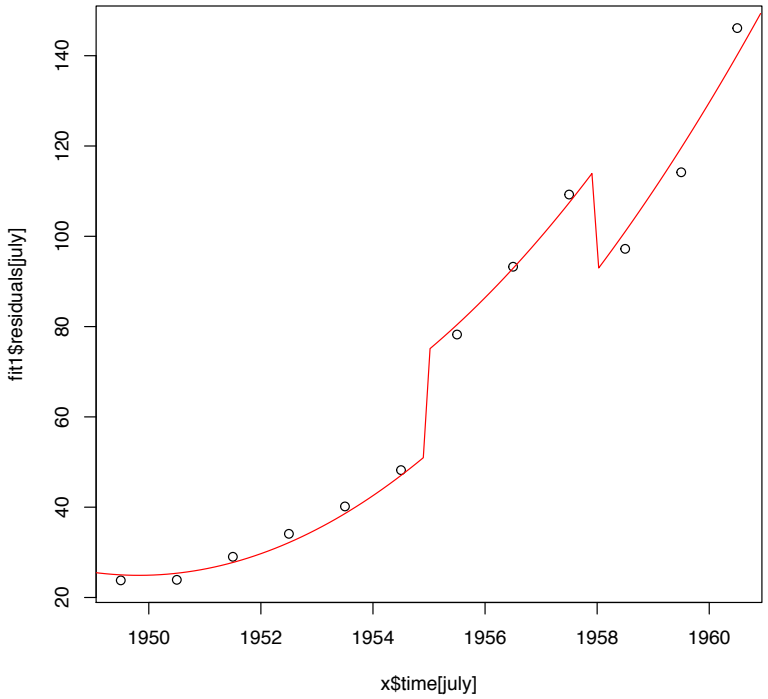
In fact, let's just look at the July points to see if we can fit their increase over time, hoping that it is representative of the overall scale of fluctuations.

```
plot(x$time[july], fit1$residuals[july])
```



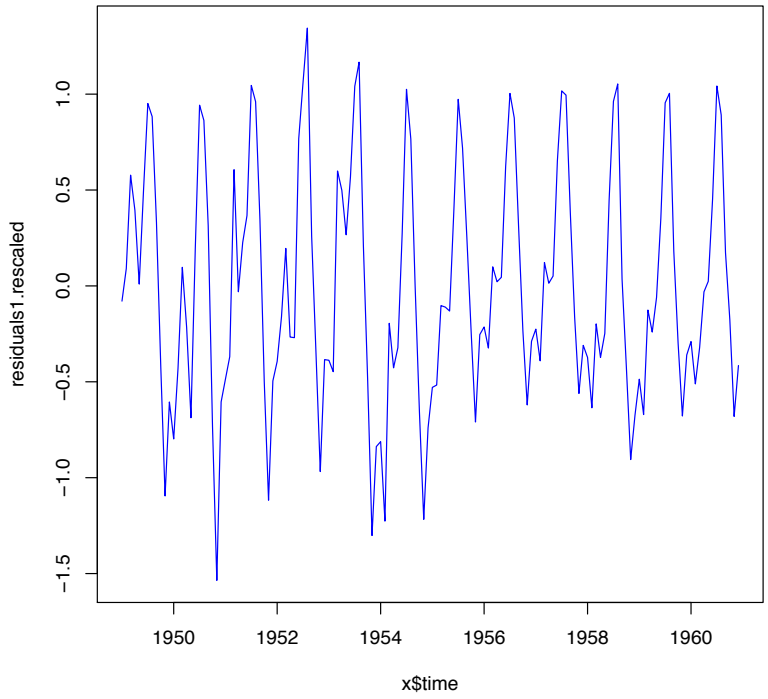
A quadratic curve seems appropriate here, except for the three points that appear to be shifted upward. We can actually fit a quadratic curve along with a an extra constant parameter to accomodate those shifted data points.

```
fit.july <- lm(fit1$residuals[july] ~ time[july] + I(time[july]^2) + I(time[july] > 1955 & time[july] <
grid <- seq(min(x$time), max(x$time), length.out=100)
plot(x$time[july], fit1$residuals[july])
lines(grid, fit.july$coef[1] + fit.july$coef[2]*grid
      + fit.july$coef[3]*grid^2 + fit.july$coef[4]*(grid > 1955 & grid < 1958), col=2)
```



Now we will use this curve to rescale all of the residuals.

```
rescale <- fit.july$coef[1] + fit.july$coef[2]*x$time +
           fit.july$coef[3]*x$time^2 +
           fit.july$coef[4]*(x$time > 1955 & x$time < 1958)
residuals1.rescaled <- fit1$residuals/rescale
plot(x$time, residuals1.rescaled, type="l", col=4)
```

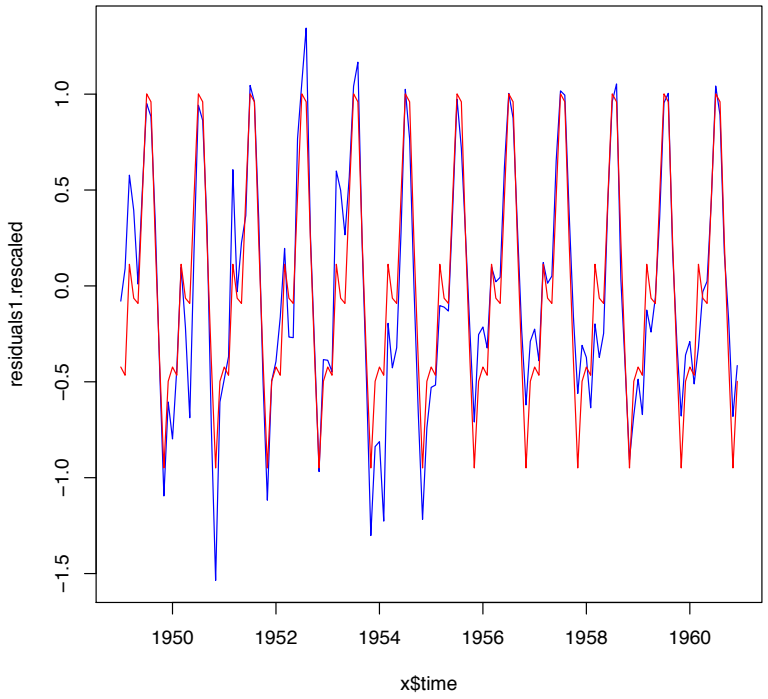


That worked well; the periodic fluctuations in these rescaled residuals don't seem to be getting larger or smaller over time.[25] At last, we can try to fit the periodic behavior. It's not really smooth enough to be a simple sine curve.[26] A simple thing we can do is just fit a prediction for each month; the obvious choice is to just take the means. For instance, we will predict all January points with the mean of the January points, and so on.

[25] The fluctuations are a little smaller in the years that we allowed a shift parameter, but that's okay; it's not much smaller.

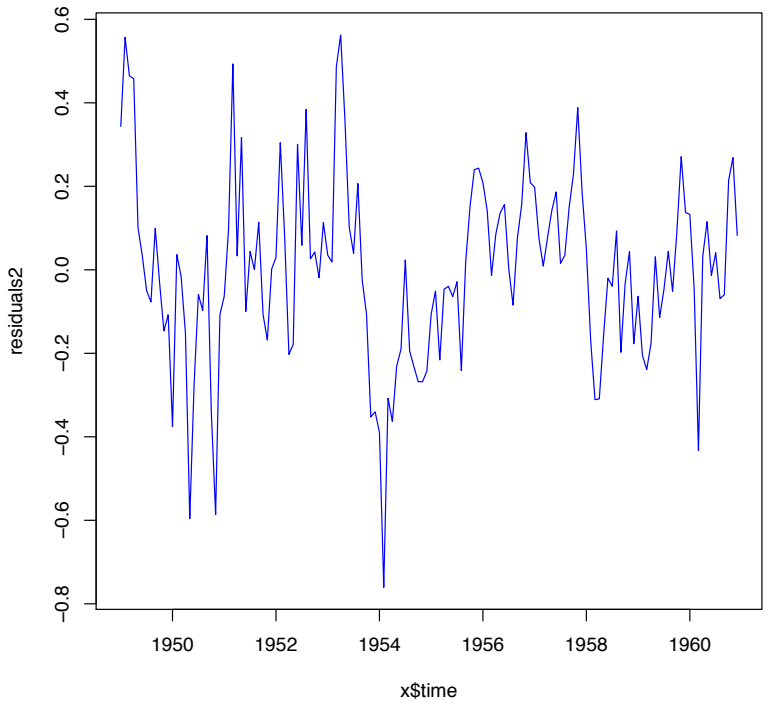
[26] Maybe a superposition of a few sine curves would work well, but that seems a little too advanced for this book.

```
d <- matrix(residuals1.rescaled, ncol=12, byrow=TRUE)
means <- apply(d, 2, mean)
plot(x$time, residuals1.rescaled, type="l", col=4)
lines(x$time, rep(means, 12), col=2)
```



Looks good. Now let's look at the next set of residuals.

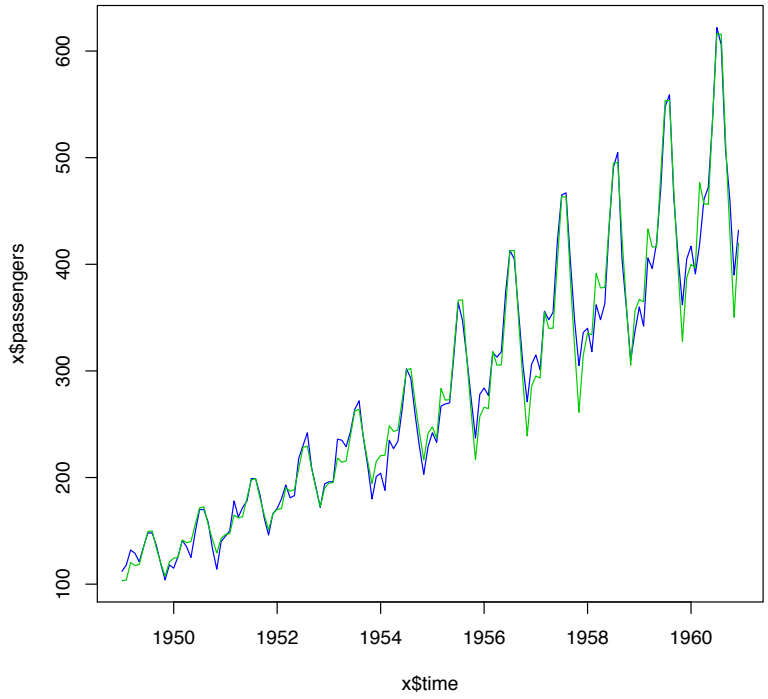
```
residuals2 <- residuals1.rescaled - rep(means, 12)
plot(x$time, residuals2, type="l", col=4)
```



There's not much pattern left to squeeze out of the data, as far as I'm

concened. That means we're done fitting. Now, let's piece together our overall fit and see how it looks in comparison to the original data.

```
fit <- fit1$coef[1] + fit1$coef[2]*x$time + fit1$coef[3]*x$time^2 +
       rescale*rep(means, 12)
plot(x$time, x$passengers, type="l", col=4)
lines(x$time, fit, col=3)
```



Beautiful! What's the R^2?

```
1 - var(residuals2)/var(x$passengers)

## [1] 0.9999966
```

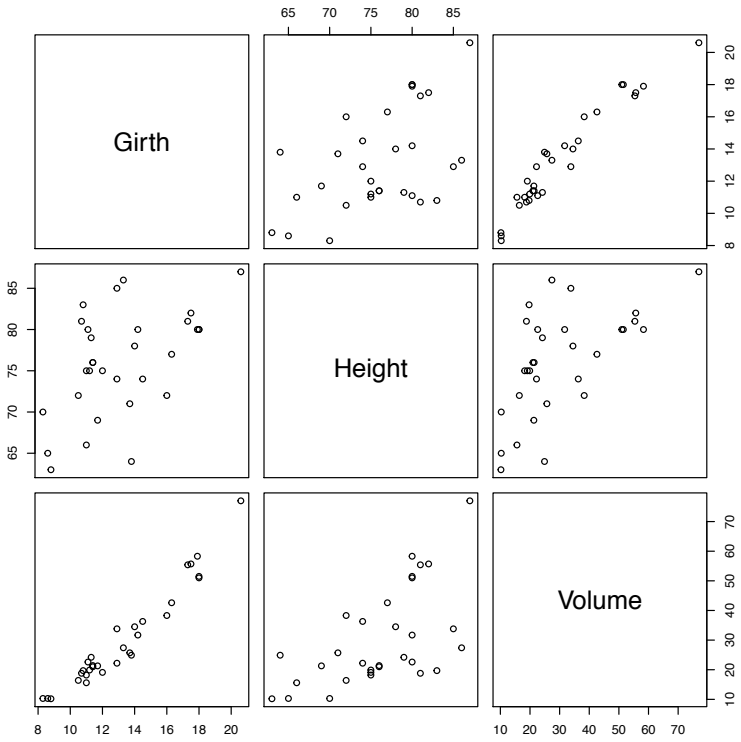
First, we used three parameters in our quadratic fit of the overall trend. Next, we used sixteen parameters on the periodic fit: four parameters for its change in scale over time and another twelve parameters for the periodic behavior. In total, we used nineteen parameters to construct a fit that reduced the variability in the 144 data points by about 99.9997%. While we've fit the data well, we haven't actually explained why the number of airline passengers has behaved the way it has over this time period. Someone with more knowledge of the situation might offer a hypothesis that comports with the data. We also haven't made any claims that the number of airline passengers will continue tracking along our fit curve. If you had to guess the number of airline passengers over the next couple years after the data ends, this curve offers a good starting point. But you never re-

ally know what the future will hold, especially once you start looking five or ten years ahead.

Finally, when your data set has more than two quantitative variables, you may want to make a scatterplot for every pair of them. A *pairs plot* arranges all of these scatterplots into a matrix shape.[27]

[27] If you have too many variables, R's `pairs` plots will be too small to read. In that case, you might want to just run `pairs` on selected subsets of your variables at a time.

```
help(trees)
head(trees)

##   Girth Height Volume
## 1   8.3     70   10.3
## 2   8.6     65   10.3
## 3   8.8     63   10.2
## 4  10.5     72   16.4
## 5  10.7     81   18.8
## 6  10.8     83   19.7

pairs(trees)
```



For each scatterplot, the variable plotted on the horizontal axis is the one whose name appears in the same column as the plot; the variable plotted on the vertical axis is the one whose name appears in the same row as the plot.

5.3 Three Quantitative Variables

The trees dataset that you just saw has three quantitative variables. When dealing with more than two quantitative variables, you can

look at each pair as was done in the pairs plot. Another example of this is the *correlation matrix*, which displays the correlations between each pair of variables.

```
cor(trees)

##          Girth  Height  Volume
## Girth  1.00000 0.51928 0.96712
## Height 0.51928 1.00000 0.59825
## Volume 0.96712 0.59825 1.00000
```

But that's really nothing new because each correlation only involves two variables. We'll now explore ways of analyzing all three of these variables together.

5.3.1 *Statistics*

When we had two variables, we imagined each observation is a point in the two-dimensional plane. Now that we have three variables, we can instead imagine them as points in a three-dimensional plane. And instead of finding a least-squares line that predicts one response variable as a function of one explanatory variable, we can find the *least-squares plane* that predicts one response variable as a funcion of two explanatory variables.[28] We will denote the two explanatory variables' axes as x_1 and x_2. The least-squares plane will correspond to an equation of the form $y = a + bx_1 + cx_2$. We just need to find the (a, b, c) that minimizes the sum of squared residuals. Again, the `lm` command does just what we're looking for.[29]

In the `trees` dataset, it seems most natural to think of the volume of timber in the tree as being a mostly result of its girth and height. Therefore, we will treat `Volume` as the response variable, with `Girth` and `Height` as the explanatory variables.

```
fit <- lm(Volume ~ Girth + Height, data=trees)
fit$coefficients

## (Intercept)       Girth      Height
##   -57.98766     4.70816     0.33925
```

To two significant digits, the least-squares plane summarizing the relationship is

$$\text{Volume} = -58 + 4.7 * \text{Girth} + 0.34 * \text{Height}$$

Again, we can calculate the R^2 of this least-squares plane fit, telling us what proportion of the variation in Volume is explained by `Girth` and `Height` together.

```
1 - var(fit$residuals)/var(trees$Volume)

## [1] 0.94795
```

[28] This idea generalizes to any number of variables. You can choose one to be the response variable, and construct a least-squares fit as a function of the rest. In fact, "least-squares plane" is not a common term becuase fitting with more than one explanatory variable is usually just called "multiple regression."

[29] This bit of code demonstrates a slightly different way of telling R which variables you want to use. Remember, R usually has lots of different ways of doing the same task.

5.3.2 *Plots*

Your first thought may be to extend the scatterplot idea by adding an extra dimension to it. In fact, sometimes people do use these "3-D scatterplots," but they're not as useful as you might think. When you try to draw the 3-D picture on a 2-D plane, it's difficult to convey the depth that each data point is supposed to have.[30] Instead, I prefer to plot the two explanatory variables on a two-dimensional plane and then use a color gradient to indicate the value of the response variable. We will call such a picture a *color gradient scatterplot*. I've uploaded a bit of R code that includes a function called `CGSplot` to make color gradient scatterplots. Running the following code will read the function into your R environment.[31]

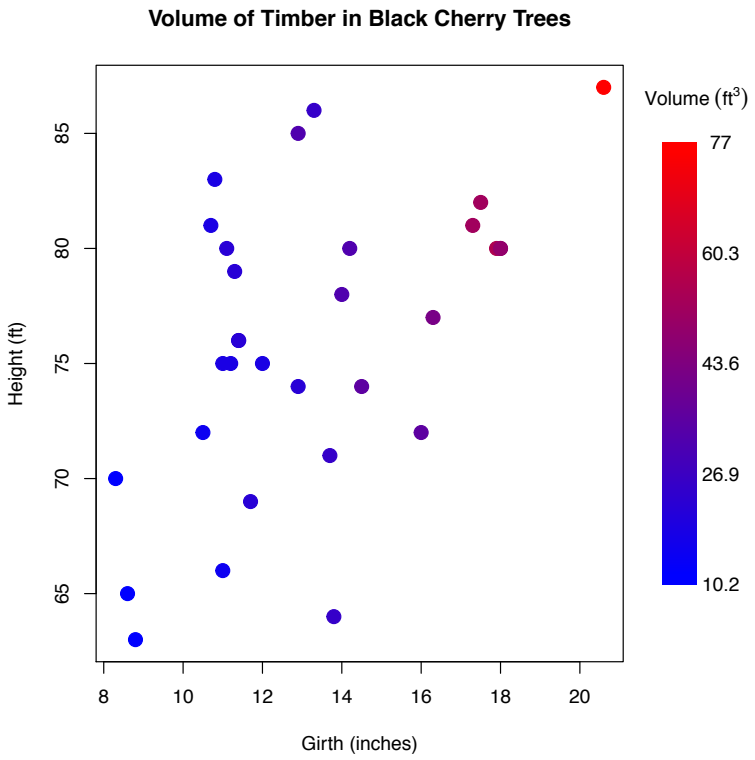
[30] There are some clever bits of software (such as the R package `rgl`) that let you rotate the image around to see the positions of the points well. On the downside, you still can't print out these plots easily if you need to prepare a report.

[31] Type the url below into your browser if you want to see the code. If you ever have a problem running this function on your data, you can always download the code and try modifying it to suit your needs.

```
source("http://www.stat.yale.edu/~wdb22/CGSplot.R")
```

Now, let's see the function in action on our `trees` data.

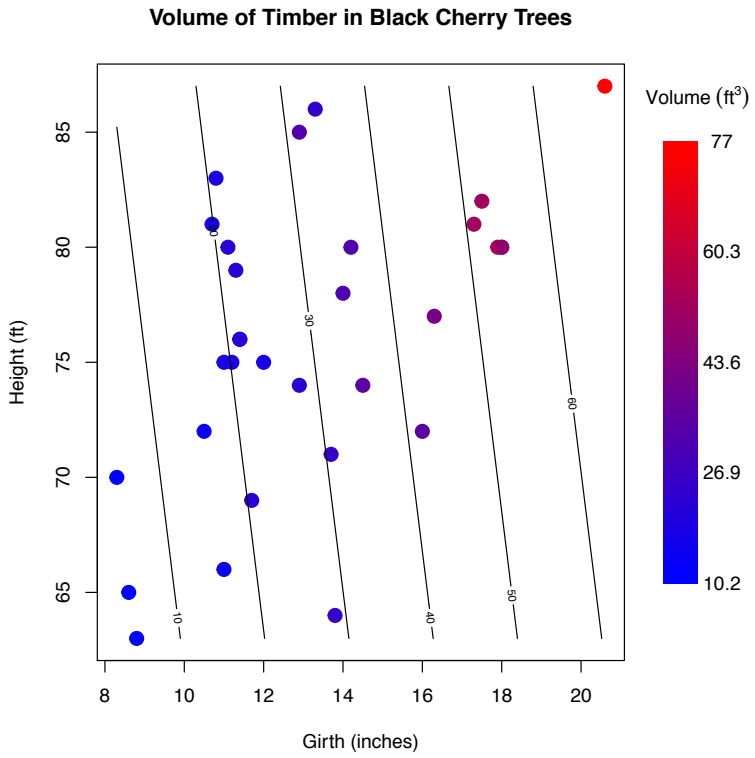
```
d <- trees
CGSplot(d$Girth, d$Height, d$Volume, pch=19, cex=1.6,
        xlab="Girth (inches)", ylab="Height (ft)",
        zlab=expression(Volume~(ft^3)),
        main="Volume of Timber in Black Cherry Trees")
```



It was easy to overlay the least-squares line on top of the scatterplot in Section 5.2.2 It's not as easy to add our least-squares plane to the plot. But I've written some code below that will add the fit's *level curves* to the color gradient scatterplot.[32]

[32] This code is probably going to be challenging for you to follow, if you're a beginner. I recommend not paying too much attention to it on your first pass through the book.

```
fcontour <- function(x, y, f, size=50, ...) {
  xgrid <- seq(min(x), max(x), length.out=size)
  ygrid <- seq(min(y), max(y), length.out=size)
  z <- matrix(NA, size, size)
  for(i in 1:size) {
    for(j in 1:size) {
      z[i, j] <- f(xgrid[i], ygrid[j], ...)
    }
  }
  contour(xgrid, ygrid, z, nlevels=6, add=TRUE)
}

plane <- function(x1, x2, abc) {
  return(abc[1] + abc[2]*x1 + abc[3]*x2)
}

CGSplot(d$Girth, d$Height, d$Volume, pch=19, cex=1.6,
        xlab="Girth (inches)", ylab="Height (ft)",
        zlab=expression(Volume~(ft^3)),
        main="Volume of Timber in Black Cherry Trees")
fcontour(d$Girth, d$Height, plane, abc=fit$coefficients)
```



Each curve tells you what `Volume` the least-squares plane predicts at those points.

Although fitting a linear function of the explanatory variables is often fruitful, you may sometimes be able to use prior knowledge of the system to guess a type of function that works better. In this

case, we're trying to predict the volume of a tree's usable timber. Volume should be approximately height times area; area should be proportional to girth squared.[33] So let's try to fit an equation of the form

[33] This is because area is proportional to the radius squared, and the radius is proportional to the girth (i. e. circumference).

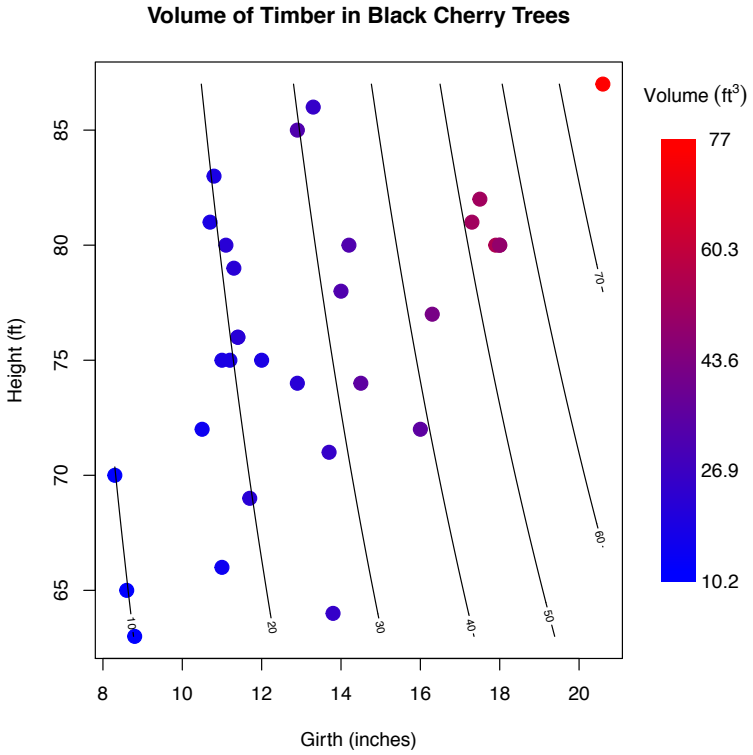
$$\text{Volume} = a + b * \text{Height} * \text{Girth}^2$$

using the `lm` command and compare it to our linear fit.

```
fit2 <- lm(Volume ~ I(Height*Girth^2), data=trees)
fit2$coefficients

##         (Intercept) I(Height * Girth^2)
##          -0.2976794           0.0021244

1 - var(fit2$residuals)/var(trees$Volume)

## [1] 0.97777
```

We've achieved a larger R^2 while using fewer parameters, which is almost always preferable and certainly so in our case due to the easy interpretation of this equation. Let's plot the level curves for this fit too.

```
volume.surface <- function(x1, x2, ab) {
  return(ab[1] + ab[2]*x1^2*x2)
}

CGSplot(d$Girth, d$Height, d$Volume, pch=19, cex=1.6,
        xlab="Girth (inches)", ylab="Height (ft)",
        zlab=expression(Volume~(ft^3)),
        main="Volume of Timber in Black Cherry Trees")
fcontour(d$Girth, d$Height, volume.surface, ab=fit2$coefficients)
```



5·4 *Conclusion*

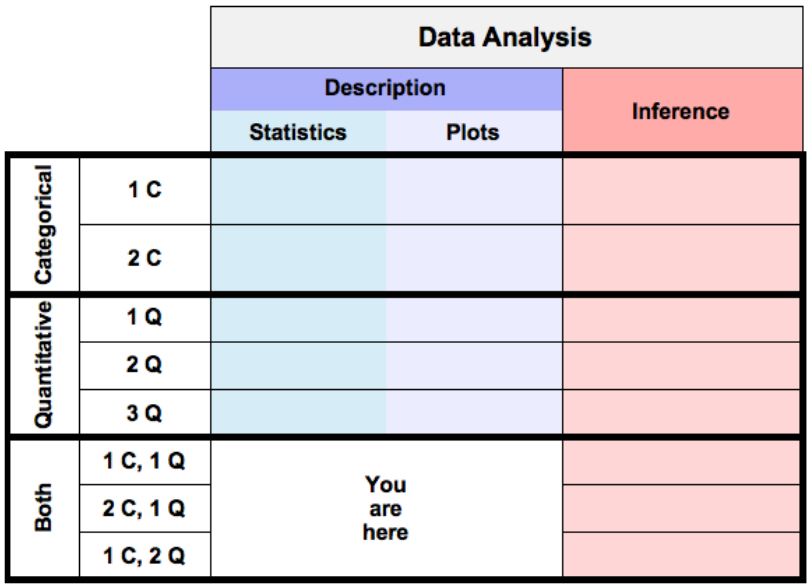
With one quantitative variable, we saw a number of statistics and plots for summarizing the location and spread of the data points. Once we had two or more quantitative variable, we became interested in summarizing the relationships among them. We repeatedly found ourselves looking for a function of the explanatory variables that would fit the response variable well, a task known as *regression*. In each case, we settled on a class of functions that was linear in its parameters, then used the `lm` command to find the parameter values of the "best-fitting" function in the class (the function with the smallest sum of squared residuals). In the two-variable case, the set of straight lines is the most commonly used class of functions, and the best-fitting one is called the least-squares line. In the three-variable case, the set of planes is the most commonly used class, and we called the best-fitting one the least-squares plane.

Again, we can compile what we've learned into a table as we work our way toward an understanding of the big picture (Figure 1.3).

	Description	
	Statistics	**Plots**
One Quantitative Variable	mean median quartiles standard deviation interquartile range	histogram density plot boxplot
Two Quantitative Variables	correlation least-squares line R^2 residuals	scatterplot residual plot
Three Quantitative Variables	least-squares plane R^2	color gradient scatterplot

6 Description of Both Types Together

Finally, we will look at a few ways to synthesize categorical and quantitative information to describe our data. The tools available to us (statistics and plots) depend on how many of each type of variable we are trying to analyze together. This chapter will go through three different mixes of variables: one categorical with one quantitative, two categorical with one quantitative, and one categorical with two quantitative.



6.1 One Categorical Variable and One Quantitative Variable

We now return to the dataset of computer files (Example 1.2.1) that we first saw in Chapter 1.

```
x <- read.csv("http://www.stat.yale.edu/~wdb22/Files.csv")
head(x)

##   type size
## 1  JPG 0.38
## 2  DOC 0.04
## 3  MP3 0.31
## 4  JPG 0.16
```

```
## 5  JPG 0.55
## 6  DOC 0.29
```

The data frame is made up of one categorical variable (`type`) and one quantitative variable (`size`), so we will use it to demonstrate some possible statistics and plots for this combination of variable types.

6.1.1 Statistics

Any statistic that you can calculate on a single quantitative variable, you can calculate as *aggregate statistics* by finding the quantity for each category separately. The statistic for each category can be organized into a *one-way table*, which generalizes the idea of a one-way frequency table. Now instead of the entries of the table being counts, they can be any function of a quantitative variable.

The point of calculating aggregate statistics is typically to compare different groups, and the mean is the most commonly used statistic here.

```
means <- aggregate(x$size, list(type=x$type), mean)
names(means)[2] <- "mean"
means

##   type     mean
## 1  DOC  0.20247
## 2  JPG  0.24856
## 3  MP3 10.34014
```

The average size of a `JPG` is a little bigger than the average size of a `DOC`, while the average `MP3` is many times larger than both of them.

As another example, you could find the size of the largest file of each type.

```
maxes <- aggregate(x$size, list(type=x$type), max)
names(maxes)[2] <- "max"
maxes

##   type   max
## 1  DOC  1.57
## 2  JPG  2.62
## 3  MP3 88.01
```

Or you might want to calculate a an aggregate statistic that doesn't have a built-in function in R. In that case, you can write your own function; to demonstrate, let's find the sum of the squared file sizes for each type.

```
sumsquare <- function(v) {
        return(sum(v^2))
}
```

```
sumsquares <- aggregate(x$size, list(type=x$type), sumsquare)
names(sumsquares)[2] <- "sum of squares"
sumsquares

##   type sum of squares
## 1  DOC         328.38
## 2  JPG         612.57
## 3  MP3      214625.43
```

In our discussion so far, we have implicitly thought of the categorical variable as explanatory and the quantitative variable as the response. There are *classification* techniques that reverse those roles, such as *logistic regression*, but they are beyond the scope of this book.

6.1.2 *Plots*

First, let's talk about how to display the results in a one-way table. An obvious choice is the *bar chart* that we saw when we wanted to display one-way frequency tables. There's really nothing about the bar chart that is specific to frequencies, so we can use it to display any aggregate statistics we want.

```
barplot(means$mean, names.arg=means$type, col=2:4,
        ylab="File Size (MB)", main="Average File Sizes")
```

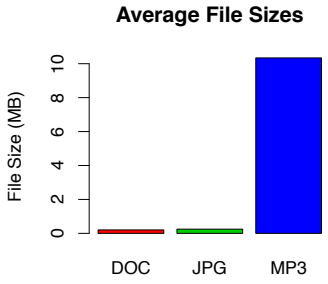
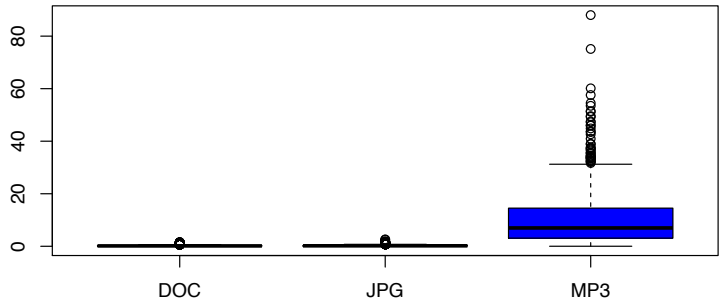


Figure 6.1: The bar chart shows the average file size for each filetype. We see that, on average, an MP3 is many times larger than any other filetype.

Now we know how to compare a single statistic from each category. What if you want to compare the *overall shape of the data* from the different categories? We'll look at two types of plots that are useful. First, *side-by-side boxplots* are, as you might guess from the name, separate boxplots of the data within each category placed alongside each other.

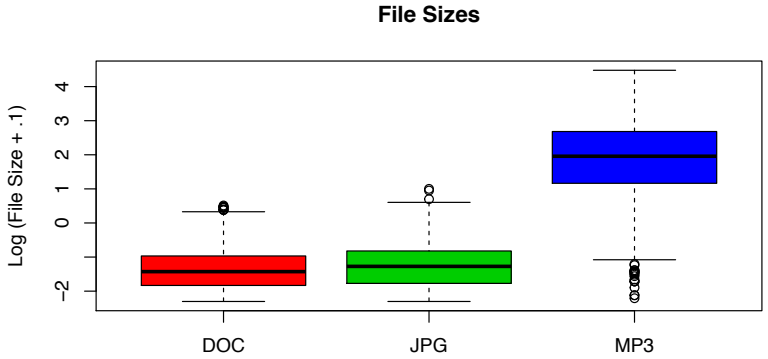
```
boxplot(size ~ type, data=x, col=2:4)
```



In this figure, the range of the MP3 files' sizes is so much larger than the others that it's impossible to see any of the detail of the other two boxplots. Let's try to transform the data to get them on more equal footing. Below the side-by-side boxplots are drawn again, after adding .1 to each file size then taking the natural logarithm of it.[1]

[1] Taking logarithms of very small values makes them shoot off toward negative infinity, so often we add a small number to each data point first to avoid that problem. You may wonder whether that's "allowed." It's a *monotonic* transformation, just like the logarithm transformation is. Any monotonic transformation is

```
boxplot(log(size+.1) ~ type, data=x, col=2:4,
        ylab="Log (File Size + .1)",
                main="File Sizes")
```



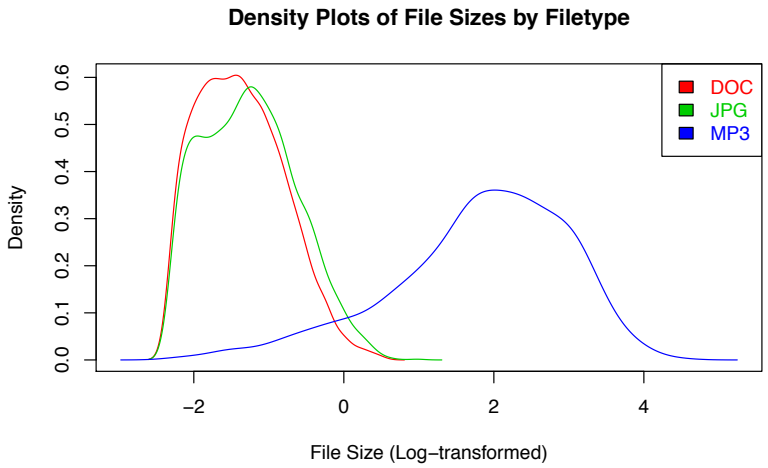
Notice that this shows us much more information about the file sizes than the bar chart of means (Figure 6.1) showed us. We can see right away that, for instance, even the largest `DOC` or `JPG` is still smaller than three quarters of the `MP3` files.[2]

[2] If you don't see why that's clear, review the definition of boxplots in Section 5.1.1.

Another way to visualize the whole shape of the data among the various groups is to create a *colored density plot* which superimposes each category's density plot on the same graph in different colors. My code for making Figure 6.2 is stored on my server, but you can `source` it into your R environment.

```
source("http://www.stat.yale.edu/~wdb22/CDplot.R")
CDplot(log(x$size+.1), x$type, xlab="File Size (Log-transformed)",
       main="Density Plots of File Sizes by Filetype")
```

Figure 6.2: Colored Density Plots show that the `DOC` and `JPG` files have very a very distribution of sizes, while the `MP3` files tend to be much larger.



The colored density plot conveys similar information to the side-by-side boxplot.

6.2 Two Categorical Variables and One Quantitative Variable

Recall the exercise and smoking survey data that we explored in Chapter 4. We'll take another look at that data set, this time including the quantitative variable `pulse`.

```
library(MASS)
y <- survey[, c("Exer", "Smoke", "Pulse")]
head(y)

##   Exer Smoke Pulse
## 1 Some Never    92
## 2 None Regul   104
## 3 None Occas    87
## 4 None Never    NA
## 5 Some Never    35
## 6 Some Never    64

# Simplify the categories, as before
levels(y$Smoke) <- c("Smokes", "Never smokes", "Smokes", "Smokes")
y$Exer <- factor(y$Exer, levels(y$Exer)[c(2, 3, 1)])
y$Smoke <- factor(y$Smoke, levels(y$Smoke)[2:1])
```

Notice the `NA` in the `pulse` variable. That is an example of *missing data*, and it's to be expected in real world data analysis. A common method for handling missing data is to just ignore the observations that have `NA`s, which can be done as follows.

```
# How many observations did we start with?
dim(y)

## [1] 237   3

# Remove any observations with missing values
y <- y[complete.cases(y), ]
# How many observations do we have left?
dim(y)

## [1] 191   3
```

6.2.1 Statistics

This case is very similar to the one we just saw. Now that we have two categorical variables, however, we might want to calculate an *aggregate statistic over every combination of categories*. The results can be organized into a *two-way table*.

```
mean.pulse <- tapply(y$Pulse, list(y$Exer, y$Smoke), mean)
mean.pulse

##      Never smokes Smokes
```

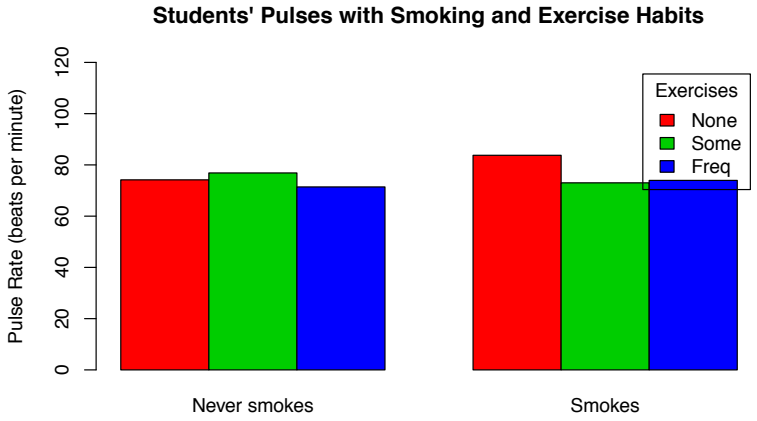
```
## None       74.167 83.750
## Some       76.864 73.000
## Freq       71.405 73.952
```

The two-way table is a generalization of the two-way frequency table in the same way that the one-way table is a generalization of the one-way frequency table. Again, we're implicity treating the quantitative variable as the response variable; we will not cover the *classification* techniques that treat the categorical variable as the response.

6.2.2 *Plots*

Just as the *multiple bar chart* was good for displaying the two-way frequency table, it's also effective at displaying the information from a two-way table in general.

```
barplot(mean.pulse, beside=TRUE, legend=rownames(mean.pulse), col=2:4,
        main="Students' Pulses with Smoking and Exercise Habits",
        ylab="Pulse Rate (beats per minute)", ylim=c(0, 120),
        args.legend=list(title="Exercises"))
```

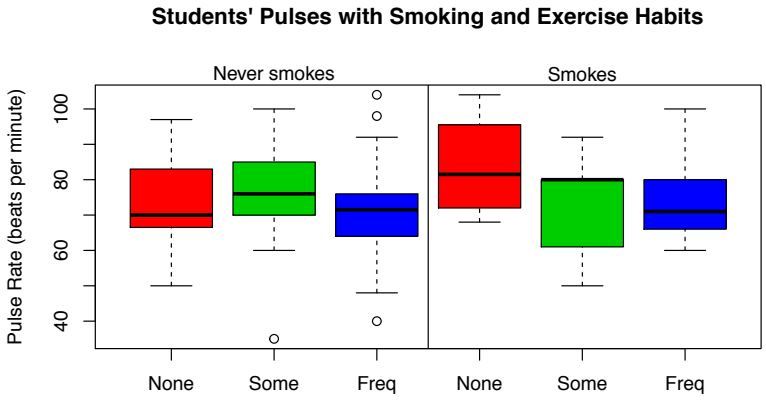


We see that among non-smokers the differences in average pulse based on exercise frequency weren't very large. However, among the smokers, those who never exercise had an average pulse about ten beats per minute higher than those who did exercise. This is another case where the roles of the two categorical variables can be reversed if it leads to a more intuitive interpretation. In practice, you might want to try it both ways; recall from Section 4.2 that the `t` command returns the transpose of a table.

To see more detail about the shape of the quantitative variable's data within each combination of categories, we can draw *multiple side-by-side boxplots*.

```
par(mar=c(5.1, 4.1, 6, 2.1))
boxplot(Pulse ~ Exer + Smoke, data=y, col=2:4,
        names=rep(levels(y$Exer), length(levels(y$Smoke))),
```

```
        main="Students' Pulses with Smoking and Exercise Habits",
        ylab="Pulse Rate (beats per minute)")
abline(v=3.5)
par(xpd=TRUE)
text(x=2, y=110, labels=levels(y$Smoke)[1])
text(x=5, y=110, labels=levels(y$Smoke)[2])
par(mar=c(5.1, 4.1, 4.1, 2.1), xpd=FALSE)
```



6.3 *One Categorical Variable and Two Quantitative Variables*

At last, we've reached the last mix of variable types we will discuss: one categorical and two quantitative variables. Again, we won't go into the classification techniques that treat the categorical variable as the response. Instead we will follow the same pattern that we've followed throughout this chapter by splitting up the data according to the value of the categorical variable, and looking at the quantitative variables within each group.

We will use a dataset of 74 models of automobiles sold in the United States in 1979. Our categorical variable will be the model's origin (America, Europe, or Japan); the two quantitative variables will be the model's weight and gas mileage.

```
library(corrgram)
help(auto)
head(auto)

##               Model Origin Price MPG Rep78 Rep77 Hroom Rseat Trunk Weight
## 1 AMC Concord             A  4099  22     3     2   2.5  27.5    11   2930
## 2 AMC Pacer               A  4749  17     3     1   3.0  25.5    11   3350
## 3 AMC Spirit              A  3799  22    NA    NA   3.0  18.5    12   2640
## 4 Audi 5000               E  9690  17     5     2   3.0  27.0    15   2830
## 5 Audi Fox                E  6295  23     3     3   2.5  28.0    11   2070
## 6 BMW 320I                E  9735  25     4     4   2.5  26.0    12   2650
##   Length Turn Displa Gratio
```

```
## 1    186   40    121   3.58
## 2    173   40    258   2.53
## 3    168   35    121   3.08
## 4    189   37    131   3.20
## 5    174   36     97   3.70
## 6    177   34    121   3.64

y <- auto
levels(y$Origin)

## [1] "A" "E" "J"

levels(y$Origin) <- c("America", "Europe", "Japan")
```

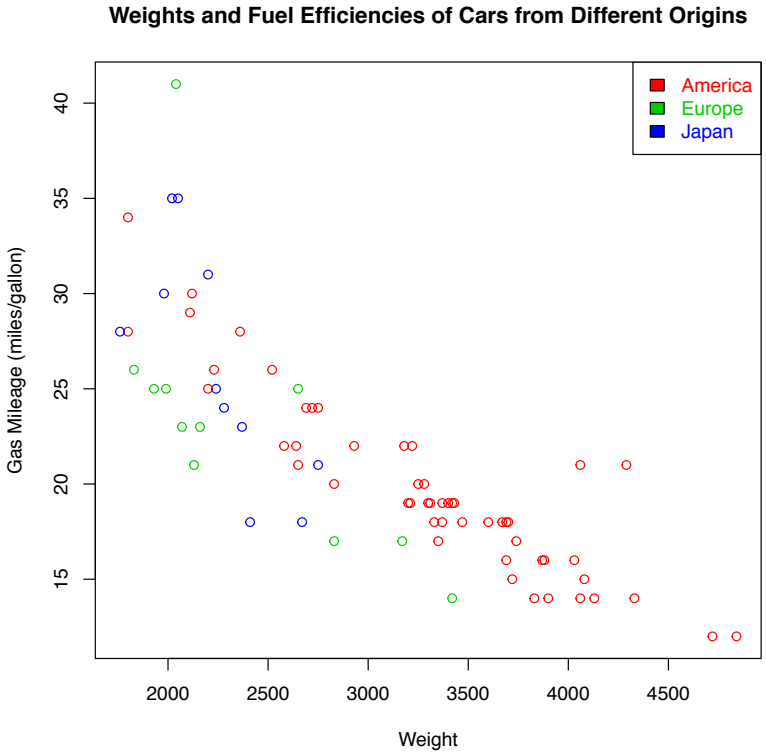
6.3.1 *Statistics*

In summarizing the relationship among these variables, we might want to use *least-squares lines*. We could ignore the categorical variable and just calculate the overall least-squares line for the two quantitative variables. Or we could calculate a different least-squares line for each category. One line is simpler than three, but the three lines will fit the data better. Yet again, we find ourselves facing the simplicity versus fit trade-off. We even have options between these two extremes, like finding a line for each category, while requiring that they all have the same slope. To decide which option makes the most sense, we need to visualize the data, so let's draw it before we do any calculating.

6.3.2 *Plots*

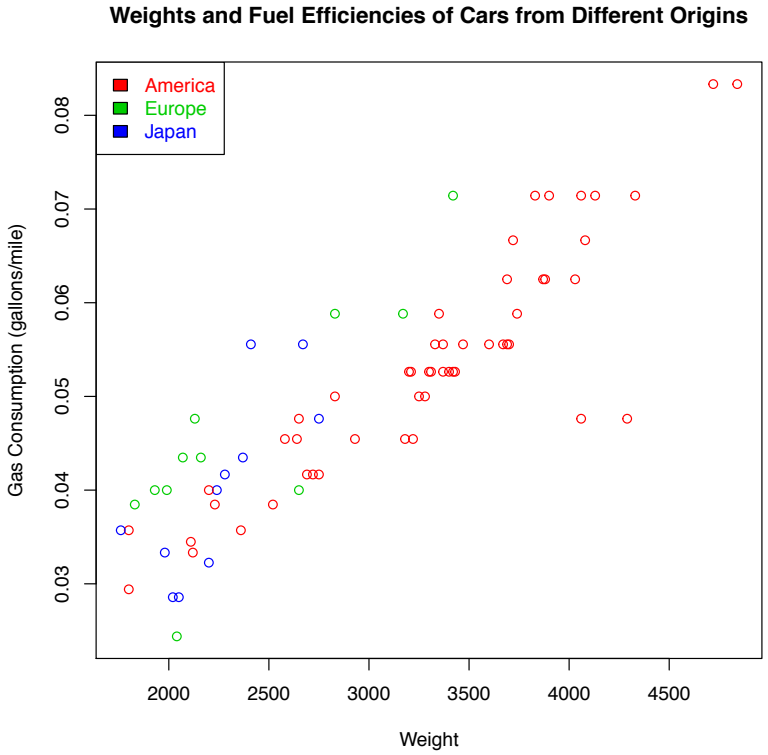
An intuitive way to plot the data is the *colored scatterplot* in which we create a scatterplot of the two quantitative variables using a different color for each point depending on the categorical variable's value.

```
plot(y$Weight, y$MPG, col=as.numeric(y$Origin)+1, xlab="Weight",
     ylab="Gas Mileage (miles/gallon)",
     main="Weights and Fuel Efficiencies of Cars from Different Origins")
k <- length(levels(y$Origin))
legend("topright", legend=levels(y$Origin),
       text.col=2:(k+1), fill=2:(k+1))
```



We can see that the two variables are negatively related. But a line doesn't really capture the relationship. In the grades dataset from Section 5.2.2, we successfully found a slightly more complicated curve that captured the pattern in the data. In this case, a transformation might make more sense. The data follow a curve that has the same basic shape as $y = 1/x$ (or equivalently $1/y = x$). Based on that observation, we might think that the reciprocal of one of the variables may be linearly related with the other variable. Let's take the reciprocal of the response variable, thus changing the measure of fuel efficiency (in miles per gallon) to a measure of fuel inefficiency (in gallons per mile).

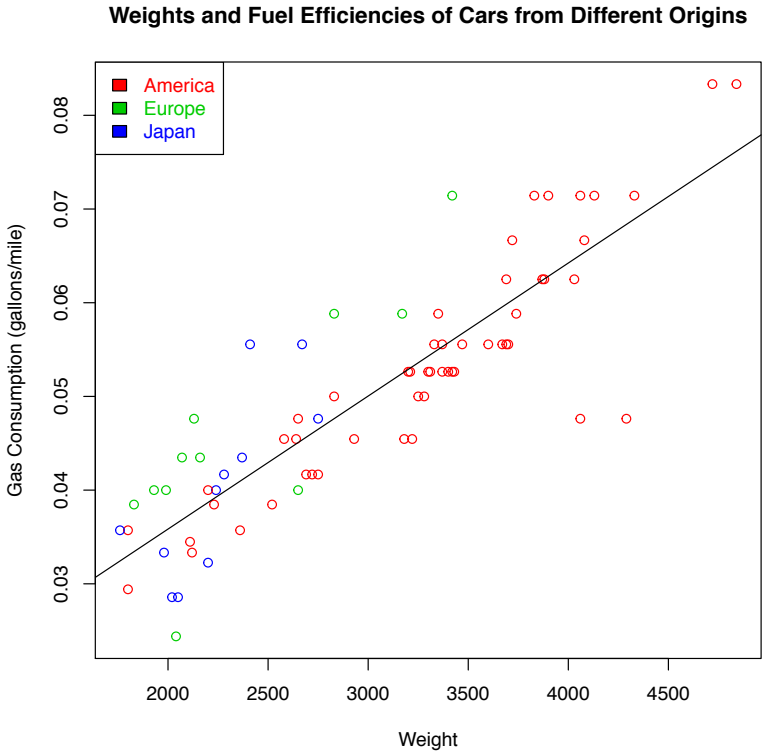
```
plot(y$Weight, 1/y$MPG, col=as.numeric(y$Origin)+1, xlab="Weight",
     ylab="Gas Consumption (gallons/mile)",
     main="Weights and Fuel Efficiencies of Cars from Different Origins")
legend("topleft", legend=levels(y$Origin),
       text.col=2:(k+1), fill=2:(k+1))
```



Now the relationships look much more linear. Let's try all three of the least-squares fits that we proposed in Section 6.3.1.

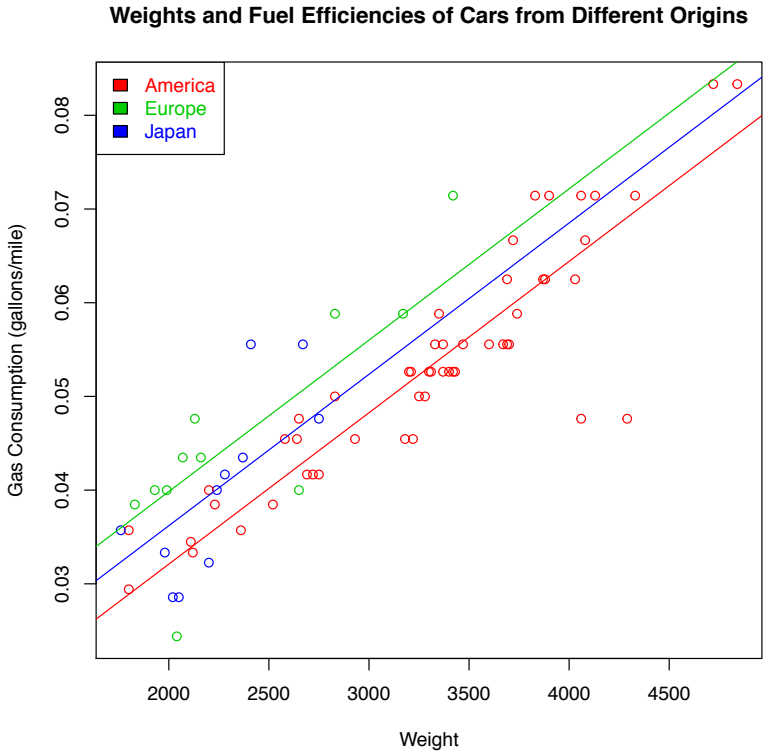
First, we draw a single least-squares line to summarize the overall relationship, ignoring the categorical variable's information.

```
fit1 <- lm(I(1/MPG) ~ Weight, data=y)
plot(y$Weight, 1/y$MPG, col=as.numeric(y$Origin)+1, xlab="Weight",
     ylab="Gas Consumption (gallons/mile)",
     main="Weights and Fuel Efficiencies of Cars from Different Origins")
legend("topleft", legend=levels(y$Origin),
       text.col=2:(k+1), fill=2:(k+1))
abline(fit1)
```



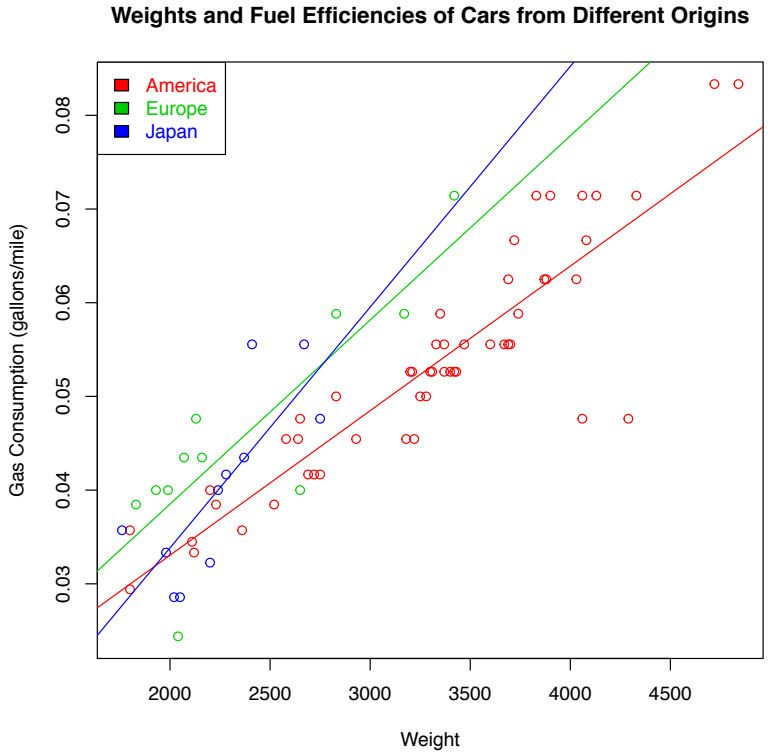
Next, we will allow the different categories to have different intercepts. This gives us three parallel least-squares lines.

```
fit2 <- lm(I(1/MPG) ~ Weight + Origin, data=y)
plot(y$Weight, 1/y$MPG, col=as.numeric(y$Origin)+1, xlab="Weight",
     ylab="Gas Consumption (gallons/mile)",
     main="Weights and Fuel Efficiencies of Cars from Different Origins")
legend("topleft", legend=levels(y$Origin),
       text.col=2:(k+1), fill=2:(k+1))
abline(fit2$coef[1], fit2$coef[2], col=2)
abline(fit2$coef[1] + fit2$coef[3], fit2$coef[2], col=3)
abline(fit2$coef[1] + fit2$coef[4], fit2$coef[2], col=4)
```



Finally, we also allow for *interactions*, which means that the categories' least-squares lines are also allowed to have different slopes.

```
fit3 <- lm(I(1/MPG) ~ Weight*Origin, data=y)
plot(y$Weight, 1/y$MPG, col=as.numeric(y$Origin)+1, xlab="Weight",
     ylab="Gas Consumption (gallons/mile)",
     main="Weights and Fuel Efficiencies of Cars from Different Origins")
legend("topleft", legend=levels(y$Origin),
       text.col=2:(k+1), fill=2:(k+1))
abline(fit3$coef[1], fit3$coef[2], col=2)
abline(fit3$coef[1] + fit3$coef[3], fit3$coef[2] + fit3$coef[5], col=3)
abline(fit3$coef[1] + fit3$coef[4], fit3$coef[2] + fit3$coef[6], col=4)
```



We won't spend any more time discussing which of these fits we should prefer. For descriptive analysis, it's a judgement call by the data analyst that depends on the purpose of the analysis. We will return to this question, however, in the context of inference (Chapter 9).

6.4 *Conclusion*

When analyzing both categorical and quantitative variables together, a typical approach is to split the data up by category (or by combination of categories) and analyze the quantitative separately for each group. Each technique we saw in this chapter follows that pattern. We summarize them in the table below to complete the *Description* column of the big picture (Figure 1.3).

	Description	
	Statistics	**Plots**
One Categorical and One Quantitative Variable	one-way table	bar chart side-by-side boxplots colored density plot
Two Categorical and One Quantitative Variables	two-way table	multiple bar charts multiple side-by-side boxplots
One Categorical and Two Quantitative Variables	least-squares lines	colored scatterplot

This concludes our survey of the descriptive stage of data analysis.

In Part III, we will revisit each of the mixes of variable types we've looked at already, this time asking some of the most common and useful inference questions.

Part III

Inference

7 *Inference on Categorical Data*

We begin our tour of inference by looking at categorical variables again. With a single categorical variable, we will consider all three inference tasks in the context of trying to figure out the proportion of the full population that belongs to a given category. With two categorical variables, we will consider the task of testing for independence of the two variables.

		Data Analysis		
		Description		Inference
		Statistics	Plots	
Categorical	1 C			You are here
	2 C			
Quantitative	1 Q			
	2 Q			
	3 Q			
Both	1 C, 1 Q			
	2 C, 1 Q			
	1 C, 2 Q			

7.1 *One Categorical Variable*

Our primary example of *iid* data in Chapter 3 came from coin-tossing. This example actually goes a very long way in helping us do inference on one categorical variable. In fact, if you assume that your observations are iid, then any categorical variable with only two categories can be modeled by a set of independent Bernoulli random variables just like the coin tosses. Even if there are more than two categories, you will see that many questions of interest can still be addressed by the same reasoning.

Often a data set consists of observations that are *not* independent, and thus not iid. An example is a sample of people from a population. Assume you take a random sample of one hundred different

Americans, and ask them if they will vote for Candidate A or Candidate B (for now, pretend that those are the only two options). Assume there are N total Americans to choose from, and that a of them are in favor of Candidate A. If the first person selected is an A-voter (which happens with probability a/N), then your selection second selection is from the $N-1$ remaining Americans, $a-1$ of which are A-voters. So for the second selection, the probability of getting an A-voter is $(a-1)/(N-1)$, which is slightly smaller than it was for the first selection. This is called "sampling without replacement," and it does not produce independent trials. On the other hand, if after each selection you were to leave the selected person in the pool for future selections, this is called "sampling with replacement." In our example, the probability of getting an A-voter would be a/N for each selection, regardless of the outcomes of the previous selections. Random sampling with replacement results in iid data.[1]

[1] If the population is a few orders of magnitude larger than the sample, then a random sample without replacement is still approximately iid and can be treated as such. For instance, the US population has a population of well over 300 million people. A typical survey might only sample a few hundred or a few thousand people. There's very little difference in doing the survey with or without replacement. For simplicity, you can definitely treat the data as if it were iid even if the survey was without replacement.

7.1.1 *Estimation*

A natural question when thinking about a categorical variable is, what is the *population proportion* of each category. If you have taken a sample (with replacement) of n people in the US asking them to choose between Candidates A and B, then your data would include the categorical variable with categories "A" and "B" indicating the response of each person surveyed. Define $p := a/N$ the proportion of Americans who prefer Candidate A. It is also the probability that each observation in your sample says "A." Each observation of this categorical variable is logically equivalent to a coin-flip, and can thus be modeled by a Bernoulli(μ) random variable, where a response of "A" is counted as a 1 and any other response is 0. Recall from Chapter 3 that the expected value of a Bernoulli(μ) random variable is exactly μ. Recall also that the expected value of the sample mean of any sample of identically distributed random variables is exactly equal to the expected value of the random variables in the sample. The sample mean in this case is just the proportion of observations that said "A." That is, the proportion of A-voters in the sample is an *unbiased estimator* of the proportion of A-voters in the American population. This reasoning applies to each category individually, no matter how many categories there are.

As an example, let's look at the computer files data yet again; let's take a random sample of 200 files (with replacement) from that population, and then estimate the population proportion of `DOC`s.

```
x <- read.csv("http://www.stat.yale.edu/~wdb22/Files.csv")
head(x)

##   type size
## 1  JPG 0.38
## 2  DOC 0.04
## 3  MP3 0.31
## 4  JPG 0.16
```

```
## 5  JPG 0.55
## 6  DOC 0.29

set.seed(1)
s <- sample(1:nrow(x), 200, replace=T)
y <- x[s, ]
dim(y)

## [1] 200   2

head(y)

##      type size
## 2656  DOC 0.37
## 3722  DOC 0.07
## 5729  JPG 0.19
## 9083  JPG 0.21
## 2017  JPG 0.26
## 8984  DOC 0.15

t <- table(y$type)
d <- as.numeric(t["DOC"]/sum(t))
d # is the sample proportion (and our estimate of population proportion)

## [1] 0.38
```

Variance is often used to assess the quality of an unbiased estimator. Let $\bar{X}$ be the sample proportion of A-voters, which is also a sample mean of Bernoulli(μ) random variables. We know that the variance of a sample mean is equal to the variance of the individual observations divided by n. Each Bernoulli(μ) observation has variance $\mu(1-\mu)$. Because we don't know μ, we also don't know the variance. However, it is interesting to note that we can upper bound the sample mean's variance in this case. It can be shown that the largest possible value for $\mu(1-\mu)$ is $1/4$ (occurring when $\mu = 1/2$). So we know that the variance of $\bar{X}$ is no larger than $1/4n$.

7.1.2 *Hypothesis Testing*

Back in Section 3.2.2 when hypothesis testing was first introduced, you saw an example of how it can be applied to the heads probability of a coin, and thus equivalently to the population proportion of a category. In that same chapter, you also saw that an example of a bell-shaped distribution for the sum of 20 Bernoulli(.5) random variables. Dividing by n doesn't change the shape, so the average of those random variables is also bell-shaped. In fact, the CLT guarantees that the distribution of an average of Bernoulli(μ) random variables (for any μ) will increasingly resemble a Normal distribution as the sample size increases. This section will assume that the sample size is large enough for a Normal approximation to be appropriate, which is certainly the case with large-scale opinion polling. Inference

using sample means is discussed in some detail in Section 8.1, so *you should skip ahead and read that section before proceeding*.

If the sample size is large enough, then $\bar{X}$ is approximately Normal[2] with expected value μ and variance $\text{Var}(X_i)/n$. Recall from the derivation in Section 3.1.3 that a Bernoulli(μ) random variable's variance is equal to $\mu(1-\mu)$. So $\bar{X} \overset{\text{approx}}{\sim} N(\mu, \mu(1-\mu)/n)$.

[2] The close μ is to zero, the larger the sample size needs to be for the Normal approximation to work well.

Consider a hypothesis specifying a particular value for the population proportion, $H_0 : \mu = \mu_0$. Assuming that hypothesis were true, then the following z-statistic is approximately standard Normal.[3]

$$z := \frac{\bar{X} - \mu_0}{\sqrt{\mu_0(1-\mu_0)/n}}$$

[3] Notice that the null hypothesis H_0 also *determines the variance* in this case because of the relationship between $E(X_i)$ and $\text{Var}(X_i)$ for Bernoulli random variables.

Therefore, the siginificance probability of the observed z-statistic is $2F(-|z|)$, with F representing the standard Normal cdf.

In our example, we'll consider the null hypothesis that the population proportion of DOCs is 0.5, and find the significance probability.

```
mu0 <- .5
s <- mu0*(1-mu0)

z <- (d-mu0)/sqrt(s/n)
p <- 2*pnorm(-abs(z))
```

7.1.3 *Confidence Intervals*

If the sample size is large enough, then

$$\frac{\bar{X} - \mu_0}{\text{SE}}$$

is approximately standard Normal. As described in Section 8.1, SE is an estimate of the standard deviation of $\bar{X}$ defined as $\hat{\sigma}/\sqrt{n}$. The estimated standard deviation $\hat{\sigma}$ of a quantitative variable was described in Chapter 5. In this context, another valid estimator that you could use for σ is $\sqrt{\bar{X}(1-\bar{X})}$; notice that it replaces the true μ with our estimate $\bar{X}$ in the expression $\sqrt{\mu(1-\mu)}$ which is exactly σ.

Observe that

$$P\left(-2 \leq \frac{\bar{X} - \mu}{\text{SE}} \leq 2\right) \approx .95$$

follows from the Normal approximation. Rearranging the inequalities gives an equivalent statement.

$$P\left(\bar{X} - 2\,\text{SE} \leq \mu \leq \bar{X} + 2\,\text{SE}\right) \approx .95$$

That means that the probability is about .95 that the interval from $\bar{X} - 2\,\text{SE}$ to $\bar{X} + 2\,\text{SE}$ will cover the true population proportion μ. We call the interval $[\bar{X} - 2\,\text{SE}, \bar{X} + 2\,\text{SE}]$ a 95% confidence interval.

An approximate 95% confidence interval for the population proportion of DOCs is.

```
SE <- d*(1-d)/sqrt(n)
c(d - 2*SE, d + 2*SE)

## [1] 0.25834 0.50166
```

The true population proportions are shown below for comparison.

```
table(x$type)/nrow(x)

##
##   DOC   JPG   MP3
## 0.400 0.499 0.101
```

7.2 *Two Categorical Variables*

When we discussed description of two categorical variables in Chapter 4, we looked at plots of conditional distributions of one categorical variable given the other. This led us to introduce the concept of *independence*; if the two categorical variables are unrelated to each other (i.e. knowing the value of one doesn't affect your prediction of the other) then we consider the variables independent. Informally, we said that if the conditional distributions all look about the same, then we might think the variables are in fact independent. Now, we'll take a formal approach to that question.

7.2.1 *Hypothesis Testing*

The *chi-squared test for independence* begins with the null hypothesis will be that the two variables are independent of each other. MORE TO WRITE.

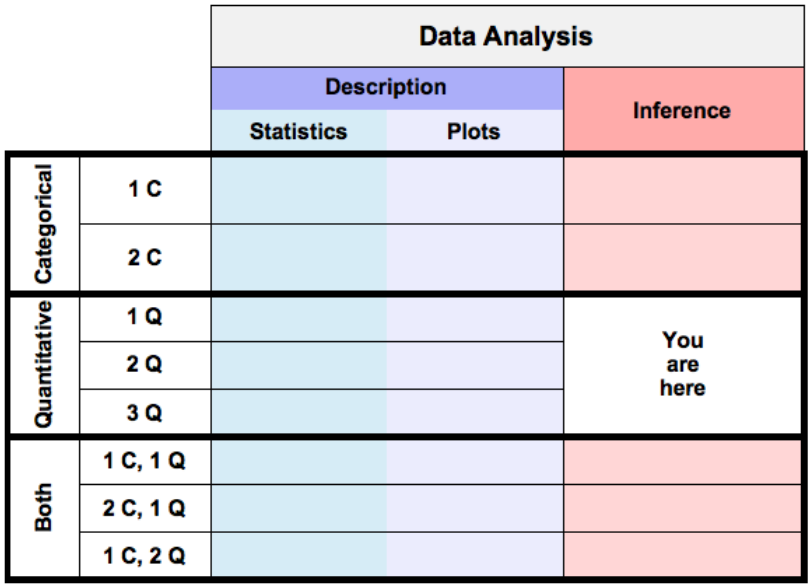
7.3 *Conclusion*

For any single categorical variable, we estimated the proportion of the population that belongs to each category. Using Normal approximation, we also covered the tasks of hypothesis testing and selecting confidence intervals as long as the sample size is large enough. With two categorical variables, we saw how to test the hypothesis that they are independent. The methods we covered in this section are collected into a piece of our concept map. The "E," "H," and "C" indicate which of the three inference tasks (estimation, hypothesis testing, and confidence intervals) are part of the method.

	Inference
One Categorical Variable	population proportion (E, H, C)
Two Categorical Variables	independence (H)

8 Inference on Quantitative Data

The simplest applications of statistical inference will be demonstrated in this chapter. In the context of one quantitative variable, we will go through a careful explanation and demonstration of each of the three tasks of inference applied to the population mean. You'll also see how Normal distributions and the Central Limit Theorem play an essential role. We will also briefly summarize the concept of a linear model for analyzing multiple quantitative variables together.



8.1 One Quantitative Variable

This section will focus on making inferences about the *population mean* of one quantitative variable. Throughout, we will assume that our n observations are modeled by iid random variables with a finite mean μ and finite variance σ^2. You may want to review the properties of the sample mean of iid random variables discussed in Chapter 3.

In this context, we'll undertake all three inference tasks in turn: estimation, hypothesis testing, and confidence intervals. As our example, we'll take the file sizes of the 4990 `JPG`s from the computer files dataset.

```
x <- read.csv("http://www.stat.yale.edu/~wdb22/Files.csv")
head(x)

##   type size
## 1  JPG 0.38
## 2  DOC 0.04
## 3  MP3 0.31
## 4  JPG 0.16
## 5  JPG 0.55
## 6  DOC 0.29

y <- x$size[x$type=="JPG"]
length(y)

## [1] 4990
```

These files constitute the "full population." We will take a random sample (with replacement) of 35 of the files, and then try to make inferences about the population based only on our sample.

```
n <- 35
set.seed(1)
z <- sample(y, size=n, replace=T)
```

8.1.1 *Estimation*

As show in Chapter 3, the sample mean $\bar{X}$ is an unbiased estimator of the population mean μ, regardless of the distribution of the individual random variables being averaged. Also show in that chapter, the variance of the sample mean is σ^2/n. This implies that the sample mean's distribution increasingy concentrates around its expectation of μ as the sample size increases.

In our example dataset z, the sample mean is .276 MB, so that is our estimate of the population mean.

```
m <- mean(z)
m

## [1] 0.276
```

8.1.2 *Hypothesis Testing*

Recall that the hypothesis testing process entails a number of steps:

1. Formulate a *null hypothesis* that you will try to "disprove."
2. Identify a statistic whose distribution would be known (or approximately known) if the null hypothesis were assumed to be true.
3. Calculate the value of this *test statistic* from your data set.

4. Find the probability that the test statistic would be at least as extreme as its calculated value, assuming the null hypothesis were true; this is called the *significance probability*.

5. Compare this significance probability to a pre-determined threshold (e.g. .05 for a 95% hypothesis test). If the significance probability is less than the threshold, then you *reject* the null hypothesis. Otherwise you *fail to reject* it.

The null hypothesis that we will test is that the population mean μ is equal to some specific value μ_0. Now if we assume that the population mean is μ_0, can we identify a statistic whose distribution we know? Consider the sample mean $\bar{X}$. It has expected value equal to the population mean, which we're assuming is μ_0. But the expected value is just one detail about a distribution. Is it possible for us to know the whole distribution?

It is an important fact that an average of Normal random variables also has a Normal distribution. So if the individual observations can be modeled by Normal random variables, then so can the sample mean. As noted above, many real-world mechanisms actually produce approximately Normal data, resulting in histograms that are roughly bell-shaped. If your data appear to be bell-shaped you may want to assume that the individual observations are well-modeled by a Normal distribution; then the sample mean is also well-modeled by a Normal distribution, in particular $N(\mu_0, \sigma^2/n)$. If you know σ^2, then you've fully specified a distribution for $\bar{X}$. Furthermore, the transformation

$$z := \frac{(\bar{X} - \mu_0)}{\sigma/\sqrt{n}}$$

would have a standard Normal distribution. Let's call this quantity the *z-statistic*.

Once you've calculated z from your data, you need to find the probability that a standard Normal distribution would take a value at least as extreme (in this case, at least as far from zero) as z. Letting F represent the standard Normal cdf and using the fact that the standard Normal distribution is symmetric about zero, this significance probability is simply $p := 2F(-|z|)$. Comparing this p-value to a threshold is called a *z-test*.

However, there is a glaring weakness in the technique we just saw: it assumed that we know σ^2. In practice, that's almost never the case! Instead, we have to estimate the standard deviation from the data; recall the definition of the estimated standard deviation $\hat{\sigma}$ from Section 5.1.1. Once we've estimated the standard deviation of the individual observations, a natural estimate of the sample mean's standard deviation is

$$\text{SE} := \frac{\hat{\sigma}}{\sqrt{n}},$$

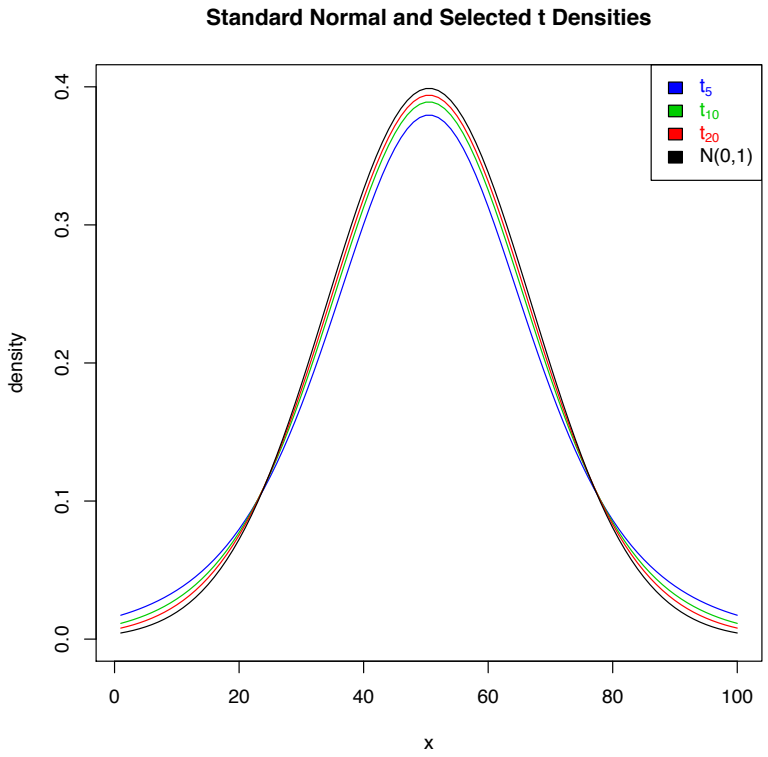
a quantity known as the *standard error*. Then instead of the z-statistic, which is unknowable, we calculate the *t-statistic* that replaces the

standard deviation of $\bar{X}$ by our estimate of it.

$$t := \frac{(\bar{X} - \mu_0)}{\text{SE}} \tag{8.1}$$

This statistic does not have a standard Normal distribution. It has a *t distribution with $n-1$ degrees of freedom* (which we will denote by t_{n-1}). t distributions have a similar shape to the standard Normal distribution, except that they have heavier tails. As n gets larger, the t_{n-1} distribution becomes closer and closer to the standard Normal distribution.

```
grid <- seq(-3, 3, length.out=100)
plot(sapply(grid, dt, df=5), type="l", col=4, ylab="density", xlab="x",
     main="Standard Normal and Selected t Densities", ylim=c(0, .4))
lines(sapply(grid, dt, df=10), col=3)
lines(sapply(grid, dt, df=20), col=2)
lines(sapply(grid, dnorm))
ltext <- c(expression(t[5]), expression(t[10]), expression(t[20]), "N(0,1
legend("topright", legend=ltext, text.col=4:1, fill=4:1)
```



t distributions are also symmetric around zero, so again the significance probability is

$$p := 2F(-|t|), \tag{8.2}$$

where in this case the F refers to the appropriate t distribution's cdf. Comparing this p-value to a threshold is called a *t-test*.

The above argument was predicated on the histogram of the variable being roughly bell-shaped. What if it's not? Then can we still

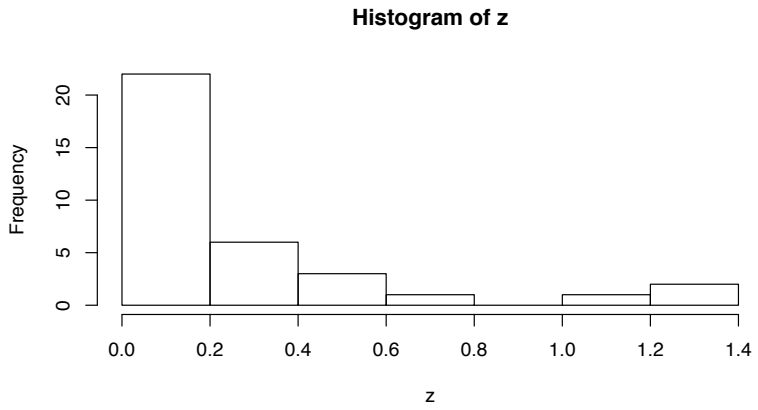
figure out the distribution of $\bar{X}$? Well, according to the CLT, a sample mean of random variables that aren't Normal also tends to be approximately Normal if the sample size is large enough (many data analysts use 30 as a rule of thumb). So you can use a t-test as long as n is large enough, no matter what the histogram of the data looks like.[1]

[1] There are other techniques available if the sample size is smaller, but they are beyond the scope of this book.

Let me reiterate this section's main message. Assume you have a quantitative variable in your dataset, and you want to test the proposition that its population mean is μ_0. If the sample size is at least 30, you can use a t-test. If the sample size is smaller than that, but the histogram is roughly bell-shaped, then you can still use a t-test.

Next, we'll work out a simple example in R with our file sizes. Let's test the null hypothesis that the population mean of `JPG` file sizes is .5. A histogram of the data confirms that they *aren't* bell-shaped, but that's okay because we can rely on the CLT and the sample-size-30 rule. Our present sample size is 35, so we will assume that the distribution of $\bar{X}$ is approximately Normal even though the data aren't.

```
hist(z)
```



Next, we need to calculate the t-statistic, as defined in equation (8.1).

```
SE <- sd(z)/sqrt(n)
t <- (m - .5)/SE
t

## [1] -3.731
```

If $\bar{X}$ were Normal with expectation .5 and variance $\sigma^2/35$, then the t-statistic would have a t_{34} distribution. The significance probability is the probability that t_{34} distributed random variable is at least as extreme as -3.73. According to the formula (8.2),

```
p <- 2*pt(-abs(t), df=n-1)
p

## [1] 0.00069502
```

The probability is only about .0007 that the sample mean would be at least this far from .5 if the true population mean were .5. This is very strong evidence against the null hypothesis. We would easily reject the null hypothesis in a .05-level test (i.e. a 95% hypothesis test).

8.1.3 *Confidence Intervals*

Recall that a standard Normal random variable has probability .95 within two standard deviations of its mean. Letting $Y \sim N(\mu, \sigma^2)$, we can express that fact mathematically as follows.

$$P(\mu - 2\sigma \leq Y \leq \mu + 2\sigma) \approx .95$$

Rearranging, we get an equivalent statement about standard Normal random variables.

$$P(-2 \leq \frac{Y - \mu}{\sigma} \leq 2) \approx .95$$

That is, if you add up the area under the standard Normal density in the region between -2 and 2, you get about .95.

```
pnorm(2) - pnorm(-2)

## [1] 0.9545
```

What if you didn't know how far out you needed to go from the mean to cover a desired amount of probability? Then you can use the *quantile* function, specifying how much probability you want to cover. Specifically, for a random variable X, the quantile function $q(p)$ is defined to be the smallest a value such that $P(X \leq a)$ is at least p. For example, if you want to find out how far you need to go before you've accumulated .975 probability under the standard Normal density curve, that is $q(.975)$ for the standard Normal quantile function q.

```
qnorm(.975)

## [1] 1.96
```

This is about 2.0. That means that the interval from $-\infty$ to 2 has probability about .975. In other words, there is only about .025 probability of being greater than 2. By symmetry, there is also only about .025 probability of being less than -2. Thus there is a total probability of .05 of being outside of $[-2, 2]$, a fact that you already knew.

The t distributions are also symmetric, so you can also find the margin required to cover 95% of the probability by finding the .975 quantile. In R, you can find the .975 quantile of the t_{20} distribution, for instance, using the command below.

```
qt(.975, df=20)

## [1] 2.086
```

If the degrees of freedom is 28 or more, then the .975 quantile rounded to two significant digits is 2.0, just like the standard Normal .975 quantile.

Now we will apply this reasoning to sample means. Recall from our discussion of hypothesis testing that if the sample size is at least 30, or if the sample size is smaller than that but the histogram is roughly bell-shaped, then

$$\frac{(\bar{X} - \mu)}{\text{SE}} \tag{8.3}$$

is approximately t_{n-1} distributed. Note that this uses the true mean rather than the hypothesized mean that we saw in the definition of the t-statistic. This is not a statistic we can calculate because we don't know μ. We will nonetheless be able to use this observation to our advantage.

Let the symbol $q_d(.975)$ represent the .975 quantile of the t_d distribution. We observed that the expression 8.3 above is approximately t_{n-1} distributed, so

$$P(-q_{n-1}(.975) \leq \frac{(\bar{X} - \mu)}{\text{SE}} \leq q_{n-1}(.975)) \approx .95$$

Rearranging, we can reach the equivalent statement

$$P(\bar{X} - q_{n-1}(.975)\,\text{SE} \leq \mu \leq \bar{X} + q_{n-1}(.975)\,\text{SE}) \approx .95$$

Consider the interval from $\bar{X} - q_{n-1}(.975)\,\text{SE}$ to $\bar{X} + q_{n-1}(.975)\,\text{SE}$. It has about a 95 percent chance of containing the true population mean μ. This interval is called a 95% confidence interval for the population mean. Observe that the interval is centered at $\bar{X}$ the estimate for the mean, and extends in either direction a distance of $q_{n-1}(.975)\,\text{SE}$, often called the *margin of error*. Another way of writing the confidence interval demonstrates this more clearly: $\bar{X} \pm q_{n-1}(.975)\,\text{SE}$.

Recall the definition of standard error as $\frac{\hat{\sigma}}{\sqrt{n}}$. You can see that the larger the sample size is the smaller the standard error tends to be. Additionally $q_{n-1}(.975)$ decreases as n increases. So larger samples result in smaller margins of error.[2] However, the higher level of confidence you want, the larger the margin of error must be. For instance, a 99% confidence interval would replace $q_{n-1}(.975)$ with $q_{n-1}(.995)$ and result in a larger margin of error than the 95% confidence interval created from the same data.[3]

It is interesting to note that the set of values in the confidence interval is exactly the same as the set of μ_0 that the t-test would fail to reject.

For our example fize size data, we first need to find the t_{34} distribution's .975 quantile, then extend our interval by that many standard errors to either side of the sample mean.

```
q <- qt(.975, df=n-1)
q

## [1] 2.0322
```

[2] This should seem intuitive. More data points means an improved ability to identify where the true mean is.

[3] This should also be intuitive. If you want to be more sure that you'll catch the fish, then you should use a bigger net.

```
CI <- c(m - q*SE, m + q*SE)
CI

## [1] 0.15399 0.39801
```

The interval from .15 to .40 is a 95% confidence interval for the population mean. The true population mean is about .25, which is well within the confidence interval's bounds.

```
mean(y)

## [1] 0.24856
```

For simplicity, you might want to just use $\bar{X} \pm 2\,\text{SE}$ as long as the sample size is at least 30. After all, in those cases, $q_{n-1}(.975)$ rounded to two significant digits is 2.0. Just add this trick to your sample-size-30 rule.

8.2 Two Quantitative Variables

Next, we'll consider ways to do inference on two quantitative variables together. Sometimes you're primarily interested in the difference between the two variables; the two variables are called a *paired sample*. For example, consider a data frame `x` that contains, for n subjects, the subject's systolic blood pressure measured at age thirty (`bp30`) and the same subject's systolic blood pressure measured at age forty (`bp40`).[4] If you're interested in how blood pressures change over the decade from age twenty to thirty, then you should create a new variable (`x$change <- x$bp40 - x$bp30`). Then questions about the difference between these two quantitative variables reduce to questions about the single quantitative variable `change`; then you simply need to apply your knowledge of inference on one quantitative variable from Section 8.2. Estimating the average change in blood pressure is equivalent to estimating the population mean of the `change` variable. Often, we want to ask if there is any change on average; this is equivalent to performing a hypothesis test in which the null hypothesis is that the population mean of `change` is zero. Likewise a confidence interval for the difference can be found by looking at the `change` variable.

[4] The data described here could also be written in a data frame with one categorical and one quantitative variable. The categorical variable would provide the age (either 30 or 40) at which the blood pressure was measured. But a data frame of that sort would not give us any indication which blood pressure measurements belong to the same people. Also, the observations wouldn't be independent. Arranging the data into two quantitative variables as described here is much better.

```
# Simulate blood pressure data
n <- 100
set.seed(1)
x <- data.frame(bp30=115+round(4*rnorm(n)),
                bp40=120+round(5*rnorm(n)))
head(x)

##   bp30 bp40
## 1  112  117
## 2  116  120
```

```
## 3  112  115
## 4  121  121
## 5  116  117
## 6  112  129

# Create a new variable for the change
x$change <- x$bp40 - x$bp30
head(x)

##   bp30 bp40 change
## 1  112  117      5
## 2  116  120      4
## 3  112  115      3
## 4  121  121      0
## 5  116  117      1
## 6  112  129     17

m <- mean(x$change)
m

## [1] 4.38

SE <- sd(x$change)/sqrt(n)
SE

## [1] 0.59861

# Consider the hypothesis that the average change is zero
t <- m/SE
t

## [1] 7.3169

p <- 2*pt(-abs(t), df=n-1)
p

## [1] 6.7552e-11

# 95% confidence interval for change in blood pressure
c(m - 2*SE, m + 2*SE)

## [1] 3.1828 5.5772
```

Typically, however, any two different quantitative variables in your data frame aren't comparable and/or you aren't interested in their difference. Recall from Chapter 5 that we made scatterplots and calculated the least-squares line to visualize and summarize the relationship between the variables. This process is formalized in a branch of inference called *linear models*, in which you assume that the response variable is a linear function of some set of parameters[5] plus some independent Normal$(0, \sigma^2)$ random errors for each of the observation.[6] If we let $Y_1, \ldots, Y_n$ denote the response variable,

[5] Notice that the term "linear" refers, perhaps surprisingly, to the relationship between the response variable and the parameters rather than the relationship between the response variable and the explanatory variables.

[6] In fact, many interesting results in linear models do not need the assumption that the errors are Normal.

and $X_1, \ldots, X_n$ denote the explanatory variable, then the *simple linear model* can be expressed as follows.

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \tag{8.4}$$

Below, we will see how to estimate the parameters β_0 and β_1, also known as the *linear model coefficients*, and how to test whether we should reject the hypothesis that β_1 is zero.

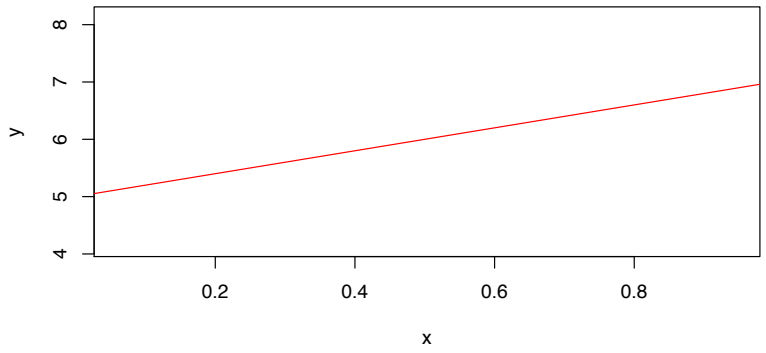
8.2.1 Estimation

We've already seen one way of estimating the slope of the linear relationship between two quantitative variables. In Chapter 5, we found the least-squares line and gave a formula for its slope and intercept. In fact, it can be shown that the least-squares intercept and slope (here we'll call them $\hat{\beta}_0$ and $\hat{\beta}_1$) are unbiased estimators for β_0 and β_1. The difference is that now we are assuming that there is some true line that is generating the data according to (8.4). For instance, imagine that the true mechanism generating the response variable is

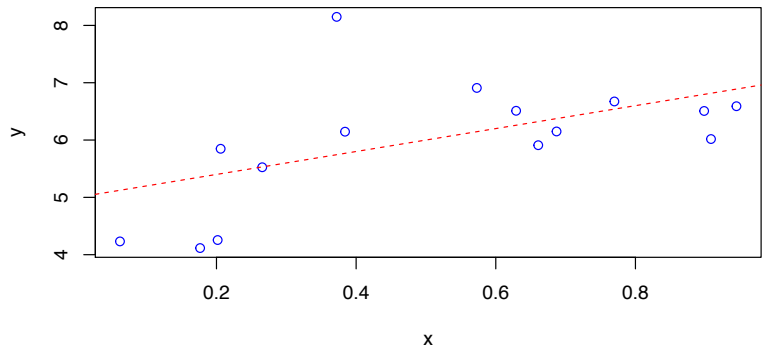
$$Y_i = 5 + 2X_i + \epsilon_i$$

with $\sigma^2 = 1$ as the variance of the ϵ_i. Let's take a look at the true line then at some data drawn from that mechanism.

```
# Simulating the data
set.seed(1)
n <- 15
x <- runif(n)
b0 <- 5; b1 <- 2
y <- b0 + b1*x + rnorm(n)

plot(x, y, type="n")
abline(b0, b1, col=2)
```

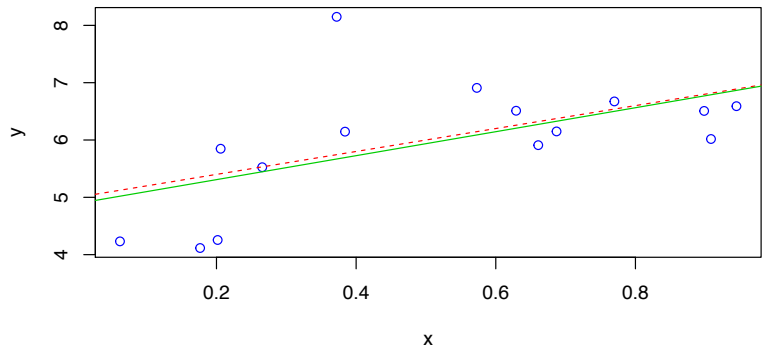


```
plot(x, y, col=4)
abline(b0, b1, col=2, lty=2)
```



In practice, we don't know the true line, but we can estimate one from the data. It won't be equal to the *true* line, but hopefully it will be close.

```
plot(x, y, col=4)
abline(b0, b1, col=2, lty=2)
fit <- lm(y ~ x)
abline(fit, col=3)
```



On average, the larger the sample, the more accurate our estimated line will be.

One primary reason for estimating the linear model coefficients is to predict future response variable values from expanatory variable values. Suppose you'll get a new observation of only the explanatory variable; let's call it X_{n+1}. If you knew the true coefficients, then $\beta_0 + \beta_1 X_{n+1}$ would be your best guess for what the corresponding response variable value Y_{n+1} will be. But in reality, you don't know the linear model coefficients; you only know the past data. But you can use your past data to estimate the linear model coefficients! Then you can use those estimates to predict what the new response vari-

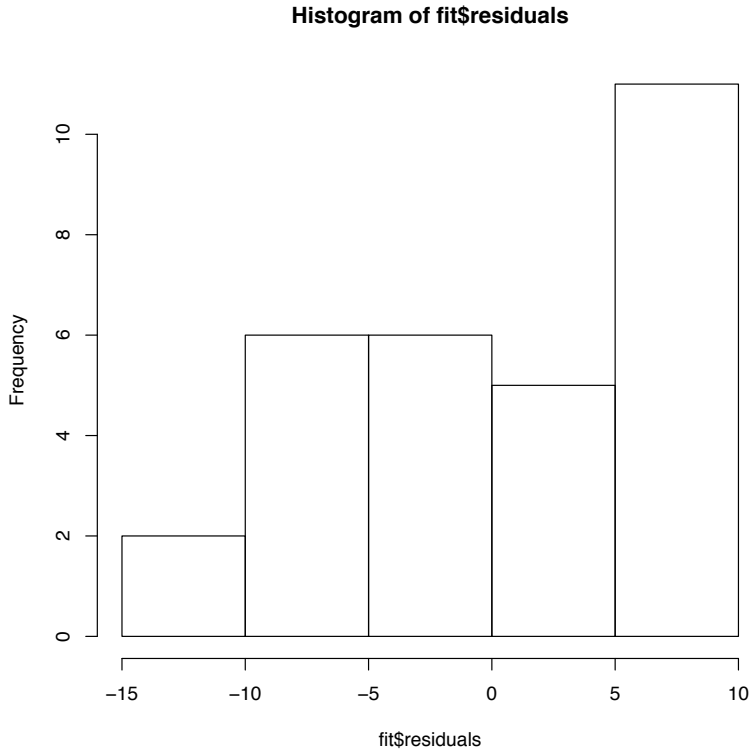
able value will be:

$$\hat{Y}_{n+1} := \hat{\beta}_0 + \hat{\beta}_1 X_{n+1}.$$

```
head(attitude)

##   rating complaints privileges learning raises critical advance
## 1     43         51         30       39     61       92      45
## 2     63         64         51       54     63       73      47
## 3     71         70         68       69     76       86      48
## 4     61         63         45       47     54       84      35
## 5     81         78         56       66     71       83      47
## 6     43         55         49       44     54       49      34

fit <- lm(rating ~ complaints, data=attitude)
fit

##
## Call:
## lm(formula = rating ~ complaints, data = attitude)
##
## Coefficients:
## (Intercept)   complaints
##      14.376        0.755
```

To check that the errors are approximately Normal, you might want to make a histogram of the residuals and look for a bell-shape.

```
hist(fit$residuals)
```



With only 15 observations, it isn't too surprising that the histogram isn't convincingly bell-shaped. You can use simulation to get a sense for how bell-shaped a histogram of 15 Normal draws *should* look by repeatedly running `hist(rnorm(15))`.

The linear model reasoning extends to more complicated relationships between the response and explanatory variables. For instance, the linear model expressing the response variable as a quadratic function of the explanatory variable plus error is[7]

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \epsilon_i.$$

[7] Remember, the "linear" part refers to the relationship between the response variables and the parameters (not the explanatory variable), so the term *linear model* is still appropriate here.

8.2.2 *Hypothesis Testing*

If the assumptions of the linear model hold, then $\hat{\beta}_1$ is Normally distributed. As shown below, the `summary` function applied to a `lm` fit calculates a standard error[8] (i.e. an estimated standard deviation) for $\hat{\beta}_1$.

[8] I won't explain how this standard error is defined. Just trust that R is doing the right thing here.

```
summary(fit)

##
## Call:
## lm(formula = rating ~ complaints, data = attitude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.880  -5.990   0.178   6.298   9.629
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14.3763     6.6200    2.17    0.039 *
## complaints    0.7546     0.0975    7.74    2e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.99 on 28 degrees of freedom
## Multiple R-squared:  0.681,Adjusted R-squared:  0.67
## F-statistic: 59.9 on 1 and 28 DF,  p-value: 1.99e-08
```

Consider the hypothesis that the true β_1 is zero. If that were true, then the estimate $\hat{\beta}_1$ divided by the standard error has a t_{n-k} distribution, where k is the number of linear model coefficients that you estimated for your fit.

$$t := \frac{\hat{\beta}_1}{\text{SE}}$$

As usual, the significance probability is 2 times the appropriate t distribution's cdf evaluated at $-|t|$. This significance probability is part of the `summary` output as well, as you can see in the R output above. If you reject this null hypothesis, then you are affirming that explanatory variable actually does have a term in the true linear model (8.4) between the variables.

8.3 Three Quantitative Variables

As you saw in the previous section, sometimes you're more interested in the difference between two quantitative variables than you are about the variables themselves. Whenever you identify such a pair, then you should create a new variable corresponding to the n differences. This could turn an analysis of three quantitative variables into an analysis of two quantitative variables, which we've already covered.

Often, however, you want to analyze three (or more) quantitative variables together. The principles from linear models can be directly extended to this case for the equation. This time we have two explanatory variables $X_{1,1}, \ldots X_{1,n}$ and $X_{2,1}, \ldots X_{2,n}$, so the equation becomes

$$Y_i = \beta_0 + \beta_1 X_{1,i} + \beta_2 X_{2,i} + \epsilon_i.$$

8.3.1 Estimation

Again, we want to estimate the linear model coefficients. And yet again, the least-squares plane provides unbiased estimators.

```
fit2 <- lm(rating ~ complaints + raises, data=attitude)
fit2
```

```
##
## Call:
## lm(formula = rating ~ complaints + raises, data = attitude)
##
## Coefficients:
## (Intercept)   complaints       raises
##     11.9873       0.7128       0.0801
```

Notice that the coefficient estimate for `complaints` is a little different now that the `raises` variable is included in the model.

8.3.2 *Hypothesis Testing*

As before, the `summary` command gives us significance probabilities for testing each of the different null hypotheses that one particular linear model coefficient is zero.

```
summary(fit2)

##
## Call:
## lm(formula = rating ~ complaints + raises, data = attitude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.514  -6.496   0.205   6.233   9.591
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.9873     8.4226    1.42     0.17
## complaints    0.7128     0.1331    5.35  1.2e-05 ***
## raises        0.0801     0.1705    0.47     0.64
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.09 on 27 degrees of freedom
## Multiple R-squared:  0.684,Adjusted R-squared:  0.66
## F-statistic: 29.2 on 2 and 27 DF,  p-value: 1.77e-07
```

The significance probability for `complaints` isn't as small as it was in our first fit, but it's still small enough to indicate solid evidence against the null hypothesis that β_1 is zero. The significance probability for `raises`, however, isn't very small; you don't have much evidence against the claim that the true value of the coefficient β_2 is zero. However, if you consider the linear model that only includes the `raises` variable, you find that the significance probability is tiny.

```
fit3 <- lm(rating ~ raises, data=attitude)
summary(fit3)
```

```
##
## Call:
## lm(formula = rating ~ raises, data = attitude)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -19.12  -7.11   1.38   5.74  19.57
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   19.978     11.688    1.71   0.0985 .
## raises         0.691      0.179    3.87   0.0006 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10 on 28 degrees of freedom
## Multiple R-squared:  0.348,Adjusted R-squared:  0.325
## F-statistic:   15 on 1 and 28 DF,  p-value: 0.000598
```

The estimate and the significance probability of each coefficient depends on what other variables are being included in the model!

8.4 Conclusion

A natural estimator for a population mean is the sample mean. Sample means are often approximately Normally distributed. Subtracting the mean and dividing by the standard error results in a quantity that is approximately t_{n-1} distributed. Using this insight, we derived techniques to do hypothesis tests and to construct confidence intervals for the population mean.

When we want to analyze the relationship between multiple quantitative variables, we can estimate the linear model coefficients using the least-squares approach seen in Chapter 5. R's output tells us whether or not these coefficients are significantly different from zero based on the data.

	Inference
One Quantitative Variable	population mean (E, H, C)
Two Quantitative Variables	paired samples (E, H, C) linear model coefficients (E, H)
Three Quantitative Variables	linear model coefficients (E, H)

9 Inference on Both Types Together

		Data Analysis		
		Description		Inference
		Statistics	Plots	
Categorical	1 C			
	2 C			
Quantitative	1 Q			
	2 Q			
	3 Q			
Both	1 C, 1 Q			You are here
	2 C, 1 Q			
	1 C, 2 Q			

9.1 One Categorical Variable and One Quantitative Variable

As in Chapter 6, we can use the categorical variable to split up the quantitative variable values into groups. In this section, we'll be interested in the difference between the groups' population means. If there are only two groups, we can easily estimate this difference, test whether it's zero, and derive a confidence interaval. If there are more than two groups, our main interest will be in whether or not there is a difference among their means.

9.1.1 Estimation

In the case of two groups, let's use $X_1, X_2, \ldots$ and $Y_1, Y_2, \ldots$ to denote the values of the quantitative variable for the different groups. Let μ_X and μ_Y be the two population means. $\bar{X}$ and $\bar{Y}$ are natural estimators for the population means, and $\bar{X} - \bar{Y}$ is an unbiased estimator for the difference between the two means.

$$\begin{aligned} E(\bar{X} - \bar{Y}) &= E(\bar{X}) - E(\bar{Y}) \\ &= \mu_X - \mu_Y \end{aligned}$$

Let's look at the `survey` dataset again to demonstrate; `Sex` will be the categorical variable, and `Pulse` will be the quantitative variable. The following code calculates the estimated difference in the mean pulse between the two genders (female mean pulse minus male mean pulse). It also draws side-by-side boxplots (to which I've added dots for the groups' sample means).

```
library(MASS)
head(survey)
```

```
##      Sex Wr.Hnd NW.Hnd W.Hnd    Fold Pulse    Clap Exer Smoke Height
## 1 Female   18.5   18.0 Right  R on L    92    Left Some Never 173.00
## 2   Male   19.5   20.5  Left  R on L   104    Left None Regul 177.80
## 3   Male   18.0   13.3 Right  L on R    87 Neither None Occas     NA
## 4   Male   18.8   18.9 Right  R on L    NA Neither None Never 160.00
## 5   Male   20.0   20.0 Right Neither    35   Right Some Never 165.00
## 6 Female   18.0   17.7 Right  L on R    64   Right Some Never 172.72
##        M.I    Age
## 1   Metric 18.250
## 2 Imperial 17.583
## 3     <NA> 16.917
## 4   Metric 20.333
## 5   Metric 23.667
## 6 Imperial 21.000
```

```
z <- survey[, c("Sex", "Pulse")]
z <- z[complete.cases(z), ]
dim(z)
```

```
## [1] 191   2
```

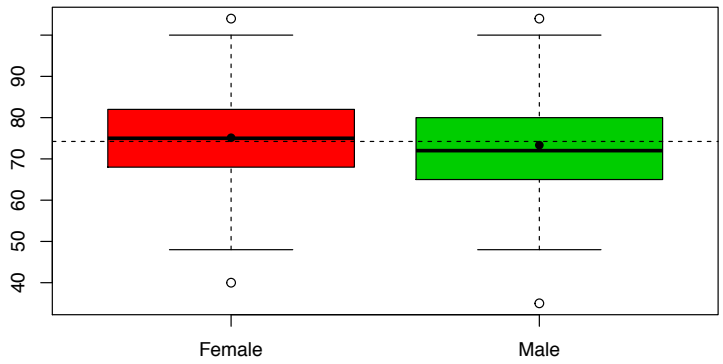
```
n <- nrow(z)
k <- length(levels(z$Sex))

# There is one more male than there are females.
# We'll discard one male at random for this example.
table(z$Sex)
```

```
##
## Female   Male
##     95     96
```

```
set.seed(1)
males <- which(z$Sex=="Male")
remove <- sample(males, 1)
z <- z[-remove, ]
table(z$Sex)
```

```
##
## Female   Male
##     95     95
```

```
n <- nrow(z)

# Draw the boxplots with dots for means.
boxplot(Pulse ~ Sex, data=z, col=2:(k+1))
m <- mean(z$Pulse)
abline(h=m, lty=2)
means <- aggregate(Pulse ~ Sex, data=z, mean)
points(1:nrow(means), means$Pulse, pch=16)

# Estimated difference in means
d <- means$Pulse[1] - means$Pulse[2]
d

## [1] 1.7895
```



In the case of more than two groups, the quantitative variable can be split up by group, and each group's population mean can be estimated by its sample mean.

9.1.2 *Hypothesis Testing*

We'll start with a partial explanation of testing the hypothesis that two groups' population means are equal, showing that it can be approached similarly to the hypothesis testing problem from Section 8.1.2. To simplify things, let's assume that an equal number of observations belong to each group, $n/2$ each. We'll also need to assume that the two groups have the same variance:[1] $\text{Var}(X_i) = \sigma^2 = \text{Var}(Y_i)$. Then, if the observations are approximately Normal or if the sample size is large enough, the sample means are approximately Normal.[2] More specifically,

$$\bar{X} \overset{\text{approx}}{\sim} N\left(\mu_X, \frac{\sigma^2}{n/2}\right) \qquad \text{and} \bar{Y} \overset{\text{approx}}{\sim} N\left(\mu_Y, \frac{\sigma^2}{n/2}\right)$$

Using the expected value and variance rules from Chapter 3, the distribution of the difference $\bar{X} - \bar{Y}$ must be $N(\mu_X - \mu_Y, 4\sigma^2/n)$. To (approximately) standardize this variable, we need to estimate σ. In

[1] A side-by-side boxplot can give you a visual impression of whether or not the different groups seem to have about the same spread.

[2] You might think that this section requires $n \geq 60$, so that each group's sample size is at least 30. However, as you'll see, we're primarily going to be interested in $\bar{X} - \bar{Y}$, which is a sum of n random variables. By the CLT phenomenon, it is more Nomral than either $\bar{X}$ or $\bar{Y}$. So you can still use the sample-size-30 rule for this method.

this chapter we will use a slightly different estimator than the $\hat{\sigma}$ defined in Chapter 5 and used in Chapter 8.[3] Instead, we use s defined by

$$s^2 := \frac{\sum(X_i - \bar{X}) + \sum(Y_i - \bar{Y})}{n - 2}$$

Our estimate of the standard deviation of $\bar{X} - \bar{Y}$ is the square root of its variance, substituting s for σ:

$$\text{SE} := \frac{2s}{\sqrt{n}}$$

The (approximately) standardized variable

$$\frac{(\bar{X} - \bar{Y}) - (\mu_X - \mu_Y)}{\text{SE}}$$

has a t_{n-2} distribution.[4]

Now consider the null hypothesis that $\mu_X = \mu_Y$. Assuming this is true, the numerator of the standardized variable simplifies, and we can define the t-statistic

$$t := \frac{\bar{X} - \bar{Y}}{\text{SE}} \overset{\text{approx}}{\sim} t_{n-2}$$

Once you've calculated this test statistic, the significance probability is simply $2F(-|t|)$, where F is the t_{n-2} cdf. If you compare this p-value to a threshold, this is called a *two-sample t-test*.

We continue the survey data example below.

[3] Actually, the new estimator we define can be considered a generalization of the one you already know.

[4] It has *exactly* this distribution if the data are exactly Normal; on the other hand, it has *approximately* this distribution if you're relying on the CLT.

```
sum.squares.res <- function(v) {
  ss <- sum((v - mean(v))^2)
  return(ss)
}
SSRs <- aggregate(Pulse ~ Sex, data=z, sum.squares.res)
SSRs

##      Sex Pulse
## 1 Female 12230
## 2   Male 13503

SSR <- sum(SSRs$Pulse)
SSR

## [1] 25734

sigma.hat <- sqrt(SSR/(n-2))
SE <- 2*sigma.hat/sqrt(n)

t <- d/SE
t

## [1] 1.0541

p <- 2*pt(-abs(t), df=n-2)
p

## [1] 0.29317
```

The R function `t.test` provides a simpler way to do this.

```
t.test(Pulse ~ Sex, data=z, var.equal=TRUE)

##
##  Two Sample t-test
##
## data:  Pulse by Sex
## t = 1.05, df = 188, p-value = 0.29
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.5592  5.1382
## sample estimates:
## mean in group Female   mean in group Male
##               75.126               73.337

summary(aov(Pulse ~ Sex, data=z))

##              Df Sum Sq Mean Sq F value Pr(>F)
## Sex           1    152     152    1.11   0.29
## Residuals   188  25734     137
```

If you have unequal group sizes or if the two groups don't have the same variance, things work a little differently. The theory is a bit more complicated to derive, so we won't cover it. But you can still use the `t.test` function, and it will do the right thing.

What if there are more than two groups? In that case, we can still test for a difference using *one-way ANOVA*.[5] Let k be the number of different categories represented in the dataset. We don't need equal group sizes, but ANOVA theory does require that the groups have normal distributions with equal variances. This time the test statistic that we define will have an F distribution. The family of F distributions are identified by two parameters called the *numerator degrees of freedom* (`df1` in R) and the *denominator degrees of freedom* (`df2`). For example, the plot below shows the pdf of the $F_{2,189}$ distribution.

[5] *ANOVA* stands for "analysis of variance."

```
grid <- seq(0, 10, length.out=100)
density <- sapply(grid, df, df1=2, df2=189)
plot(grid, density, type="l", col=3, xlab="x",
     main="The F distribution with 2 and 189 degrees of freedom")
abline(h=0, lty=2)
```

Let's take a look at an example from the `survey` dataset, using exercise as the categorical variable and pulse as the quantitative variable.

```
x <- survey[, c("Pulse", "Exer")]
# Change the order of Exer from (Freq, None, Some)
# to (None, Some, Freq)
x$Exer <- factor(x$Exer, levels(x$Exer)[c(2, 3, 1)])
# Drop the observations with missing values
```

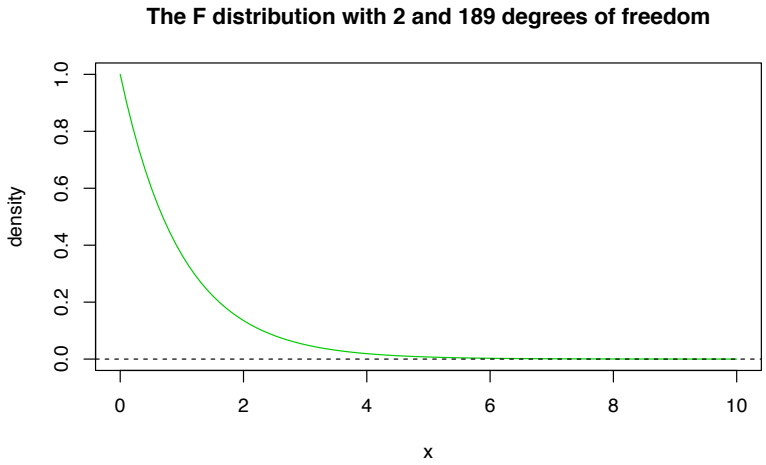
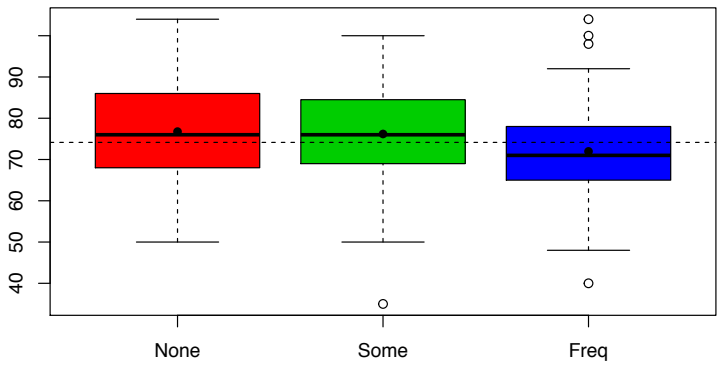


Figure 9.1: The pdf of the F distribution with 2 and 189 degrees of freedom, shown from 0 to 10. The vast majority of the probability is in the region shown, but the density is positive from 0 to infinity.

```
x <- x[complete.cases(x), ]
dim(x)

## [1] 192   2

n <- nrow(x)
k <- length(levels(x$Exer))

boxplot(Pulse ~ Exer, data=x, col=2:(k+1))
m <- mean(x$Pulse)
abline(h=m, lty=2)
means <- aggregate(Pulse ~ Exer, data=x, mean)
points(1:nrow(means), means$Pulse, pch=16)
```



The groups look to have symmetric distributions with similar spreads, so one-way ANOVA seems justifiable. The groups' sample means have been drawn in on the boxplot, because they will play an important role in the method.

Our null hypothesis will be that the *groups all have the same population mean*.[6] If this hypothesis were true, then the *F-statistic* has an

[6] The different categories of the categorical variable split the full population up into k subpopulations. Each of these subpopulations has its own population mean. Our null hypothesis is that all of these subpopulations' means are equal (which also implies that they're all equal to the overall population mean).

$F_{k-1,n-k}$ distribution.

$$F := \frac{\text{SSG}/(k-1)}{\text{SSR}/(n-k)} \tag{9.1}$$

where SSG is the “group sum of squares" and SSR is the “residual sum of squares." Each data point contributes to both SSG and SSR. Its SSG contribution is the squared difference between its group’s sample mean and the overall sample mean; its SSR contribution is the squared difference between the the point and its group’s sample mean. Add up the contributions of all the observations to get the sums: SSG and SSR.

```
# Calculate the group sum of squares
sum.squares.group <- function(v) {
  ss <- length(v)*(mean(v) - m)^2
  return(ss)
}
SSGs <- aggregate(Pulse ~ Exer, data=x, sum.squares.group)
SSGs

##   Exer  Pulse
## 1 None 116.13
## 2 Some 331.77
## 3 Freq 452.56

SSG <- sum(SSGs$Pulse)
SSG

## [1] 900.47

# Calculate the residual sum of squares
SSRs <- aggregate(Pulse ~ Exer, data=x, sum.squares.res)
SSRs

##   Exer   Pulse
## 1 None  3201.1
## 2 Some 10760.2
## 3 Freq 11226.9

SSR <- sum(SSRs$Pulse)
SSR

## [1] 25188
```

Let’s go ahead and calculate the value of the F-statistic.

```
num <- SSG/(k-1)
denom <- SSR/(n-k)
f <- num/denom
f

## [1] 3.3783
```

Take another look at the definition of the F-statistic in equation (**??**); it is proportional to SSG. The larger SSG is, the further away the group sample means are from the overall sample mean, overall. So it makes sense that *large* values of F provide evidence against the null hypothesis that the group means are equal. The significance probability for this null hypothesis is equal to the probability that an $F_{k-1,n-k}$ random variable would be at least as large as the observed F value, which is one minus the appropriate cdf evaluated at F.

```
p <- 1-pf(f, df1=k-1, df2=n-k)
p

## [1] 0.036176
```

The built-in `aov` function can also be used for ANOVA. In fact, so can the `lm` function you already seen in Chapters 5 and 8.

```
a <- aov(Pulse ~ Exer, data=x)
a

## Call:
##    aov(formula = Pulse ~ Exer, data = x)
##
## Terms:
##                     Exer Residuals
## Sum of Squares    900.5   25188.2
## Deg. of Freedom       2       189
##
## Residual standard error: 11.544
## Estimated effects may be unbalanced

summary(a)

##              Df Sum Sq Mean Sq F value Pr(>F)
## Exer          2    900     450    3.38  0.036 *
## Residuals   189  25188     133
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

fit <- lm(Pulse ~ Exer, data=x)
summary(fit)

##
## Call:
## lm(formula = Pulse ~ Exer, data = x)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -41.19  -7.97  -0.19   7.81  32.03
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   76.765      2.800   27.42   <2e-16 ***
## ExerSome      -0.577      3.083   -0.19     0.85
## ExerFreq      -4.796      3.040   -1.58     0.12
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.5 on 189 degrees of freedom
## Multiple R-squared:  0.0345,Adjusted R-squared:  0.0243
## F-statistic: 3.38 on 2 and 189 DF,  p-value: 0.0362
```

9.1.3 *Confidence Intervals*

Returning to the two-group case, we'll construct a 95% confidence interval for the difference between the two genders' average pulses (female minus male). Because the sample sizes were large, the standardized variable is close enough to standard Normal that we can just use 2 standard errors as our margin of error.

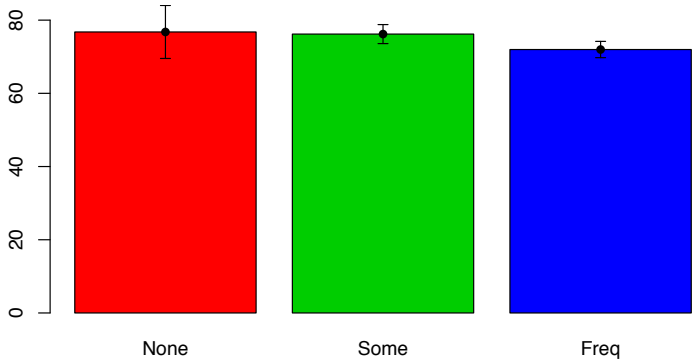
```
c(d - 2*SE, d + 2*SE)

## [1] -1.6056  5.1846
```

Error bars on plots can give an indication of the uncertainty of estimates. For instance, in plot below,[7] we've placed error bars giving approximate 95% confidence intervals for each subpopulation's mean, calculated separately according to the method in Section 8.1.3. Often, data analysts draw error bars extending out one standard error in each direction. Whenever you draw error bars, you should explain to your readers what they indicate.

[7] You'll need to run `install.packages("Hmisc")` the first time you want to use the `Hmisc` library demonstrated in the code below.

```
library(Hmisc)

## Loading required package:  grid
## Loading required package:  lattice
## Loading required package:  survival
## Loading required package:  Formula
## Loading required package:  ggplot2
##
## Attaching package:  'Hmisc'
##
## The following objects are masked from 'package:base':
##
##    format.pval, round.POSIXt, trunc.POSIXt, units

ns <- table(x$Exer)
SEs <- aggregate(Pulse ~ Exer, data=x, sd)$Pulse/sqrt(ns)
qs <- qt(.975, df=ns)
barplot(means$Pulse, names.arg=means$Exer, col=2:4,
        ylim=c(0, max(means$Pulse + qs*SEs)))
```

```
errbar((1:3)*1.2-.5, means$Pulse, means$Pulse + qs*SEs, means$Pulse - qs*
       add=TRUE)
```



People often try to draw conclusions about whether or not two unknown quantities are *significantly different* based on whether or not their error bars overlap. That practice is roughly accurate sometimes, but you should be careful about taking such conclusions too seriously.

9.2 Two Categorical Variables and One Quantitative Variable

two-way ANOVA

9.2.1 Estimation

9.2.2 Hypothesis Testing

9.3 One Categorical Variable and Two Quantitative Variables

In this section, we will revisit the gas mileage data from our coverage of Description in Section 6.3. This time we will take a more formal approach to the question of whether or not the different groups should get different fit lines.

As in Section 8.2, the key concept here will be the Linear Model; in this case, we'll have one quantitative response variable, along with a categorical and a quantitative explanatory variable. It is important to realize that there is actually no limit to how many variables (and which types of variables[8]) can be used in a Linear Model, making it a remarkably versatile method. In fact, one-way ANOVA and two-way ANOVA covered earlier in this chapter, are simply special cases of Linear Models. I consider the Linear Model to be the workhorse of data analysis.

In the case at hand, we will

[8] The cases covered in this book have all used a quantitative response variable. A categorical response variable is handled a little differently.

9.3.1 *Estimation*

linear model coefficients

9.3.2 *Hypothesis Testing*

9.4 *Conclusion*

	Inference
One Categorical and One Quantitative Variable	difference between two population means (E, H, C) one-way ANOVA (E, H)
Two Categorical and One Quantitative Variables	two-way ANOVA (E, H)
One Categorical and Two Quantitative Variables	linear model coefficients (E, H)

CONCLUDES OUR TOUR of different mixes of variable types. ... In Part IV, ...

Part IV

Additional Thoughts

10 *Writing a Report*

Writing a Report

11 *Statistical Reasoning*

Statistical Reasoning