A Second Semester Statistics Course with R

Mark Greenwood and Katherine Banner

2017-06-25

Contents

Acknowledgments

We would like to thank all the students and instructors who have provided input in the development of the current version of STAT 217 and that have impacted the choice of topics and how we try to teach them. Dr. Robison-Cox initially developed this course using R and much of this work retains his initial ideas. Many years of teaching these topics and helping researchers use these topics has helped to refine how they are presented here. Observing students years after the course has also helped to refine what we try to teach in the course, trying to prepare these students for the next levels of statistics courses that they might encounter and the next class where they might need or want to use statistics.

I (Greenwood) have intentionally taken a first person perspective at times to be able to include stories from some of those interactions to try to help you avoid some of their pitfalls in your current or future usage of statistics. I would like to thank my wife, Teresa Greenwood, for allowing me the time and support to work on this. I would also like to acknowledge Dr. Gordon Bril (Luther College) who introduced me to statistics while I was an undergraduate and Dr. Snehalata Huzurbazar (University of Wyoming) that guided me to completing my Master's and Ph.D. in Statistics and still serves as a valued mentor and friend to me.

The development of this text was initially supported with funding from Montana State University's Instructional Innovation Grant Program with a grant titled Towards more active learning in STAT 217. This book was born with the goal of having a targeted presentation of topics that we cover (and few that we don't) that minimizes cost to students and incorporates the statistical software R from day one and every day after that. The software is a free, open-source platform and so is dynamically changing over time. This has necessitated frequent revisions of the text.

This is Version 3.01 of the book. It fixes a problem created with the digital links in the book that occurred during Spring 2017. Version 3.0 of the book, prepared for Fall 2016, involved edits, a couple of partially new sections, and updated R code along with a new format for how the R code is displayed to more easily distinguish it from other text. Each revision has involved a similar amount of change with Version 2.0 published in January 2015 and Version 1.0 in January 2014 after using draft chapters that were initially developed during Fall 2013.

We have made every attempt to keep costs as low as possible by making it possible for most pages to be printed in black and white. The text (in full color and with dynamic links) is also available as a free digital download from Montana State University's ScholarWorks repository at https://scholarworks.montana.edu/xmlui/handle/1/2999.

Enjoy your journey from introductory to intermediate statistics!

Chapter 1

Preface

This book is designed primarily for use in a second semester statistics course although it can also be useful for researchers needing a quick review or ideas for using R for the methods discussed in the text. As a text primarily designed for a second statistics course, it presumes that you have had an introductory statistics course. There are now many different varieties of introductory statistics from traditional, formula-based courses (called "consensus" curriculum courses) to more modern, computing-intensity courses that use randomization ideas to try to enhance learning of basic statistical methods. We are not going to presume that you have had a particular "flavor" of introductory statistics or that you had your introductory statistics out of a particular text, just that you have had a course that tried to introduce you to the basic terminology and ideas underpinning statistical reasoning. We would expect that you are familiar with the logic (or sometimes illogic) of hypothesis testing including null and alternative hypothesis and confidence interval construction and interpretation and that you have seen all of this in a couple of basic situations. We start with a review of these ideas in one and two group situations with a quantitative response, something that you should have seen before.

This text covers a wide array of statistical tools that are connected through situation, methods used, or both. As we explore various techniques, look for the identifying characteristics of each method – what type of research questions are being addressed (relationships or group differences, for example) and what type of variables are being analyzed (quantitative or categorical). ***Quantitative variables*** are made up of numerical measurements that have meaningful units attached to them. ***Categorical variables*** take on values that are categories or labels. Additionally, you will need to carefully identify the ***response*** variables, where the study and variable characteristics should suggest which variables should be used as the explanatory variables that may explain variation in the response variable. Because this is an intermediate statistics course, we will start to handle more complex situations (many explanatory variables) and will provide some tools for graphical explorations to complement the more sophisticated statistical models required to handle these situations.

1.1 Overview of methods

After you are introduced to basic statistical ideas, a wide array of statistical methods become available. The methods explored here focus on assessing (estimating and testing for) relationships between variables, sometimes when controlling for or modifying relationships based on levels of another variable – which is where statistics gets interesting and really useful. Early statistical analyses (approximately 100 years ago) were focused on describing a single variable. Your introductory statistics course should have heavily explored methods for summarizing and doing inference in situations with one group or where you were comparing results for two groups of observations. Now, we get to consider more complicated situations – culminating in a set of tools for working with multiple explanatory variables, some of which might be categorical and related to having different groups of subjects that are being compared. Throughout the methods we will cover, it will

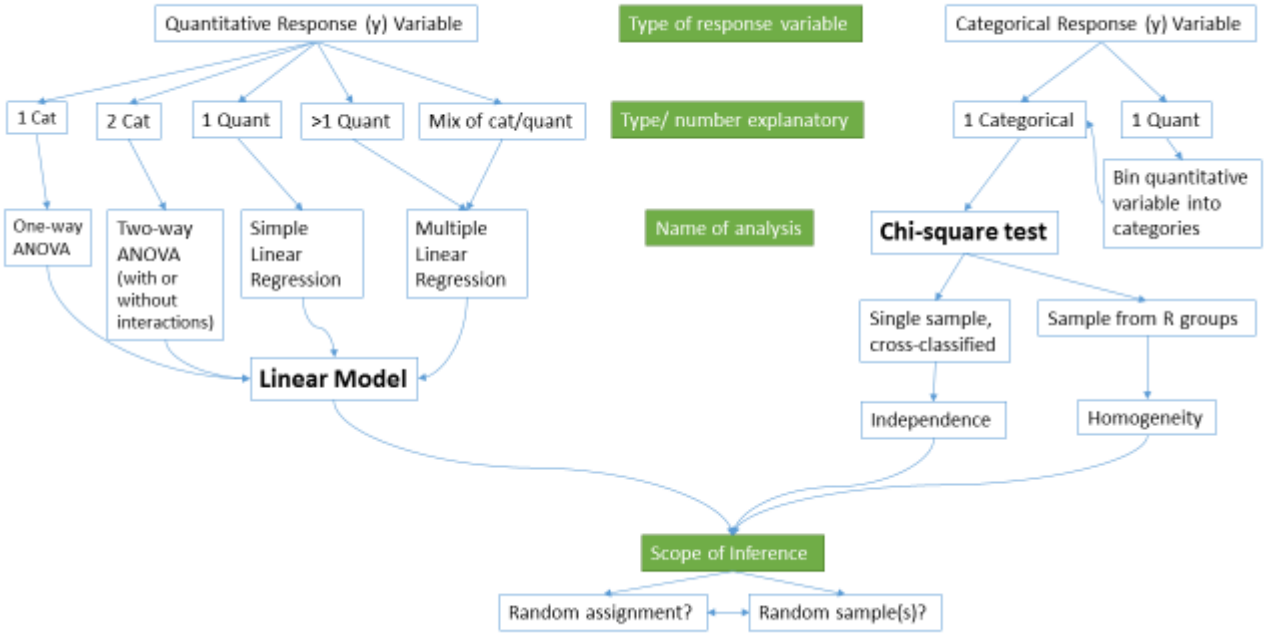


Figure 1.1: Flow chart of methods

be important to retain a focus on how the appropriate statistical analysis depends on the research question and data collection process as well as the types of variables measured.

Figure 1.1 frames the topics we will discuss. Taking a broad vision of the methods we will consider, there are basically two scenarios – one when the response is quantitative and one when the response is categorical. Examples of quantitative responses we will see later involve *suggested jail sentence* (in years) and *body fat* (percentage). Examples of categorical variables include *improvement* (none, some, or marked) in a clinical trial or whether a student has turned in copied work (never, exam or paper, or both). There are going to be some more nuanced aspects to all these analyses as the complexity of both sides of Figure 1.1 suggest, but note that near the bottom, each tree converges on a single procedure, using a ***linear model*** for a quantitative response variable or using a ***Chi-square test*** for a categorical response. After selecting the appropriate procedure and completing the necessary technical steps to get results for a given data set, the final step involves assessing the scope of inference and types of conclusions that are appropriate based on the design of the study.

We will be spending most of the semester working on methods for quantitative response variables (the left side of Figure 1.1 covered in Chapters 1, 2, 3, 5, 6, and 7) and stepping over to handle the situation with a categorical response variable (right side of Figure 1.1 that is discussed in Chapter 4). Chapter 8 contains case studies illustrating all the methods discussed previously, providing a final opportunity to explore additional examples that illustrate how finding your way through the paths in Figure 1.1 leads to the appropriate analysis.

The first topics (Chapters 0 and 1) will be more familiar as we start with single and two group situations with a quantitative response. In your previous statistics course, you should have seen methods for estimating and quantifying uncertainty for the mean of a single group and for differences in the means of two groups. Once we have briefly reviewed these methods and introduced the statistical software that we will use throughout the course, we will consider the first new statistical material in Chapter 2. It involves the situation with a quantitative response variable where there are more than 2 groups to compare – this is what we call the ***One-Way ANOVA*** situation. It generalizes the 2-independent sample hypothesis test to handle situations

where more than 2 groups are being studied. When we learn this method, we will begin discussing model assumptions and methods for assessing those assumptions that will be present in every analysis involving a quantitative response. The ***Two-Way ANOVA*** (Chapter 3) considers situations with two categorical explanatory variables and a quantitative response. To make this somewhat concrete, suppose we are interested in assessing differences in, say, the *yield* of wheat from a field based on the amount of *fertilizer* applied(none, low, or high) and *variety* of wheat (two types). Here, *yield* is a quantitative response variable that might be measured in bushels per acre and there are two categorical explanatory variables, *fertilizer*, with 3 levels, and *variety*, with two levels. In this material, we introduce the idea of an ***interaction*** between the two explanatory variables: the relationship between one categorical variable and the mean of the response changes depending on the levels of the other categorical variable. For example, extra fertilizer might enhance the growth of one variety and hinder the growth of another so we would say that *fertilizer* has different impacts based this interaction may or may not actually be present, we will consider two versions of the model in Two-Way ANOVAs, what are called the ***additive*** (no interaction) and the ***interaction*** models.

Following the methods for two categorical variables and a quantitative response, we explore a method for analyzing data where the response is categorical, called the ***Chi-square test*** in Chapter 4. This most closely matches the One-Way ANOVA situation with a single categorical explanatory variable, except now the response variable is categorical. For example, we will assess whether taking a drug (vs taking a ***placebo***[1]) has an ***effect***[2] on the type of improvement the subjects demonstrate. There are two different scenarios for study design that impact the analysis technique and hypotheses tested in Chapter 4. If the explanatory variable reflects the group that subjects were obtained from, either through randomization of the treatment level to the subjects or by taking samples from separate populations, this is called a ***Chi-square Homogeneity Test***. It is also possible to obtain a single sample from a population and then obtain information on the levels of the explanatory variable for each subject. We will analyze these results using what is called a ***Chi-square Independence Test***. They both use the same test statistic but we use slightly different graphics and are testing different hypotheses in these two related situations. Figure 1.1 also shows that if we had a quantitative explanatory variable and a categorical response that we would need to "bin" or create categories of responses from the quantitative variable to use the Chi-square testing methods.

If the predictor and response variables are both quantitative, we start with scatterplots, correlation, and ***simple linear regression*** models (Chapters 5 and 6) – things you should have seen, at least to some degree, previously. The biggest differences here will be the depth of exploration of diagnostics and inferences for this model and discussions of transformations of variables. If there is more than one explanatory variable, then we say that we are doing ***multiple linear regression*** (Chapter 7) – the "multiple" part of the name reflects that there will be more than one explanatory variable. We use the same name if we have a mix of categorical and quantitative predictor variables but there are some new issues in setting up the models and interpreting the coefficients that we need to consider. In the situation with one categorical predictor and one quantitative predictor, we revisit the idea of an interaction. It allows us to consider situations where the estimated relationship between a quantitative predictor and the mean response varies among different levels of the categorical variable.

By the end of Chapter 8 you should be able to identify, perform using the statistical software R (R Core Team, 2016), and interpret the results from each of these methods. There is a lot to learn, but many of the tools for using R and interpreting results of the analyses accumulate and repeat during the semester. If you work hard to understand the initial methods, it will help you when the methods get more complicated. You will likely feel like you are just starting to learn how to use R at the end of the semester and for learning a new language that is actually an accomplishment. We will just be taking you on the first steps of a potentially long journey and it is up to you to decide how much further you want to go with learning the software.

All the methods you will learn require you to carefully consider how the data were collected, how that pertains to the population of interest, and how that impacts the inferences that can be made. The ***scope of***

[1] A ***placebo*** is a treatment level designed to mimic the potentially efficacious level(s) but that can have no actual effect. The ***placebo effect*** is the effect that thinking that an effective treatment was received has on subjects. There are other related issues in performing experiments like the ***Hawthorne*** or observer ***effect*** where subjects modify behavior because they are being observed.

[2] We will reserve the term "effect" for situations where we could potentially infer causal impacts on the response of the explanatory variable which occurs in situations where the levels of the explanatory variable are randomly assigned to the subjects.

inference from the bottom of Figure 1.1 is our shorthand term for remembering to think about two aspects of the study – ***random assignment*** and ***random sampling*** . In a given situation, you need to use the description of the study to decide if the explanatory variable was randomly assigned to study units (this allows for ***causal inferences*** if differences are detected) or not (so no causal statements are possible). As an example, think about two studies, one where students are randomly assigned to either get tutoring with their statistics course or not and another where the students are asked at the end of the semester whether they sought out tutoring or not. Suppose we compare the final grades in the course for the two groups (tutoring/not) and find a big difference. In the first study with random assignment, we can say the tutoring caused the differences we observed. In the second, we could only say that the tutoring was associated with differences but because students self-selected the group they ended up in, we can't say that the tutoring caused the differences. The other aspect of scope of inference concerns random sampling: If the data were obtained using a random sampling mechanism, then our inferences can be safely extended to the population that the sample was taken from. However, if we have non-random sample, our inference can only apply to the sample collected. In the previous example, the difference would be studying a random sample of students from the population of, say, Introductory Statistics students at a university vs studying a sample of students that volunteered for the research project, maybe for extra credit in the class. We could still randomly assign them to tutoring/not but the non-random sample would only lead to conclusions about those students that volunteered. The most powerful scope of inference is when there are randomly assigned levels of explanatory variables with a random sample from population – conclusions would be about causal impacts that would happen in the population.

By the end of this material, you should have some basic R skills and abilities to create basic ANOVA and Regression models, as well as to handle Chi-squared testing situations. Together, this should prepare you for future statistics courses or for other situations where you are expected to be able to identify an appropriate analysis, do the calculations for a given data set, and then effectively communicate interpretations for the methods discussed here.

1.2 Getting started in R

You will need to download the statistical software package called R and an enhanced interface to R called RStudio (RStudio, 2016). They are open source and free to download and use (and will always be that way). This means that the skills you learn now can follow you the rest of your life. R is becoming the primary language of statistics and is being adopted across academia, government, and businesses to help manage and learn from the growing volume of data being obtained. Hopefully you will get a sense of some of the power of R in this book.

The next pages will walk you through the process of getting the software downloaded and provide you with an initial experience using RStudio to do things that should look familiar even though the interface will be a new experience. Do not expect to master R quickly – it takes years (sorry!) even if you know the statistical methods being used. We will try to keep all your interactions with R code in a similar code format and that should help you in learning how to use R as we move through various methods. We will also usually provide you with example code. Everyone that learns R starts with copying other people's code and then making changes for specific applications – so expect to go back to examples from the text and focus on learning how to modify that code to work for your particular data set. Only really experienced R users "know" functions without having to check other resources. After we complete this basic introduction, Chapter 1 begins doing more sophisticated things with R, allowing us to compare quantitative responses from two groups, make some graphical displays, do hypothesis testing and create confidence intervals in a couple of different ways.

You will have two downloading activities to complete before you can do anything more than read this book[3]. First, you need to download R. It is the engine that will do all the computing for us, but you will only interact with it once. Go to http://cran.rstudio.com and click on the "**Download R for...**" button that

[3]I recorded a video that walks through the material on the following pages that is available here: https://camtasia.msu.montana.edu/Relay/Files/w76c139/RandRstudio_Final/RandRstudio_Final_-_20160715_130555_23.html in the digital version of the book.

corresponds to your operating system. On the next page, click on "**base**" and then it will take you to a screen to download the most current version of R that is compiled for your operating system, something like "**Download R 3.3.1 for Windows**". Click on that link and then open the file you downloaded. You will need to select your preferred language (choose English so your instructor can help you), then hit "**Next**" until it starts to unpack and install the program (all the base settings will be fine). After you hit "**Finish**" you will not do anything further with R directly.

Second, you need to download RStudio. It is an enhanced interface that will make interacting with R less frustrating. To download RStudio, go to http://www.rstudio.com/products/rstudio/download/ and select the correct version under "Installers for Supported Platforms" for your operating system. Download and then install RStudio using the installer. From this point forward, you should only open RStudio; it provides your interface with R. Note that both R and RStudio are updated frequently (up to four times a year) and if you downloaded either more than a few months previously, you should download the up-to-date versions, especially if something you are trying to do is not working. Sometimes code will not work in older versions of R and sometimes old code won't work in new versions of R.[4]

To get started, we can complete some basic tasks in R using the RStudio interface. When you open RStudio, you will see a screen like Figure 2. The added annotation in this and the following screen-grabs is there to help you get initially oriented to the software interface. R is command-line software – meaning that most of the time you have to create code and then enter and execute it at a command prompt to get any results. RStudio makes the management and execution of that code more efficient than the basic version of R. In RStudio, the lower left panel is called the "console" window and is where you can type R code directly into R or where you will see the code you run and (most importantly!) where the results of your executed commands will show up. The most basic interaction with R is available once you get the cursor active at the command prompt ">" by clicking in that panel (look for a blinking vertical line). The upper left panel is for writing, saving, and running your R code. Once you have code available in this window, the "Run" button will execute the code for the line that your cursor is on or for any text that you have highlighted with your mouse. The "data management" or environment panel is in the upper right, providing information on what data sets have been loaded. It also contains the "Import Dataset" button that provides the easiest way for you to read a data set into R so you can analyze it. The lower right panel contains information on the "Packages" (additional code we will download and install to add functionality to R) that are available and is where you will see plots that you make and requests for "Help" on specific functions.

As a first interaction with R we can use it as a calculator. To do this, click near the command prompt (>) in the lower left "console" panel, type 3+4, and then hit enter. It should look like this:

```
> 3+4
[1] 7
```

You can do more interesting calculations, like finding the mean of the numbers 3, 5, 7, and 8 by adding them up and dividing by 4:

```
> (-3+5+7+8)/4
[1] 4. 25
```

Note that the parentheses help R to figure out your desired order of operations. If you drop that grouping, you get a very different (and wrong!) result:

```
> -3+5+7+8/4
[1] 11
```

We could estimate the standard deviation similarly using the formula you might remember from introductory statistics, but that will only work in very limited situations. To use the real power of R this semester, we need to work with data sets that store the Basically, we need to store observations in named vectors (one dimensional arrays) that contain a list of the observations. To create a vector containing the four numbers and assign it to a variable named *variable1*, we need to create a vector using the function `c` which means

[4]The need to keep the code up-to-date as R continues to evolve is one reason that this book is locally published and that this is the 3^{rd} version in three years...

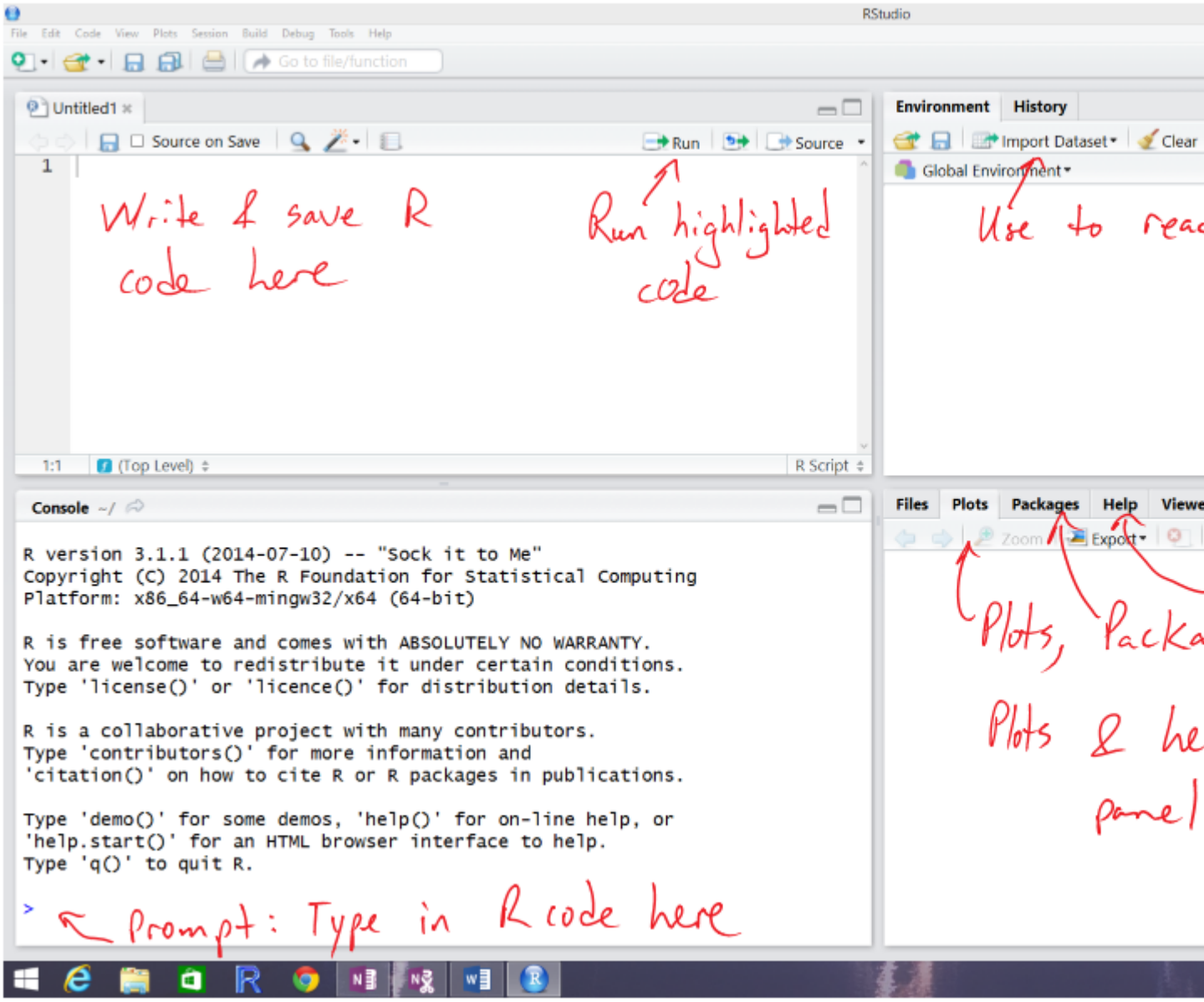


Figure 1.2: Initial RStudio layout

"combine the items" that follow, if they are inside parentheses and have commas separating the values, as follows:

```
> c(-3, 5, 7, 8)
[1] -3 5 7 8
```

To get this vector stored in a variable called *variable1* we need to use the assignment operator, `<-` (read as "stored as") that assigns the information on the right into the variable that you are creating on the left.

```
> variable1 <- c(-3, 5, 7, 8)
```

In R, the assignment operator, `<-`, is created by typing a "less than" symbol `<` followed by a "minus" sign (`-`) ever want to see what numbers are residing in an object in R, just type its name and hit *enter*. You can see how that variable contains the same information that was initially generated by `c(-3, 5, 7, 8)` but is easier to access since we just need the text for the variable name representing that vector.

```
> variable1
[1] -3 5 7 8
```

With the data stored in a variable, wean use functions such as `mean` and `sd` to find the mean and standard deviation of the observations contained in `variable1` :

```
> mean(variable1)
[1] 4.25
> sd(variable1)
[1] 4.99166
```

When dealing with real data, we will often have information about more than one variable. We could enter all observations by hand for each variable but this is prone to error and onerous for all but the smallest data sets. If you are to ever utilize the power of statistics in the evolving data-centered world, data management has to be accomplished in a more sophisticated way. While you can manage data sets quite effectively in R, it is often easiest to start with your data set in something like Microsoft Excel or OpenOffice's Calc. You want to make sure that observations are in the rows and the names of variables are in the columns and that there is no "extra stuff" in the spreadsheet. If you have missing observations, they should be represented with blank cells. The file should be saved as a ".csv" file (stands for comma-separated values although Excel calls it "CSV (Comma Delimited)", which basically strips off some of the junk that Excel adds to the necessary information in the file. Excel will tell you that this is a bad idea, but it actually creates a more stable archival format and one that R can use directly[5].

With data set converted to a CSV file, we need to read the data set into R. There are two ways to do this, either using the point-and-click GUI in RStudio (click the "Import Data Set" button in the upper right "Environment" panel as indicated in Figure 1.2) or modifying the

`read.csv` function to find the file of interest. To practice this, you can download an Excel (.xls) file from http://www.math.montana.edu/courses/s217/documents/treadmill.xls 31 males that volunteered for a study on methods for measuring fitness(Westfall and Young, 1993). In the spreadsheet, you will find a data set that starts and ends with the following information (only results for Subjects 1, 2, 30, and 31 shown here):

Sub-ject	Tread-MillOx	TreadMill-MaxPulse	RunTime	RunPulse	Rest Pulse	BodyWeight	Age
1	60.05	186	8.63	170	48	81.87	38
2	59.57	172	8.17	166	40	68.15	42
...	...	...	...	...	...	...	...
30	39.2	172	12.88	168	44	91.63	54
31	37.39	192	14.03	186	56	87.66	45

[5]There are ways to read ".xls" and ".xlsx" files directly into R but to handle multiple sheets they are more complicated and not as stable across operating systems as the simpler version we recommend.

The variables contain information on the subject number (*Subject*), subjects' treadmill oxygen consumption (*TreadMillOx*, in ml per kg per minute) and maximum pulse rate (*TreadMillMaxPulse*, in beats per minute), time to run 1.5 miles (*Run Time*, in minutes), maximum pulse during 1.5 mile run (*RunPulse*, in beats per minute), resting pulse rate (*RestPulse*, beats per minute), Body Weight (*BodyWeight*, in kg), and *Age* (in years). Open the file in Excel or equivalent software and then save it as a .csv file in a location you can find on your computer. Then go to RStudio and click on **File** , then **Import Dataset** , then **From CSV...**[6] Find your file and check "**Import**". R will store the data set as an object named whatever the .csv file was named. You could use another name as well, but it is often easiest just to keep the data set name in R related to the original file name. You should see some text appear in the console (lower left panel) like in Figure 1.3. The text that is created will look something like the following – if you had stored the file in a drive labeled D:, it would be:

```
treadmill <- read.csv("D:/treadmill.csv")
```

What is put inside the `" "` will depend on the location and name of your saved .csv file. A version of the data set in what looks like a spreadsheet will appear in the upper left window due to the second line of code (`View(treadmill)`).

Just directly typing (or using) a line of code like this is actually the other way that we can read in files. If you choose to use the text-only interface, then you need to tell R where to look in your computer to find the data file. `read.csv` is a function that takes a path as an argument. To use it, specify the path to your data file, put quotes around it, and put it as the input to `read.csv(...)` . For some examples later in the book, you will be able to copy a command like this from the text and read data sets and other code directly from my the course folder, assuming you are connected to the internet.

To verify that you read the data set in correctly, it is always good to check its contents. We can view the first and last rows in the data set using the `head` and `tail` functions on the data set, which show the following results for the `treadmill` data. Note that you will sometimes need to resize the console window in RStudio to get all the columns to display in a single row which can be performed by dragging the gray bars that separate the panels.

```
head(treadmill)
```

```
##   Subject TreadMillOx TreadMillMaxPulse RunTime RunPulse RestPulse
## 1       1       60.05               186    8.63      170        48
## 2       2       59.57               172    8.17      166        40
## 3       3       54.62               155    8.92      146        48
## 4       4       54.30               168    8.65      156        45
## 5       5       51.85               170   10.33      166        50
## 6       6       50.55               155    9.93      148        49
##   BodyWeight Age
## 1      81.87  38
## 2      68.15  42
## 3      70.87  50
## 4      85.84  44
## 5      83.12  54
## 6      59.08  57
```

```
tail(treadmill)
```

```
##    Subject TreadMillOx TreadMillMaxPulse RunTime RunPulse RestPulse
## 26      26       44.61               182   11.37      178        62
## 27      27       40.84               172   10.95      168        57
## 28      28       39.44               176   13.08      174        63
## 29      29       39.41               176   12.63      174        58
```

[6] If you are having trouble getting the file converted and read into R, copy and run the following code: `treadmill <-read.csv("http://www.math.montana.edu/courses/s217/documents/treadmill.csv", header=T)`.

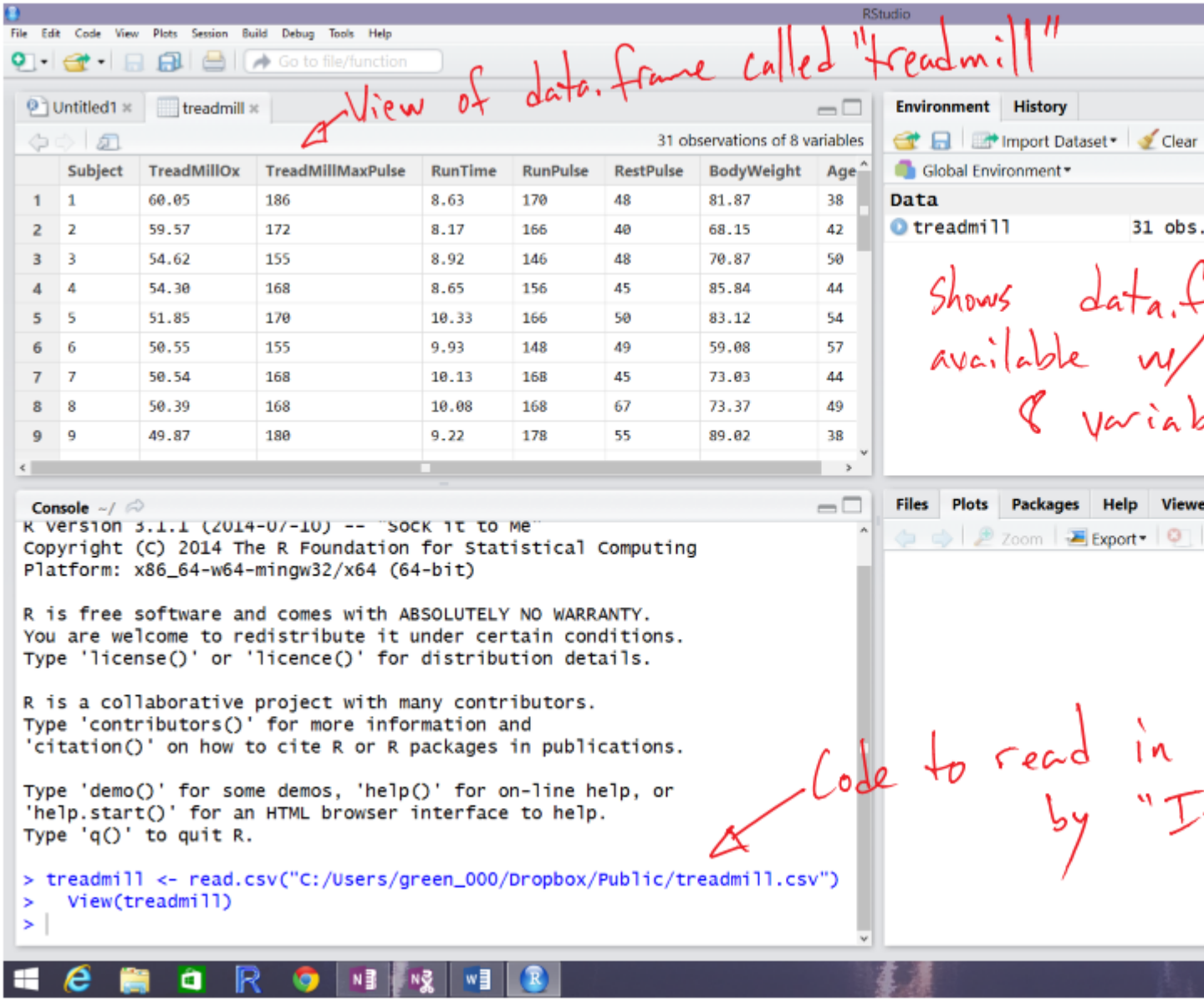


Figure 1.3: RStudio with initial data set loaded

```
## 30      30       39.20              172   12.88      168        44
## 31      31       37.39              192   14.03      186        56
##    BodyWeight Age
## 26      89.47  44
## 27      69.63  51
## 28      81.42  44
## 29      73.37  57
## 30      91.63  54
## 31      87.66  45
```

While not always required, for many of the analyses, we will tap into a large suite of additional functions available in R packages by "installing" (basically downloading) and then "loading" the packages. There are some packages that we will use frequently, starting with the `mosaic` package (Pruim, Kaplan, and Horton, tab in the lower right panel of RStudio. Click on the **Install** button and then type in the name of the package in the box (here type in `mosaic`). RStudio will try to auto-complete the package name you are typing which should help you make sure you got it typed correctly. This will be the first of *many* times that we will mention that R is case sensitive – in other words, `Mosaic` is different from `mosaic` in R syntax and this sort of thing applies to everything you do in R. You should only need to install each R package once on a given computer. If you ever see a message that R can't find a package, make sure it appears in the list in the **Packages** tab and if it doesn't, repeat the previous steps to install it.

After installing the package, we need to load it to make it active in a given work session. Go to the command prompt and type (or copy and paste) `require(mosaic)` :

```
require(mosaic)
```

You may see a warning message about versions of the package and versions of R – this is *usually* something you can ignore. Other warning messages could be more ominous for proceeding but before getting too concerned, there are couple of basic things to check. First, double check that the package is installed (see previous steps). Second, check for typographical errors in your code – especially for mis-spellings or unintended capitalization. If you are still having issues, try repeating the installation process. If that fails, find someone more used to using R to help you (for example in the Math Learning Center or by emailing your instructor)[7].

To help you go from basic to intermediate R usage and especially to help with more complicated problems, you will want to learn how to manage and save your R code. The best way to do this is using the upper left panel in RStudio using what are called R Scripts, which are files that have a file extension of ".R". To start a new ".R"" file to store your code, click on **File** , then **New File**, then **R Script*. This will create a blank page to enter and edit code – then save the file as MyFileName.R in your preferred location. Saving your code will mean that you can return to where you last were working by simply re-running the saved script file. With code in the script window, you can place the cursor on a line of code or highlight a chunk of code and hit the "Run" button on the upper part of the panel. It will appear in the console with results just like what you would obtain if you typed it after the command prompt and hit enter for each line.

Figure 1.4 shows the screen with the code used in this section in the upper left panel, saved in file called Ch0.R, with the results of highlighting and executing the first section of code using the "Run" button.

1.3 Basic summary statistics, histograms, and boxplots using R

With RStudio running, the `mosaic` package loaded, a place to write and save code, and the `treadmill` data set loaded, we can (finally!) start to summarize the results of the study. The `treadmill` object is what R calls a ***data.frame***[8] and contains columns corresponding to each variable in the spreadsheet. Every function

[7]Most computer lab computers at Montana State University have RStudio installed and so provide another venue to try this where the software is already installed.

[8]Data frames in R are objects that can contain both categorical and quantitative variables on your n subjects with a name for each variable that is also the name of each column in a matrix. Each subject is a row of the data set.

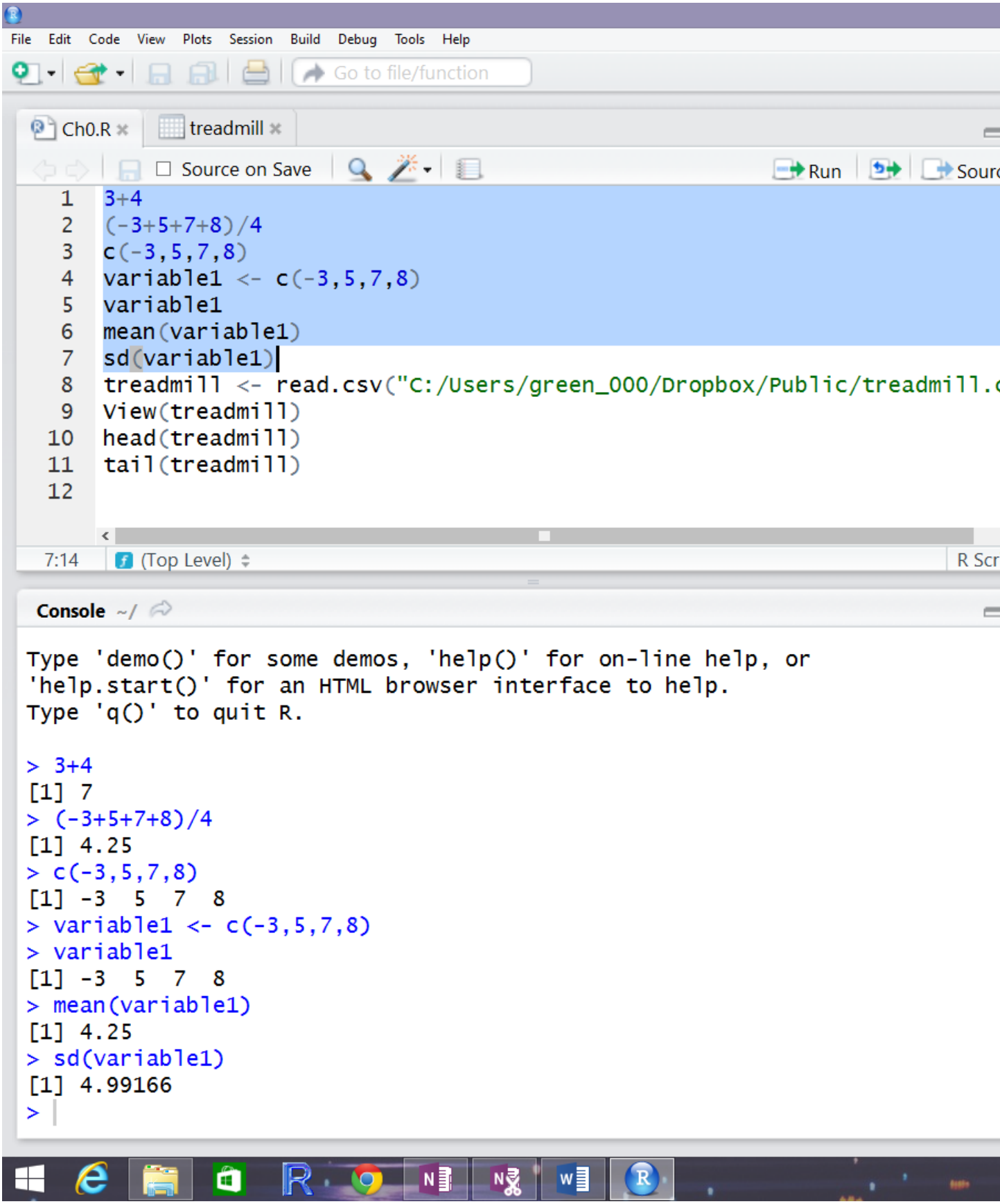


Figure 1.4: RStudio with highlighted code run

in R will involve specifying the variable(s) of interest and how you want to use them. To access a particular variable (column) in a data. frame, you can use a $ between the data. frame name and the name of the variable of interest, generically as `dataframename$variablename`. To identify the `RunTime` variable here it would be `treadmill$RunTime`. In the command line it would look like:

```
treadmill$RunTime
```

```
##  [1]  8.63  8.17  8.92  8.65 10.33  9.93 10.13 10.08  9.22  8.95 10.85
## [12]  9.40 11.50 10.50 10.60 10.25 10.00 11.17 10.47 11.95  9.63 10.07
## [23] 11.08 11.63 11.12 11.37 10.95 13.08 12.63 12.88 14.03
```

Just as in the previous section, we can generate summary statistics using functions like `mean` and `sd` by running them on a specific variable:

```
mean(treadmill$RunTime)
```

```
## [1] 10.58613
```

```
sd(treadmill$RunTime)
```

```
## [1] 1.387414
```

And now we know that the average running time for 1.5 miles for the subjects in the study was 10.6 minutes with a standard deviation (SD) of 1.39 minutes. But you should remember that the mean and SD are only appropriate summaries if the distribution is roughly ***symmetric*** (both sides of the distribution are approximately the same). The `mosaic` package provides a useful function called `favstats` that provides the mean and SD as well as the ***5 number summary*** : the minimum (`min`), the first quartile (`Q1`, the 25^{th} percentile), the median (50^{th} percentile), the third quartile (`Q3` , the 75^{th} percentile), and the maximum (`max`). It also provides the number of observations (`n`) which was 31, as noted above, and a count of whether any missing values were encountered (`missing`), which was 0 here since all subjects had measurements available on this variable.

```
favstats(treadmill$RunTime)
```

```
##   min   Q1 median    Q3   max     mean       sd  n missing
##  8.17 9.78  10.47 11.27 14.03 10.58613 1.387414 31       0
```

We are starting to get somewhere with understanding that the runners were somewhat fit with worst runner covering 1.5 miles in 14 minutes (the equivalent of a 9.3 minute mile) and the best running at a 5.4 minute mile pace. The limited variation in the results suggests that the sample was obtained from a restricted group with somewhat common characteristics. When you explore the ages and weights of the subjects in the Practice Problems in Section 0.5, you will get even more information about how similar all the subjects in this study were.

A graphical display of these results will help us to assess the shape of the distribution of run times – including considering the potential for the presence of a ***skew*** (whether the right or left tail of the distribution is noticeably more spread out with left skew meaning that the left tail is more spread out than the right tail) and ***outliers***
(unusual observations). A ***histogram*** is a good place to start. Histograms display connected bars with counts of observations defining the height of bars based on a set of bins of values of the quantitative variable. We will apply the `hist` function to the `RunTime` variable, which produces Figure 1.5.

```
hist(treadmill$RunTime)
```

I used the **Export** button found above the plot, followed by **Copy to Clipboard** and clicking on the **Copy Plot** button. Then if you open your favorite word-processing program, you should be able to paste it into a document for writing reports that include the figures. You can see the first parts of this process in the screen grab in Figure 1.6.

You can also directly save the figures as separate files using **Save as Image** or **Save as PDF**and then insert them into your word processing documents.

Figure 1.5: Histogram of Run Times #(minutes) of n=31 subjects in Treadmill study.

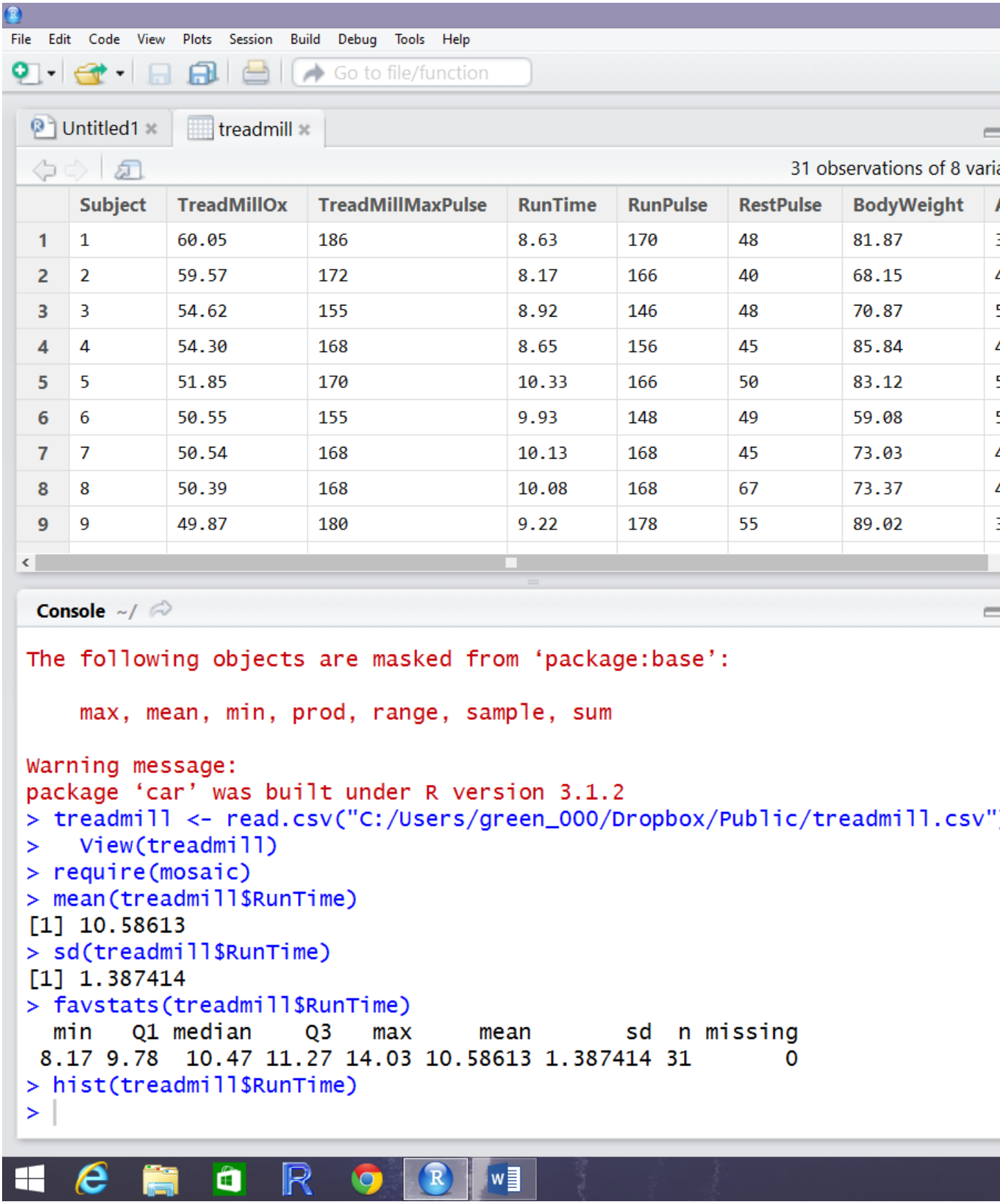


Figure 1.6: RStudio while in the process of copying the histogram

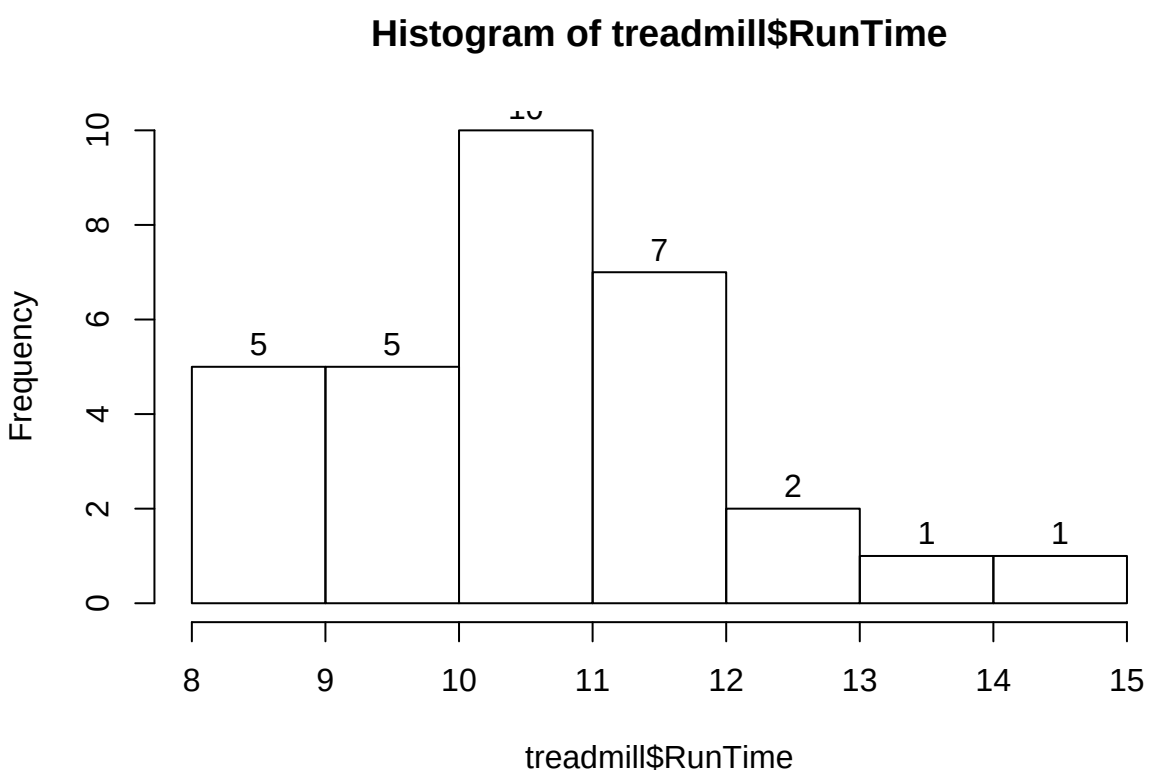


Figure 1.7: Histogram of #Run Times with counts in bars labeled.

The function `hist` defaults into providing a histogram on the ***frequency*** (count) scale. In most R functions, there are the default options that will occur if we don't make any specific choices but we can override the default options if we desire. One option we can modify here is to add labels to the bars to be able to see exactly how many observations fell into each bar. Specifically, we can turn the `labels` option "on" by making it true ("T") by adding `labels=T` to the previous call to the `hist` function, separated by a comma:

```
hist(treadmill$RunTime, labels=T)
```

Based on this histogram, it does not appear that there any outliers in the responses since there are no bars that are separated from the other observations. However, the distribution does not look symmetric and there might be a skew to the distribution. Specifically, it appears to be ***skewed right*** (the right tail is longer than the left). But histograms can sometimes mask features of the data set by binning observations and it is hard to find the percentiles accurately from the plot.

When assessing outliers and skew, the ***boxplot***
(or *Box and Whiskers* plot) can also be helpful (Figure 1.8) to describe the shape of the distribution as it displays the 5-number summary and will also indicate observations that are "far" above the middle of the observations. R's `boxplot` function uses the standard rule to indicate an observation as a ***potential outlier*** if it falls more than 1.5 times the ***IQR*** (Inter-Quartile Range, calculated as Q3-Q1) below Q1 or above Q3. The potential outliers are plotted with circles and the *Whiskers* (lines that extend from Q1 and Q3 typically to the minimum and maximum) are shortened to only go as far as observations that are within 1.5∗IQR of the upper and lower quartiles. The *box* part of the boxplot is a box that goes from Q1 to Q3 and the median is displayed as a line somewhere inside the box[9]. Looking back at the summary statistics above, Q1=9.78

[9]The median, quartiles and whiskers sometimes occur at the same values when there are many tied observations. If you can't

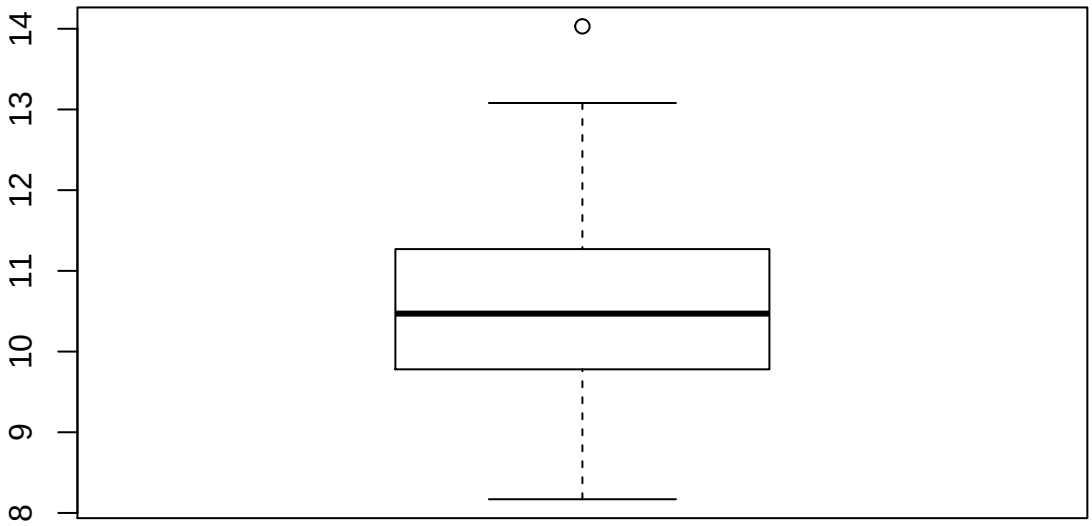


Figure 1.8: Boxplot of 1.5 mile Run Times.

and Q3=11.27, providing an IQR of:

```
IQR <- 11.27 - 9.78
```

One observation (the maximum value of 14.03) is indicated as a potential outlier based on this result by being larger than Q3 +1.5∗IQR, which was 13.505:

```
11.27 + 1.5*IQR
```

```
## [1] 13.505
```

The boxplot also shows a slight indication of a right skew (skew towards larger values) with the distance from the minimum to the median being smaller than the distance from the median to the maximum. Additionally, the distance from Q1 to the median is smaller than the distance from the median to Q3. It is modest skew, but worth noting.

```
boxplot(treadmill$RunTime)
```

While the default boxplot is fine, it fails to provide good graphical labels, especially on the y-axis. Additionally, there is no title on the plot. The following code provides some enhancements to the plot by using the `ylab` and `main` options in the call to `boxplot`, with the results displayed in Figure 1.9. When we add text to plots, it will be contained within quotes and be assigned into the options `ylab` (for y-axis) or `main` (for the title) here to put it into those locations.

see all the components of the boxplot, produce the numerical summary to help you understand what happened.

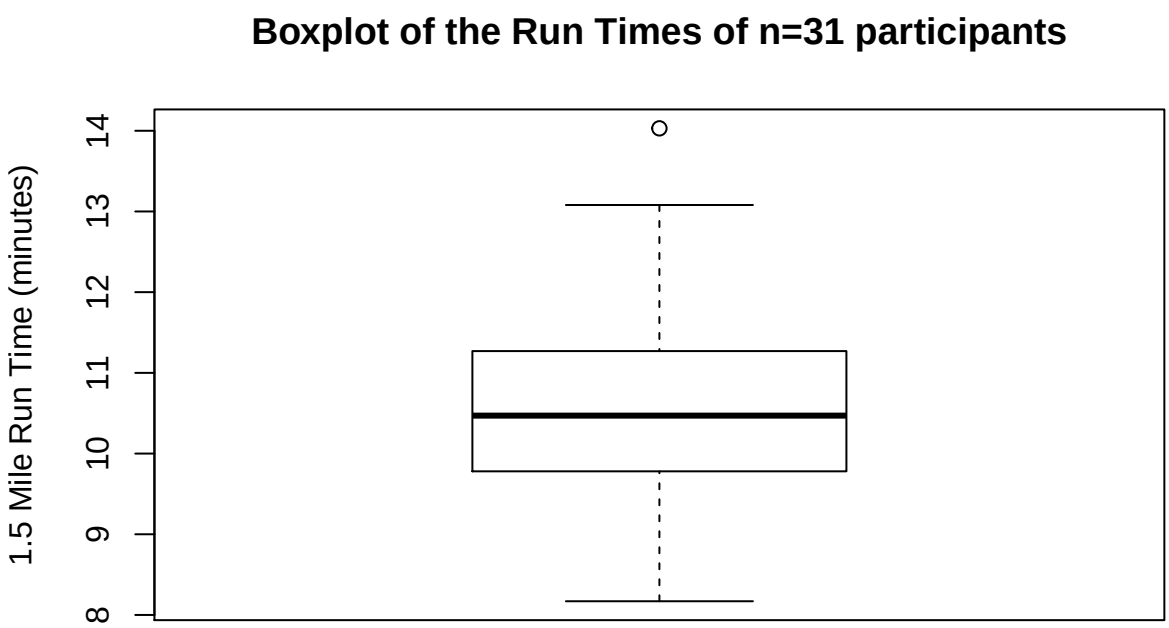


Figure 1.9: Boxplot of Run Times with improved labels.

```
boxplot(treadmill$RunTime, ylab="1.5 Mile Run Time (minutes)",
        main="Boxplot of the Run Times of n=31 participants")
```

Throughout the book, we will often use extra options to make figures that are easier for you to understand. There are often simpler versions of the functions that will suffice but the extra work to get better labeled figures is often worth it. I guess the point is that "a picture is worth a thousand words" but in data visualization, that is only true if the reader can understand what is being displayed. It is also important to think about the quality of the information that is being displayed, regardless of how pretty the graphic might be.

All the previous results were created by running the R code and then "grabbing" the results from either the console or by copying the figure. There is another way to use RStudio where you can have it compile the results (both output and figures) directly into a document together with the code that generated it, using what is called RMarkdown (http://shiny.rstudio.com/articles/rmarkdown.html). It adds some additional setup complexity we want to avoid for now but is what we used to do all the analyses that follow in the book. The main reason to mention this is that you will see a change in formatting of the R code and output from here forward as you will no longer see the command prompt (">") with the code. The output will be flagged by having two "##"'s before it. For example, the summary statistics for the *RunTime* variable from 'favstats function would look like:

```
favstats(treadmill$RunTime)
```

```
##   min   Q1 median    Q3   max     mean       sd  n missing
##  8.17 9.78  10.47 11.27 14.03 10.58613 1.387414 31       0
```

Statisticians (and other scientists) are starting to use these methods because they provide what is called "Reproducible research" (Gandrud, 2015) where all the code and output it produced are available in a single place. This allows different researchers to run and verify results or the original researchers to revisit their earlier work at a later date and recreate all their results. Scientific publications are currently encouraging researchers to work in this way and may someday require it. In this book, we focus on the R code and show the results from running it, but you may want to consider exploring these alternative options.

Finally, when you are done with your work and attempt to exit out of RStudio, it will ask you to save your workspace. You do not need to do this and would be better served not to do this. If you are in the practice of saving your workspace, you will end up with tons of data. frames that open each time you use it and it will be harder to find and manage the ones you are currently working with. If you save your R code via the script window, you can re-create any results by simply re-running that code. If you find that you have lots of "stuff" in your workspace, just run `rm(list = ls())`. It will delete all the data sets from your workspace.

1.4 Chapter summary

This chapter covered getting R and RStudio downloaded and some basics of working with R via RStudio. You should be able to read a data set into R and run some basic functions, all done using the RStudio interface. If you are struggling with this, you should seek additional help with these technical issues so that you are ready for more complicated statistical methods that are going to be encountered in the following chapters. For most assignments, we will give you a seed of the basic R code that you need and then you will modify it to work on your data set of interest. As mentioned previously, the way everyone learns R is by starting with some example code that does most of what you want to do and then you modify it. If you can complete the Practice Problems that follow, you are well on your way to learning to use R.

The statistical methods in this chapter were minimal and all should have been review. They involved a quick reminder of summarizing the center, spread, and shape of distributions using numerical summaries of the mean and SD and/or the min, Q1, median, Q3, and max and the histogram and boxplot as graphical summaries. We revisited the ideas of symmetry and skew. But the main point was really to get a start on using R to provide results you should be familiar with from your previous statistics experience(s).

1.5 Important R Code

To help you learn and use R, there is a section highlighting the most important R code used near the end of each chapter. The dark text will never change but the lighter (red) text will need to be customized to your particular application. The sub-bullet for each function will discuss the use of the function and pertinent options or packages required. You can use this as a guide to finding the function names and some hints about options that will help you to get the code to work or you can revisit the worked examples using each of the functions.

- FILENAME `<- read.csv("path to csv file/FILENAME.csv")`
  - Can be generated using "Import Dataset" button or by modifying this text.
- DATASETNAME $VARIABLENAME
  - To access a particular variable in a data. frame called DATASETNAME, use a $ and then the VARIABLENAME.
- `head`(DATASETNAME)
  - Provides a list of the first few rows of the data set for all the variables in it.
- `mean`(DATASETNAME$VARIABLENAME)
  - Calculates the mean of the observations in a variable.

- `sd`(DATASETNAME$VARIABLENAME)
  - Calculates the SD of the observations in a variable.
- `favstats`(DATASETNAME$VARIABLENAME)
  - Provides a suite of numerical summaries of the observations in a variable.
  - Requires the package to be loaded (`require(mosaic)` after installing the package).
- `hist`(DATASETNAME$VARIABLENAME)
  - Makes a histogram.
- `boxplot`(DATASETNAME$VARIABLENAME)
  - Makes a boxplot.

1.6 Practice problems

In each chapter, the last section contains some questions for you to complete to make sure you understood the material. You can download the code to answer questions 0.1 to 0.5 below at http://www.math.montana.edu/courses/s217/documents/Ch0.Rmd. But to practice learning R, it would be most useful for you to try to accomplish the requested tasks yourself and then only refer to the provided R code if/when you struggle. These questions provide a great venue to check your learning, often to see the methods applied to another data set, and for something to discuss in study groups, with your instructor, and/or at the Math Learning Center.

1.1. Read in the treadmill data set discussed above and find the mean and SD of the Ages (*Age* variable) and Body Weights (*BodyWeight* variable). In studies involving human subjects, it is common to report a summary of characteristics of the subjects. Why does this matter? Think about how your interpretation of any study of the fitness of subjects would change if the mean age had been 20 years older or 35 years younger.

1.2. How does knowing about the distribution of results for *Age* and *BodyWeight* help you understand the results for the Run Times discussed above?

1.3. The mean and SD are most useful as summary statistics only if the distribution is relatively symmetric. Make a discuss the shape of the distribution (is it skewed right, skewed left, approximately symmetric?; are there outliers?). Approximately what range of ages does this study pertain to?

1.4. The weight responses are in kilograms and you might prefer to see them in pounds. The conversion is lbs=2. 205*kgs. Create a new variable in the* `treadmill`
data.frame called BWlb* using this code:

```
treadmill$BWlb <- 2. 205*treadmill$BodyWeight
```

and find the mean and SD of the new variable (*BWlb*).

1.5. Make histograms and boxplots of the original *BodyWeight* and new *BWlb* variables. Discuss aspects of the distributions that changed and those that remained the same with the transformation from kilograms to pounds.

Chapter 2

(R)e-Introduction to statistics

The previous material served to get us started in R and to get a quick review of same basic descriptive statistics. Now we will begin to engage some new material and exploit the power of R to do some statistical inference. Because inference is one of the hardest topics to master in statistics, we will also review some basic terminology that is required to move forward in learning more sophisticated statistical methods. To keep this "review" as short as possible, we will not consider every situation you learned in introductory statistics and instead focus exclusively on the situation where we have a quantitative response variable measured on two groups, adding a new graphic called a "bean plot" to help us see the differences in the observations in the groups.

2.1 Histograms, boxplots, and density curves

Part of learning statistics is learning to correctly use the terminology, some of which is used colloquially differently than it is used in formal statistical settings. The most commonly "misused" term is ***data***. In statistical parlance, we want to note the plurality of data. Specifically, ***datum*** is a single measurement, possibly on multiple random variables, and so it is appropriate to say that "**a datum is...**". Once we move to discussing data, we are now referring to more than one observation, again on one, or possibly more than one, random variable, and so we need to use "**data are...**" when talking about our observations. We want to distinguish our use of the term "data" from its more colloquial[1] usage that often involves treating it as singular. In a statistical setting "data" refers to measurements of our cases or units. When we summarize the results of a study (say providing the mean and SD), that information is not "data". We used our data to generate that information. Sometimes we also use the term "data set" to refer to all our observations and this is a singular term to refer to the group of observations and this makes it really easy to make mistakes on the usage of this term.

It is also really important to note that ***variables*** have to vary – if you measure the sex of your subjects but are only measuring females, then you do not have a "variable". You may not know if you have real variability in a "variable" until you explore the results you obtained.

The last, but probably most important, aspect of data is the context of the measurement. The "who, what, when, and where" of the collection of the observations is critical to the sort of conclusions we can make based on the results. The information on the study design provides information required to assess the scope of inference of the study. Generally, remember to think about the research questions the researchers were trying to answer and whether their study actually would answer those questions. There are no formulas to help us sort some of these things out, just critical thinking about the context of the measurements.

[1] You will more typically hear "data is" but that more often refers to information, sometimes even statistical summaries of data sets, than to observations collected as part of a study, suggesting the confusion of this term in the general public. We will explore a data set in Chapter 4 related to perceptions of this issue collected by researchers at http://fivethirtyeight.com/.

To make this concrete, consider the data collected from a study (Plaster, 1989) to investigate whether perceived physical attractiveness had an impact on the sentences or perceived seriousness of a crime that male jurors might give to female defendants. The researchers showed the participants in the study (men who volunteered from a prison) pictures of one of three young women. Each picture had previously been decided to be either beautiful, average, or unattractive by the researchers. Each "juror" was randomly assigned to one of three levels of this factor (which is a categorical predictor or explanatory variable) and then each rated their picture on a variety of traits such as how warm or sincere the woman appeared. Finally, they were told the women had committed a crime (also randomly assigned to either be told she committed a burglary or a swindle) and were asked to rate the seriousness of the crime and provide a suggested length of sentence. We will bypass some aspects of their research and just focus on differences in the sentence suggested among the three pictures. To get a sense of these data, let's consider the first and last parts of the data set:

Subject	Attr	Crime	Years	Serious	independent	Sincere
1	Beautiful	Burglary	10	8	9	8
2	Beautiful	Burglary	3	8	9	3
3	Beautiful	Burglary	5	5	6	3
4	Beautiful	Burglary	1	3	9	8
5	Beautiful	Burglary	7	9	5	1
...	...	...	...	...	...	...
108	Average	Swindle	3	3	5	4
109	Average	Swindle	3	2	9	9
110	Average	Swindle	2	1	8	8
111	Average	Swindle	7	4	9	1
112	Average	Swindle	6	3	5	2
113	Average	Swindle	12	9	9	1
114	Average	Swindle	8	8	1	5

When working with data, we should always start with summarizing the sample size. We will use $\boldsymbol{n}$ for the number of subjects in the sample and denote the population size (if available) with $\boldsymbol{N}$. Here, the sample size is $\boldsymbol{n=114}$. In this situation, we do not have a random sample from a population (these were volunteers from the population of prisoners at the particular prison) so we cannot make inferences to a larger group. But we can assess whether there is a ***causal effect***[2]: if sufficient evidence is found to conclude that there is some difference in the responses across the treated groups, we can attribute those differences to the treatments applied, since the groups should be same otherwise due to the pictures being randomly assigned to the "jurors". The story of the data set – that it was collected on prisoners – becomes pretty important in thinking about the ramifications of any results. Are male prisoners different from the population of college males or all residents of a state such as Montana? If so, then we should not assume that the detected differences, if detected, would also exist in some other group of male subjects. The lack of a random sample makes it impossible to assume that this set of prisoners might be like other prisoners. So there are definite limitations to the inferences in the following results. But it is still interesting to see if the pictures caused a difference in the suggested mean sentences, even though the inferences are limited to this group of prisoners. If this had been an observational study (suppose that the prisoners could select one of the three pictures), then we would have to avoid any of the "causal" language that we can consider here because the pictures were not randomly assigned to the subjects. Without random assignment, the explanatory variable of picture choice could be ***confounded*** with another characteristic of prisoners that was related to which picture they selected and the rating they provided. Confounding is not the only reason to avoid causal statements with non-random assignment but the inability to separate the effect of other variables (measured or unmeasured) from the differences we are observing means that our inferences in these situations need to be carefully stated.

Instead of loading this data set into R using the "Import Dataset" functionality, we can load an R package

[2]As noted previously, we reserve the term "effect" for situations where random assignment allows us to consider causality as the reason for the differences in the response variable among levels of the explanatory variable, but this is only the case if we find evidence against the null hypothesis of no difference in the groups.

that contains the data, making for easy access to this data set. The package called `heplots` contains a data set called MockJury that contains the results of the study. We also rely the R package called `mosaic` (Pruim, Kaplan, and Horton, 2016) that was introduced previously. First (but only once), you need to install both packages, which can be done either using the Packages tab in the lower right panel of R-studio or using the `install.packages` function with quotes around the package name:

```
> install. packages("heplots")
```

After making sure that both packages are installed, we use the `require` function around the package name (no quotes now!) to load the package, something that you need to do any time you want to use features of a package.

```
require(heplots)
require(mosaic)
```

There will be some results of the loading process that may discuss loading other required packages. If the output says that it needs a package that is unavailable, then follow the same process noted above to install that package as well.

To load the data set that is available in an active package, we use the `data` function.

```
data(MockJury)
```

Now there will be a data.frame called `MockJury` available for us to analyze and some information about it in the Environment tab. Again, we can find out more about the data set in a couple of ways. First, we can use the `View` function to provide a spreadsheet type of display in the upper left panel. Second, we can use the `head` and `tail` functions to print out the beginning and end of the data set. Because there are so many variables, it may wrap around to show all the columns.

```
View(MockJury)
head(MockJury)
```

```
##        Attr    Crime Years Serious exciting calm independent sincere warm
## 1 Beautiful Burglary    10       8        6    9           9       8    5
## 2 Beautiful Burglary     3       8        9    5           9       3    5
## 3 Beautiful Burglary     5       5        3    4           6       3    6
## 4 Beautiful Burglary     1       3        3    6           9       8    8
## 5 Beautiful Burglary     7       9        1    1           5       1    8
## 6 Beautiful Burglary     7       9        1    5           7       5    8
##   phyattr sociable kind intelligent strong sophisticated happy ownPA
## 1       9        9    9           6      9             9     5     9
## 2       9        9    4           9      5             5     5     7
## 3       7        4    2           4      5             4     5     5
## 4       9        9    9           9      9             9     9     9
## 5       8        9    4           7      9             9     8     7
## 6       8        9    5           8      9             9     9     9
```

```
tail(MockJury)
```

```
##        Attr   Crime Years Serious exciting calm independent sincere warm
## 109 Average Swindle     3       2        7    6           9       9    6
## 110 Average Swindle     2       1        8    8           8       8    8
## 111 Average Swindle     7       4        1    6           9       1    1
## 112 Average Swindle     6       3        5    3           5       2    4
## 113 Average Swindle    12       9        1    9           9       1    1
## 114 Average Swindle     8       8        1    9           1       5    1
##     phyattr sociable kind intelligent strong sophisticated happy ownPA
## 109       4        7    6           8      6             5     7     2
```

```
## 110       8        9    9           9      9            9     9     6
## 111       1        9    4           1      1            1     1     9
## 112       1        4    9           3      3            9     5     3
## 113       1        9    1           9      9            1     9     1
## 114       1        9    1           1      9            5     1     1
```

When data sets are loaded from packages, there is often extra documentation available about the data set which can be accessed using the help function. In this case, it will bring up a screen with information about the study and each variable that was measured.

```
help(MockJury)
```

The help function is also useful with functions in R to help you understand options and, at the bottom of the help, see examples of using the function.

With many variables in a data set, it is often useful to get some quick information about all of them; the `summary` function provides useful information whether the variables are categorical or quantitative and notes if any values were missing.

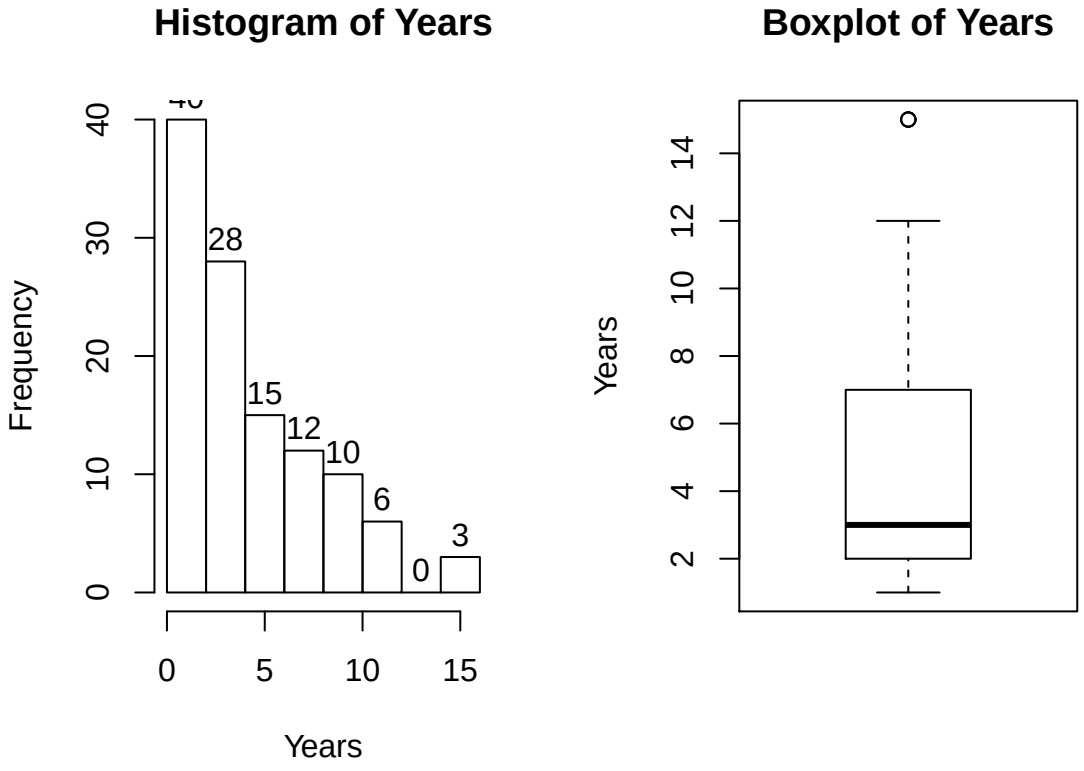
```
summary(MockJury)
```

```
##            Attr         Crime        Years           Serious
##  Beautiful   :39   Burglary:59   Min.   : 1.000   Min.   :1.000
##  Average     :38   Swindle :55   1st Qu.: 2.000   1st Qu.:3.000
##  Unattractive:37                 Median : 3.000   Median :5.000
##                                  Mean   : 4.693   Mean   :5.018
##                                  3rd Qu.: 7.000   3rd Qu.:6.750
##                                  Max.   :15.000   Max.   :9.000
##     exciting          calm        independent       sincere
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:3.000   1st Qu.:4.250   1st Qu.:5.000   1st Qu.:3.000
##  Median :5.000   Median :6.500   Median :6.500   Median :5.000
##  Mean   :4.658   Mean   :5.982   Mean   :6.132   Mean   :4.789
##  3rd Qu.:6.000   3rd Qu.:8.000   3rd Qu.:8.000   3rd Qu.:7.000
##  Max.   :9.000   Max.   :9.000   Max.   :9.000   Max.   :9.000
##       warm         phyattr        sociable          kind
##  Min.   :1.00   Min.   :1.00   Min.   :1.000   Min.   :1.000
##  1st Qu.:2.00   1st Qu.:2.00   1st Qu.:5.000   1st Qu.:3.000
##  Median :5.00   Median :5.00   Median :7.000   Median :5.000
##  Mean   :4.57   Mean   :4.93   Mean   :6.132   Mean   :4.728
##  3rd Qu.:7.00   3rd Qu.:8.00   3rd Qu.:8.000   3rd Qu.:7.000
##  Max.   :9.00   Max.   :9.00   Max.   :9.000   Max.   :9.000
##   intelligent        strong      sophisticated       happy
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:4.000   1st Qu.:4.000   1st Qu.:3.250   1st Qu.:3.000
##  Median :7.000   Median :6.000   Median :5.000   Median :5.000
##  Mean   :6.096   Mean   :5.649   Mean   :5.061   Mean   :5.061
##  3rd Qu.:8.750   3rd Qu.:7.000   3rd Qu.:7.000   3rd Qu.:7.000
##  Max.   :9.000   Max.   :9.000   Max.   :9.000   Max.   :9.000
##      ownPA
##  Min.   :1.000
##  1st Qu.:5.000
##  Median :6.000
##  Mean   :6.377
##  3rd Qu.:9.000
##  Max.   :9.000
```

If we take a few moments to explore the output we can discover some useful aspects of the data set. The output is organized by variable, providing summary information based on the type of variable, either counts by category for categorical variables `Attr`
and `Crime` mean for quantitative variables. If present, you would also get a count of missing values that are called "NAs" in R. For the first variable, called `Attr` in the data.frame and that we might we find counts of the number of subjects shown each picture: 37/114 viewed the "Unattractive" picture, 38 viewed "Average", and 39 viewed "Beautiful". We can also see that suggested prison sentences (data.frame variable `Years`) ranged from 1 year to 15 years with a median of 3 years. It seems that all the other variables except for *Crime* (type of crime that they were told the pictured woman committed) contained responses between 1 and 9 based on rating scales from 1 = low to 9 = high.

To accompany the numerical summaries, histograms, and boxplots can provide some initial information on the shape of the distribution of the responses for the Figure **??** contains the histogram and boxplot of Years, ignoring any information on which picture the "jurors" were shown. The calls to the two plotting functions are enhanced slightly to add better labels.

```
hist(MockJury$Years, xlab="Years", labels=T, main="Histogram of Years")
boxplot(MockJury$Years, ylab="Years", main="Boxplot of Years")
```



The distribution appears to have a strong right skew with three observations at 15 years flagged as potential outliers. You can only tell that there are three observations and that they are at 15 by looking at both plots – the bar around 15 years in the histogram has a count of three and the boxplot only shows a single point at 15 which is actually three tied points at exactly 15 years plotted on top of each other (we call this "overplotting"). These three observations really seem to be the upper edge of the overall pattern of a strongly right skewed distribution, so even though they are flagged in the boxplot, we likely would not want to remove them from our data set. In real data sets, outliers are commonly encountered and the first step is to verify that they were not errors in recording. The next step is to study their impact on the statistical analyses

performed, potentially considering reporting results with and without the influential observation(s) in the results. If the analysis is unaffected by the "unusual" observations, then it matters little whether they are dropped or not. If they do affect the results, then reporting both versions of results allows the reader to judge the impacts for themselves. It is important to remember that sometimes the outliers are the most interesting part of the data set.

Often when statisticians think of distributions, we think of the smooth underlying shape that led to the data set that is being displayed in the histogram. Instead of binning up observations and making bars in the histogram, we can estimate what is called a ***density curve*** as a smooth curve that represents the observed distribution of the responses. Density curves can sometimes help us see features of the data sets more clearly.

To understand the density curve, it is useful to initially see the histogram and density curve together. The density curve is scaled so that the total area[3] under the curve is 1. To make a comparable histogram, the y-axis needs to be scaled so that the histogram is also on the "density" scale which makes the bar heights required so that the proportion of the total data set in each bar is represented by the area in each bar (remember that area is height times width). So the height depends on the width of the bars and the total area across all the bars has to be 1. In the `hist` function, the `freq=F` to get density-scaled histogram bars. The density curve is added to the histogram using the R code of `lines(density())`, producing the result in Figure 2.1 with added modifications of options for `lwd` (line width) and `col` (color) to make the plot more interesting. You can see how the density curve somewhat matches the histogram bars but deals with the bumps up and down and edges a little differently. We can pick out the strong right skew using either display and will rarely make both together.

```
hist(MockJury$Years,freq=F,xlab="Years",main="Histogram of Years")
lines(density(MockJury$Years),lwd=3,col="red")
```

Histograms can be sensitive to the choice of the number of bars and even the cut-offs used to define the bins for a given number of bars. Small changes in the definition of cut-offs for the bins can have noticeable impacts on the shapes observed but this does not impact density curves. We are not going to tinker with the default choices for bars in histogram as they are reasonably selected, but we can add information on the original observations being included in each bar to better understand the choices that `hist` is making. In the previous display, we can add what is called a ***rug*** to the plot, were a tick mark is made on the x-axis for each observation. Because the responses were provided as whole years (1, 2, 3, . . . , 15), we need to use a graphical technique called ***jittering*** to add a little noise[4] to each observation so all the observations at each year value do not plot as a single line. In Figure 2.2, the added tick marks on the x-axis show the approximate locations of the original observations. We can see how there are 3 observations at 15 (all were 15 and the noise added makes it possible to see them all). The limitations of the histogram arise around the 10 year sentence area where there are many responses at 10 years and just one at both 9 and 11 years, but the histogram bars sort of miss this that aspect of the data set. The density curve did show a small bump at 10 years. Density curves are, however, not perfect and this one shows area for sentences less than 0 years which is not possible here.

```
hist(MockJury$Years, freq=F, xlab="Years",
     main="Histogram of Years with density curve and rug")
lines(density(MockJury$Years),lwd=3,col="red")
rug(jitter(MockJury$Years),col="blue",lwd=2)
```

The graphical tools we've just discussed are going to help us move to comparing the distribution of responses across more than one group. We will have two displays that will help us make these comparisons. The simplest is *the* ***side-by-side boxplot***, where a boxplot is displayed for each group of interest using the same y-axis scaling. In R, we can use its ***formula*** notation to see if the response (`Years`) differs based on the group (`Attr`) by using something like `Y~X` or, here, `Years~Attr`. We also need to tell R where to find the

[3] If you've taken calculus, you will know that the curve is being constructed so that the integral from $-\infty$ to ∞ is 1. If you don't know calculus, think of a rectangle with area of 1 based on its height and width. These cover the same area but the top of the region wiggles.

[4] Jittering typically involves adding random variability to each observation that is uniformly distributed in a range determined based on the spacing of the function, the results will change. For more details, type `help(jitter)` in R.

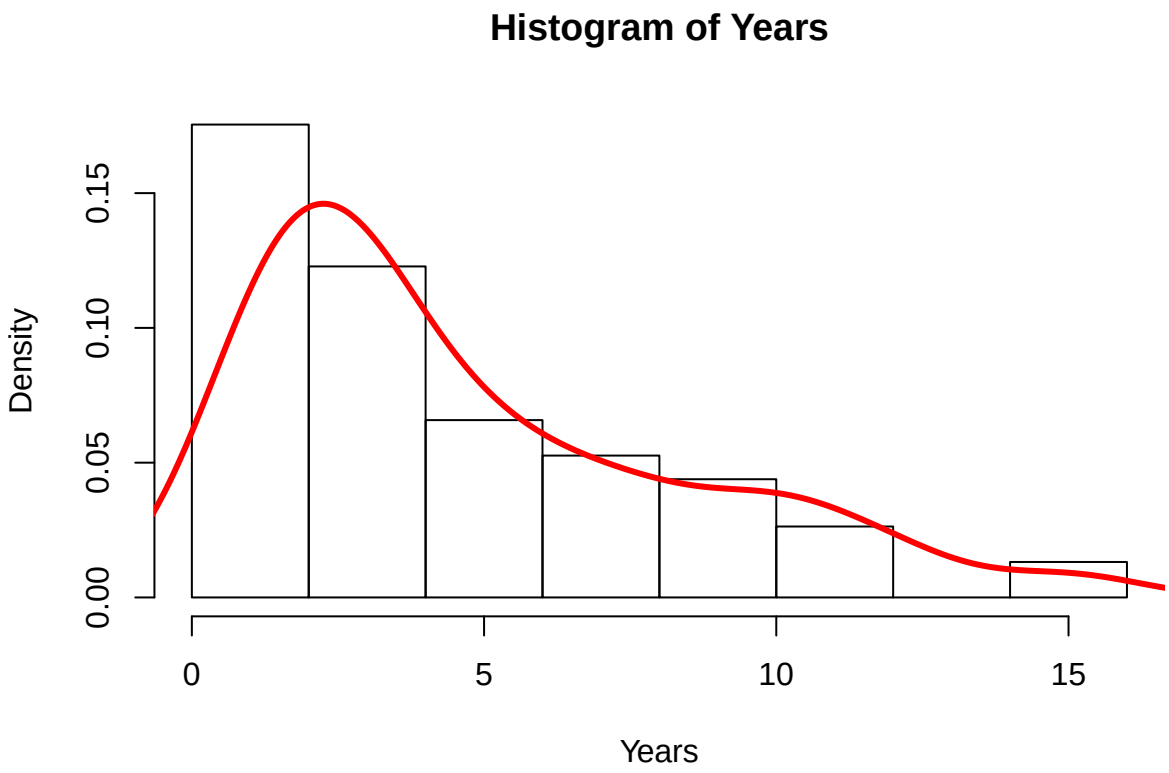


Figure 2.1: Histogram and density curve of Years data.

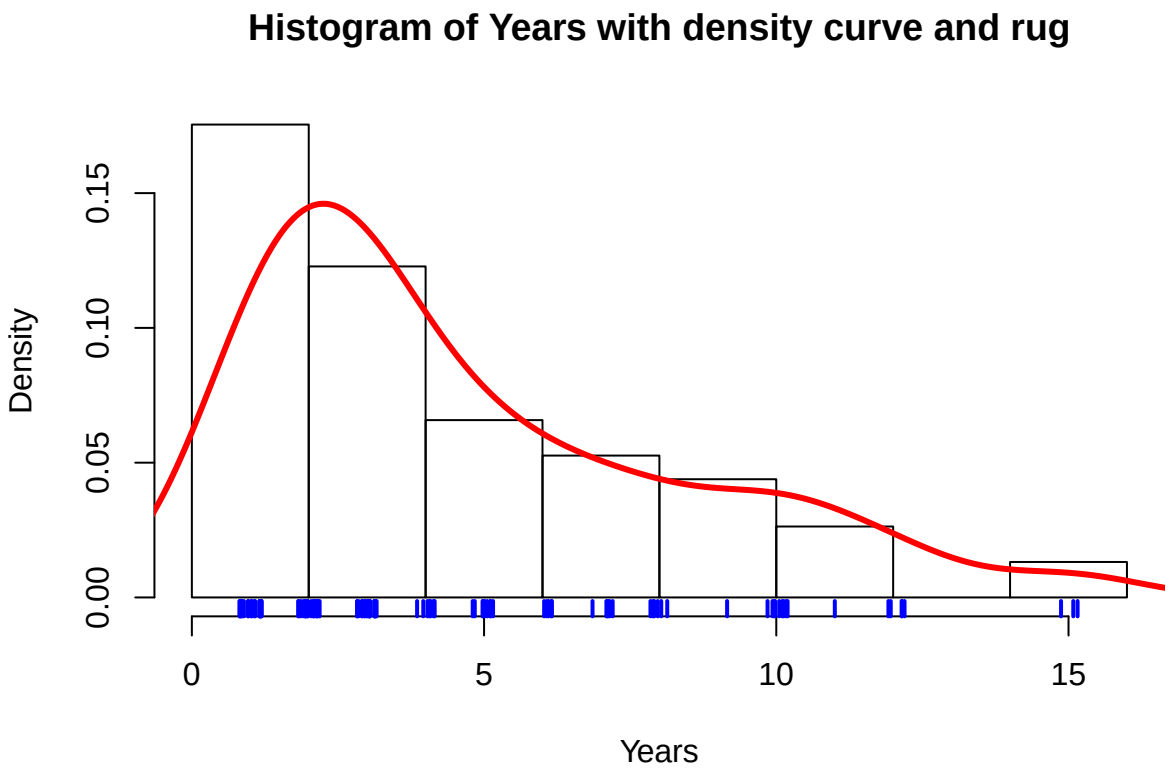


Figure 2.2: Histogram with density curve and rug plot of the jittered responses.

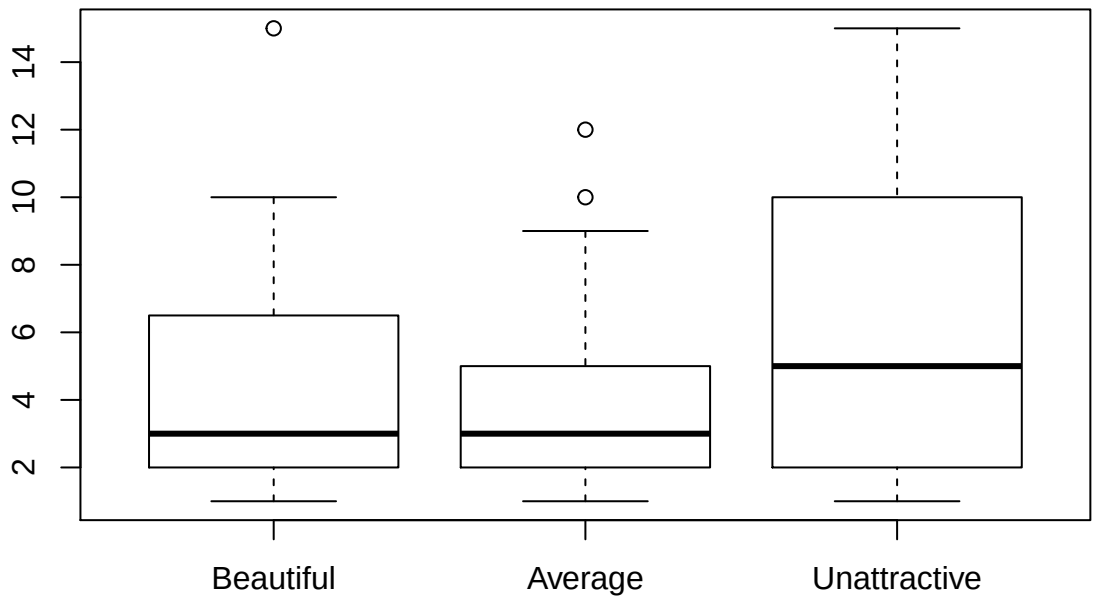


Figure 2.3: Side-by-side boxplot of Years based on picture groups.

variables – use the last option in the command, `data=DATASETNAME` , to inform R of the data.frame to look in to find the variables. In this example, `data=MockJury`. We will use the formula and `data=...` options in almost every function we use from here forward. Figure 2.3 contains the side-by-side boxplots showing right skew for all the groups, slightly higher median and more variability for the *Unattractive* group along with some potential outliers indicated in two of the three groups.

```
boxplot(Years~Attr,data=MockJury)
```

The "~" (which is read as the *tilde* symbol, which you can find in the upper left corner of your keyboard) notation will be used in two ways this semester. The formula use in R employed previously declares that the response variable here is *Years* and the explanatory variable is *Attr*. The other use for "~" is as shorthand for "is distributed as" and is used in the context of `Y~N(0,1)`, which translates (in statistics) to defining the random variable Y as following a Normal distribution[5] with mean 0 and standard deviation of 1. In the current situation, we could ask whether the `Years` variable seems like it may follow a normal distribution, in other words, is *Years*`~N(0,1)`? Since the responses are right skewed with some groups having outliers, it is not reasonable to assume that the *Years* variable for any of the three groups may follow a Normal distribution (more later on the issues this creates!). Remember that
μ and σ are parameters where
μ ("mu") is our standard symbol for the ***population mean*** and that σ ("sigma") is the symbol of the ***population standard deviation***.

[5]Remember the bell-shaped curve you encountered in introductory statistics? If not, you can see some at https://en.wikipedia.org/wiki/Normal_distribution

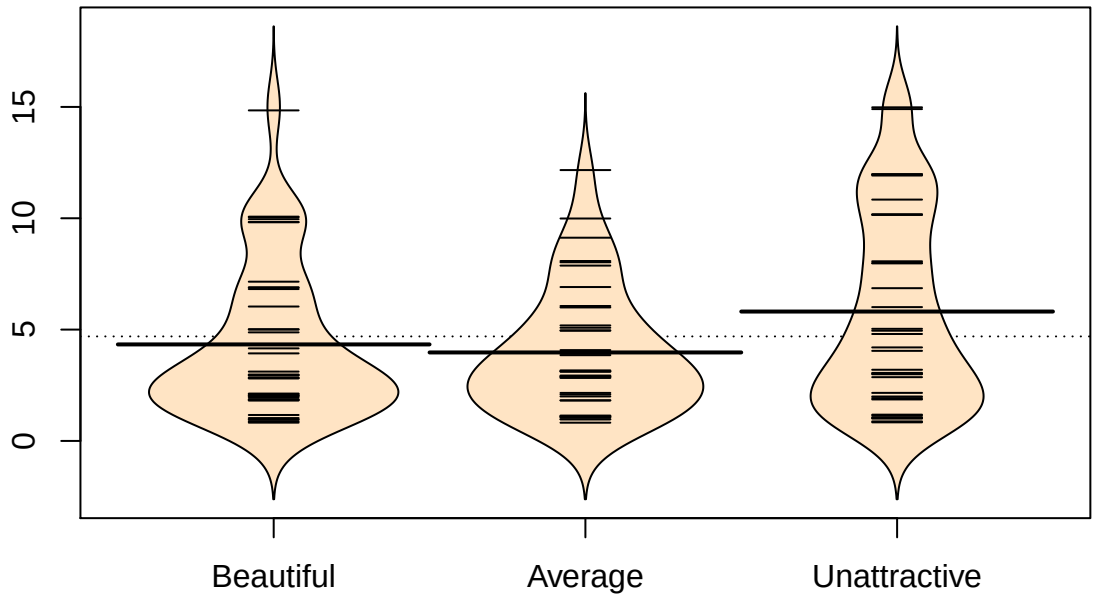


Figure 2.4: Beanplot of Years by picture group. Long, bold lines correspond to mean of each group.

2.2 Beanplots

The other graphical display for comparing multiple groups we will use is a newer display called a ***beanplot*** (Kampstra, 2008). Figure 2.4 shows an example of a beanplot that provides a side-by-side display that contains the density curves, the original observations that generated the density curve in a (jittered) rug-plot, the mean of each group, and the overall mean of the entire data set. For each group, the density curves are mirrored to aid in visual assessment of the shape of the distribution, which makes a "bean" in some cases. This mirroring also creates a shape that resembles a violin with skewed distributions so this display has also been called a "violin plot". The innovation in the beanplot is to add bold horizontal lines at the mean for each group. It also adds a lighter dashed line for the overall mean. All together this plot shows us information on the center (mean), spread, and shape of the distributions of the responses. Our inferences typically focus on the means of the groups and this plot allows us to compare those across the groups while gaining information on the shapes of the distributions of responses in each group.

To use the `beanplot` function we need to install and load the `beanplot` package. The function works like the boxplot used previously except that options for `log`, `col`, and `method` need to be specified. Use these[6] options for any beanplots you make: `log="", col="bisque", method="jitter"`

```
require(beanplot)
beanplot(Years~Attr,data=MockJury,log="",col="bisque",method="jitter")
```

Figure 2.4 reinforces the strong right skews that were also detected in the boxplots previously. The three large sentences of 15 years can now be clearly identified, with one in the *Beautiful* group and two in the

[6]Well, you can use other colors (try "lightblue" for example), but I think bisque looks nice in these plots.

Unattractive group. The *Unattractive* group seems to have more high observations than the other groups even though the *Beautiful* group had the largest number of observations around 10years. The mean sentence was highest for the *Unattractive* group and the difference in the means between *Beautiful* and *Average* was small.

In this example, it appears that the mean for *Unattractive* is larger than the other two groups. But is this difference real? We will never know the answer to that question, but we can assess how likely we are to have seen a result as extreme or more extreme than our result, assuming that there is no difference in the means of the groups. And if the observed result is (extremely) unlikely to occur, then we can reject the hypothesis that the groups have the same mean and conclude that there is evidence of a real difference. To start exploring whether there are differences in the means, we need to have numerical values to compare. We can get means and standard deviations by groups easily using the same formula notation with the `mean` and `sd` functions if the `mosaic` package is loaded.

```
mean(Years ~ Attr, data = MockJury)
```

```
##    Beautiful      Average Unattractive
##     4.333333     3.973684     5.810811
```

```
sd(Years ~ Attr, data = MockJury)
```

```
##    Beautiful      Average Unattractive
##     3.405362     2.823519     4.364235
```

We can also use the `favstats` function to get those summaries and others.

```
favstats(Years ~ Attr, data = MockJury)
```

```
##            Attr min Q1 median   Q3 max     mean       sd  n missing
## 1    Beautiful   1  2      3  6.5  15 4.333333 3.405362 39       0
## 2      Average   1  2      3  5.0  12 3.973684 2.823519 38       0
## 3 Unattractive   1  2      5 10.0  15 5.810811 4.364235 37       0
```

Based on these results, we can see that there is an estimated difference of almost 2 years in the mean sentence between *Average* and *Unattractive* groups. Because there are three groups being compared in this study, we will have to wait until Chapter 3 and the One-Way ANOVA test to fully assess evidence related to some difference among the three groups. For now, we are going to focus on comparing the mean *Years* between *Average* and *Unattractive* groups – which is a ***2 independent sample mean*** situation and something you should have seen before. Remember that the "independent" sample part of this refers to observations that are independently observed for the two groups as opposed to the paired sample situation that you may have explored where one observation from the first group is related to an observation in the second group (repeated measures on the same person or the famous "twin" studies with one twin assigned to each group).

Here we are going to use the "simple" two independent group scenario to review some basic statistical concepts and connect two different frameworks for conducting statistical inference: randomization and parametric inference techniques. ***Parametric*** statistical methods involve making assumptions about the distribution of the responses and obtaining confidence intervals and/or p-values using a *named* distribution (like the z or *t*-distributions). Typically these results are generated using formulas and looking up areas under curves or cutoffs using a table or a computer. ***Randomization***-based statistical methods use a computer to shuffle, sample, or simulate observations in ways that allow you to obtain distributions of possible results to find areas and cutoffs without resorting to using tables and named distributions. Randomization methods are what are called ***nonparametric*** methods that often make fewer assumptions (they are ***not free of assumptions***!) and so can handle a larger set of problems more easily than parametric methods. When the assumptions involved in the parametric procedures are met by a data set, the randomization methods often provide very similar results to those provided by the parametric techniques. To be a more sophisticated statistical consumer, it is useful to have some knowledge of both of these approaches to statistical inference and the fact that they can provide similar results might deepen your understanding of both approaches.

We will start with comparing the *Average* and *Unattractive* groups to compare these two ways of doing inference. We could remove the *Beautiful* group observations in a spreadsheet program and read that new

data set back into R, but it is actually pretty easy to use R to do data management once the data set is loaded. To remove the observations that came from the *Beautiful* group, we are going to generate a new variable that we will call `NotBeautiful` that is true when observations came from another group (*Average* or *Unattractive*) and false for observations from the *Beautiful* group. To do this, we will apply the ***not equal*** logical function (`!=`) to the variable `Attr`, inquiring whether it was different from the `"Beautiful"` level. You can see the content of the new variable in the output:

```
MockJury$NotBeautiful <- MockJury$Attr != "Beautiful"
MockJury$NotBeautiful
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
##  [23]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [34]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [45]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [56]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [67]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
##  [78] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [89] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
## [100]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [111]  TRUE  TRUE  TRUE  TRUE
```

This new variable is only FALSE for the *Beautiful* responses as we can see if we compare some of the results from the original and new variable:

```
head(data.frame(MockJury$Attr, MockJury$NotBeautiful))
```

```
##   MockJury.Attr MockJury.NotBeautiful
## 1     Beautiful                 FALSE
## 2     Beautiful                 FALSE
## 3     Beautiful                 FALSE
## 4     Beautiful                 FALSE
## 5     Beautiful                 FALSE
## 6     Beautiful                 FALSE
```

```
tail(data.frame(MockJury$Attr, MockJury$NotBeautiful))
```

```
##     MockJury.Attr MockJury.NotBeautiful
## 109       Average                  TRUE
## 110       Average                  TRUE
## 111       Average                  TRUE
## 112       Average                  TRUE
## 113       Average                  TRUE
## 114       Average                  TRUE
```

To get rid of one of the groups, we need to learn a little bit about data management in R. ***Brackets*** `([, ])` are used to modify the rows or columns in a data.frame with entries before the comma operating on rows and entries after the comma on the columns. For example, if you want to see the results for the 5^{th} subject, you can reference the 5^{th} row of the data.frame using `[5, ]` after the data.frame name:

```
MockJury[5,]
```

```
##        Attr    Crime Years Serious exciting calm independent sincere warm
## 5 Beautiful Burglary     7       9        1    1           5       1    8
##   phyattr sociable kind intelligent strong sophisticated happy ownPA
## 5       8        9    4           7      9             9     8     7
##   NotBeautiful
## 5        FALSE
```

We could just extract the *Years* response for the 5^{th} subject by incorporating information on the row and column of interest (`Years` is the 3^{rd} column):

```
MockJury[5,3]
```

```
## [1] 7
```

In R, we can use logical vectors to keep any rows of the data.frame where the variable is true and drop any rows where it is false by placing the logical variable in the first element of the brackets. The reduced version of the data set should be saved with a different name such as `MockJury2` that is used here to reduce the chances of confusing it with the previous full data set:

```
MockJury2 <- MockJury[MockJury$NotBeautiful,]
```

You will always want to check that the correct observations were dropped either using `View(MockJury2)` or by doing a quick summary of the `Attr` variable in the new data.frame.

```
summary(MockJury2$Attr)
```

```
##    Beautiful      Average Unattractive
##            0           38           37
```

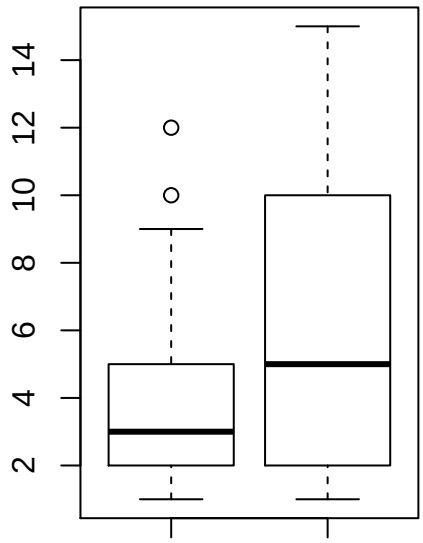
It ends up that R remembers the *Beautiful* category even though there are 0 observations in it now and that can cause us some problems. When we remove a group of observations, we sometimes need to clean up categorical variables to just reflect the categories that are present. The `factor` function creates categorical variables based on the levels of the variables that are observed and is useful to run here to clean up `Attr`.

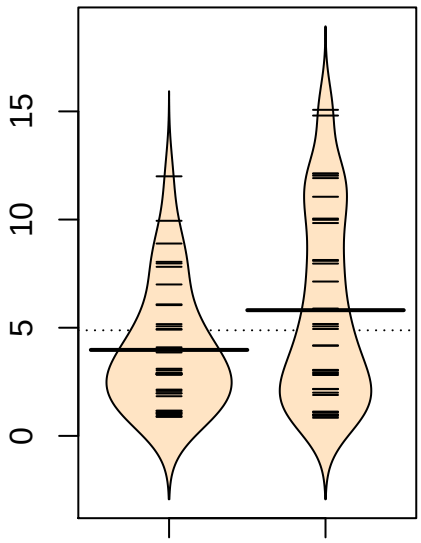
```
MockJury2$Attr <- factor(MockJury2$Attr)
summary(MockJury2$Attr)
```

```
##      Average Unattractive
##           38           37
```

Now if we remake the boxplots and beanplots, they only contain results for the two groups of interest here as seen in Figure **??**.

```
boxplot(Years ~ Attr,data=MockJury2)
beanplot(Years ~ Attr,data=MockJury2,log="",col="bisque",method="jitter")
```





The two-sample mean techniques you learned in your previous course all start with comparing the means the two groups. We can obtain the two means using the `mean` function or directly obtain the difference in the means using the `diffmean` function (both require the `mosaic` package). The `diffmean` function provides $\bar{x}_{Unattractive} - \bar{x}_{Average}$ where $\bar{x}$ (read as "x-bar") is the sample mean of observations in the subscripted group. Note that there are two directions that you could compare the means and this function chooses to take the mean from the second group name *alphabetically* and subtract the mean from the first alphabetical group name. It is always good to check the direction of this calculation as having a difference of -1.84 years versus 1.84 years could be important.

```
mean(Years ~ Attr, data=MockJury2)
```

```
##      Average Unattractive
##     3.973684     5.810811
```

```
diffmean(Years ~ Attr, data=MockJury2)
```

```
## diffmean
## 1.837127
```

2.3 Models, hypotheses, and permutations for the 2 sample mean situation

There appears to be some evidence that the *Unattractive* group is getting higher average lengths of sentences from the prisoner "jurors" than the *Average* group, but we want to make sure that the difference is real – that there is evidence to reject the assumption that the means are the same "in the population". First, a ***null***

hypothesis[7] which defines a ***null model***[8] needs to be determined in terms of ***parameters*** (the true values in the population). The research question should help you determine the form of the hypotheses for the assumed population. In the 2 independent sample mean problem, the interest is in testing a null hypothesis of $H_0 : \mu_1 = \mu_2$ versus the alternative hypothesis of $H_A : \mu_1 \neq \mu_2$, where μ_1 is the parameter for the true mean of the first group and μ_2 is the parameter for the true mean of the second group. The alternative hypothesis involves assuming a statistical model for the $i^{th}(i = 1, \ldots, n_j)$ response from the $j^{th}(j = 1, 2)$ group, $\boldsymbol{y}_{ij}$, that involves modeling it as $y_{ij} = \mu_j + \epsilon_{ij}$, where we assume that $\epsilon_{ij} \sim N(0, \sigma^2)$. For the moment, focus on the models that either assume the means are the same (null) or different (alternative), which imply:

- Null Model: $y_{ij} = \mu + \epsilon_{ij}$ There is **no** difference in **true** means for the two groups.
- Alternative Model: $y_{ij} = \mu_j + \epsilon_{ij}$ There is **a** difference in **true** means for the two groups.

Suppose we are considering the alternative model for the 4th observation ($i = 4$) from the second group ($j = 2$), then the model for this observation is $y_{42} = \mu_2 + \epsilon_{42}$, that defines the response as coming from the true mean for the second group plus a random error term for that observation, ϵ_{42}. For, say, the 5th observation from the first group ($j = 1$), the model is $y_{51} = \mu_1 + \epsilon_{51}$. If we were working with the null model, the mean is always the same (μ) - the group specified does not change the mean we use for that observation.

It can be helpful to think about the null and alternative models graphically. By assuming the null hypothesis is true (means are equal) and that the random errors around the mean follow a normal distribution, we assume that the truth is as displayed in the left panel of Figure 2.5 – two normal distributions with the same mean and variability. The alternative model allows the two groups to potentially have different means, such as those displayed in the right panel of Figure 2.5 where the second group has a larger mean. Note that in this scenario, we assume that the observations all came from the same distribution except that they had different means. Depending on the statistical procedure we are using, we basically are going to assume that the observations (y_{ij}) either were generated as samples from the null or alternative model. You can imagine drawing observations at random from the pictured distributions. For hypothesis testing, the null model is assumed to be true and then the unusualness of the actual result is assessed relative to that assumption. In hypothesis testing, we have to decide if we have enough evidence to reject the assumption that the null model (or hypothesis) is true. If we reject the null hypothesis, then we would conclude that the other model considered (the alternative model) is more reasonable. The researchers obviously would have hoped to encounter some sort of noticeable difference in the sentences provided for the different pictures and been able to find enough evidence to reject the null model where the groups "look the same".

In statistical inference, null hypotheses (and their implied models) are set up as "straw men" with every interest in rejecting them even though we assume they are true to be able to assess the evidence against them. Consider the original study design here, the pictures were randomly assigned to the subjects. If the null hypothesis were true, then we would have no difference in the population means of the groups. And this would apply if we had done a different random assignment of the pictures to the subjects. So let's try this: assume that the null hypothesis is true and randomly re-assign the treatments (pictures) to the observations that were obtained. In other words, keep the sentences (*Years*) the same and shuffle the group labels randomly. The technical term for this is doing a ***permutation*** (a random shuffling of the treatments relative to the responses). If the null is true and the means in the two groups are the same, then we should be able to re-shuffle the groups to the observed sentences (*Years*) and get results similar to those we actually observed. If the null is false and the means are really different in the two groups, then what we observed should differ from what we get under other random permutations. The differences between the two groups should be more noticeable in the observed data set than in (most) of the shuffled data sets. It helps to see an example of a permutation of the labels to understand what this means here.

In the `mosaic` package, the `shuffle` function allows us to easily perform a permutation[9]. Just one time, we

[7] The hypothesis of no difference that is typically generated in the hopes of being rejected in favor of the alternative hypothesis which contains the sort of difference that is of interest in the application.

[8] The null model is the statistical model that is implied by the chosen null hypothesis. Here, a null hypothesis of no difference translates to having a model with the same mean for both groups.

[9] We'll see the `shuffle` function in a more common usage below; while the code to generate `Perm1` is provided, it isn't something to worry about right now.

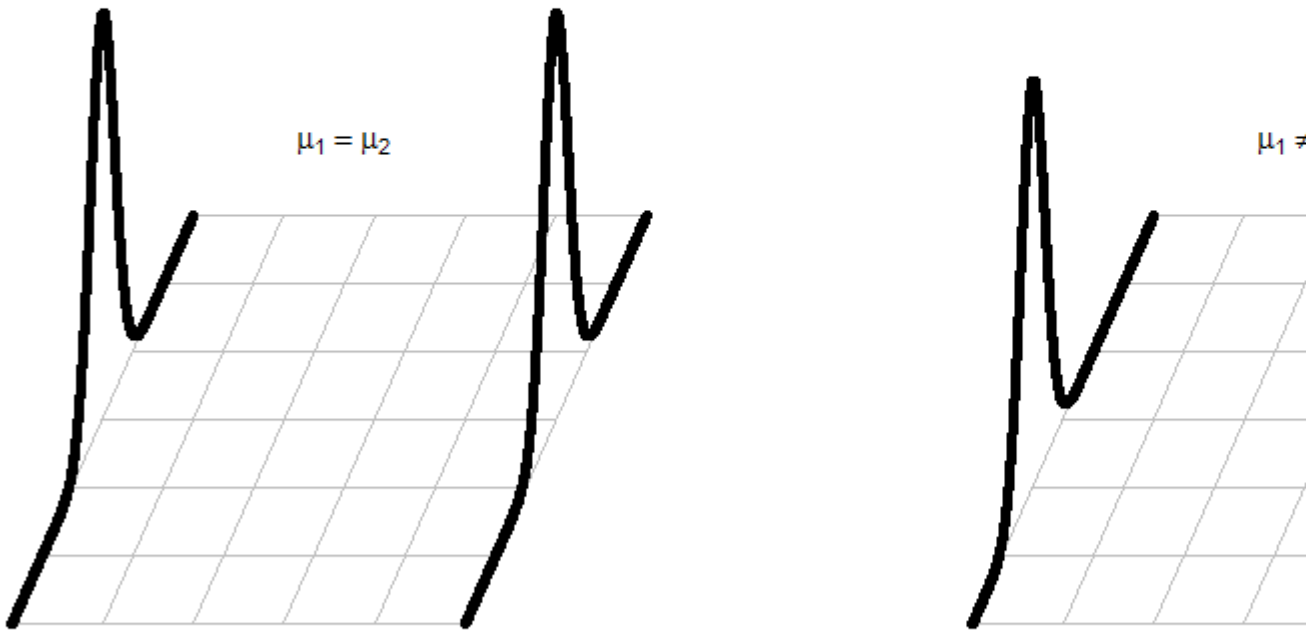


Figure 2.5: Illustration of the assumed situations under the null (left) and a single possibility that could occur if the alternative were true (right) and the true means were different.

can explore what a permutation of the treatment labels could look like in the `PermutedAttr` variable below. Note that the `Years` are held in the same place the group labels are shuffled.

```
set.seed(1234)
```

```
Perm1 <- with(MockJury2,data.frame(Years,Attr,PermutedAttr=shuffle(Attr)))
Perm1
```

```
##    Years         Attr PermutedAttr
## 1      1 Unattractive Unattractive
## 2      4 Unattractive      Average
## 3      3 Unattractive      Average
## 4      2 Unattractive      Average
## 5      8 Unattractive      Average
## 6      8 Unattractive      Average
## 7      1 Unattractive Unattractive
## 8      1 Unattractive Unattractive
## 9      5 Unattractive      Average
## 10     7 Unattractive Unattractive
## 11     1 Unattractive      Average
## 12     5 Unattractive Unattractive
## 13     2 Unattractive Unattractive
## 14    12 Unattractive      Average
## 15    10 Unattractive      Average
## 16     1 Unattractive      Average
## 17     6 Unattractive Unattractive
## 18     2 Unattractive      Average
## 19     5 Unattractive Unattractive
## 20    12 Unattractive Unattractive
## 21     6 Unattractive      Average
## 22     3 Unattractive      Average
## 23     8 Unattractive      Average
## 24     4 Unattractive Unattractive
## 25    10 Unattractive Unattractive
## 26    10 Unattractive      Average
## 27    15 Unattractive Unattractive
## 28    15 Unattractive      Average
## 29     3 Unattractive      Average
## 30     3 Unattractive      Average
## 31     3 Unattractive Unattractive
## 32    11 Unattractive      Average
## 33    12 Unattractive      Average
## 34     2 Unattractive Unattractive
## 35     1 Unattractive Unattractive
## 36     1 Unattractive Unattractive
## 37    12 Unattractive      Average
## 38     5      Average Unattractive
## 39     5      Average Unattractive
## 40     4      Average Unattractive
## 41     3      Average Unattractive
## 42     6      Average      Average
## 43     4      Average      Average
## 44     9      Average      Average
## 45     8      Average Unattractive
## 46     3      Average      Average
```

```
## 47     2      Average Unattractive
## 48    10      Average      Average
## 49     1      Average Unattractive
## 50     1      Average Unattractive
## 51     3      Average Unattractive
## 52     1      Average      Average
## 53     3      Average      Average
## 54     5      Average      Average
## 55     8      Average Unattractive
## 56     3      Average      Average
## 57     1      Average      Average
## 58     1      Average Unattractive
## 59     1      Average      Average
## 60     2      Average      Average
## 61     2      Average Unattractive
## 62     1      Average      Average
## 63     1      Average Unattractive
## 64     2      Average Unattractive
## 65     3      Average Unattractive
## 66     4      Average Unattractive
## 67     5      Average      Average
## 68     3      Average Unattractive
## 69     3      Average      Average
## 70     3      Average      Average
## 71     2      Average Unattractive
## 72     7      Average Unattractive
## 73     6      Average Unattractive
## 74    12      Average Unattractive
## 75     8      Average      Average
```

If you count up the number of subjects in each group by counting the number of times each label (Average, Unattractive) occurs, it is the same in both the `Attr` and `PermutedAttr` columns. Permutations involve randomly re-ordering the values of a variable – here the `Attr` group labels – without changing the content of the variable. This result can also be generated using what is called ***sampling without replacement***: sequentially select n labels from the original variable, removing each used label and making sure that each original `Attr` label is selected once and only once. The new, randomly selected order of selected labels provides the permuted labels. Stepping through the process helps to understand how it works: after the initial random sample of one label, there would $n-1$ choices possible; on the n^{th} selection, there would only be one label remaining to select. This makes sure that all original labels are re-used but that the order is random. Sampling without replacement is like picking names out of a hat, one-at-a-time, and not putting the names back in after they are selected. It is an exhaustive process for all the original observations. ***Sampling with replacement*** , in contrast, involves sampling from the specified list with each observation having an equal chance of selection for each sampled observation – in other words, observations can be selected more than once. This is like picking n names out of a hat that contains n names, except that every time a name is selected, it goes back into the hat – we'll use this technique in Section 2.8 to do what is called ***bootstrapping***. Both sampling mechanisms can be used to generate inferences but each has particular situations where they are most useful. For hypothesis testing, we will use permutations (sampling without replacement).

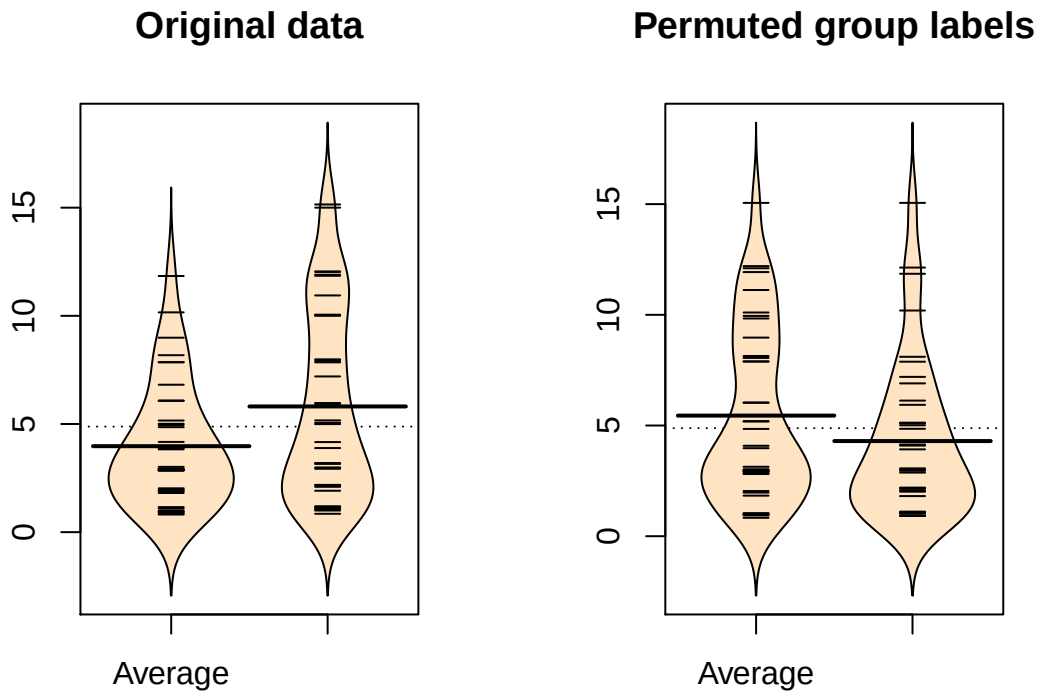
The comparison of the beanplots for the real data set and permuted version of the labels is what is really interesting (Figure **??**). The original difference in the sample means of the two groups was 1.84 years (Unattractive minus Average). The sample means are the ***statistics***
that estimate the parameters for the true means of the two groups. In the permuted data set, the difference in the means is 1.15 years in the opposite direction (Average had a higher mean than Unattractive in the permuted data).

```
mean(Years ~ PermutedAttr, data=Perm1)
```

```
##      Average Unattractive
##     5.447368     4.297297
```

```
diffmean(Years ~ PermutedAttr, data=Perm1)
```

```
##  diffmean
## -1.150071
```



These results suggest that the observed difference was larger than what we got when we did a single permutation although it was only a little bit larger than a difference we could observe in permutations if we ignore the difference in directions. Conceptually, permuting observations between group labels is consistent with the null hypothesis – this is a technique to generate results that we might have gotten if the null hypothesis were true since the responses are the same in the two groups if the null is true. We just need to repeat the permutation process many times and track how unusual our observed result is relative to this distribution of potential responses if the null were true. If the observed differences are unusual relative to the results under permutations, then there is evidence against the null hypothesis, the null hypothesis should be rejected (Reject H_0), and a conclusion should be made, in the direction of the alternative hypothesis, that there is evidence that the true means differ. If the observed differences are similar to (or at least not unusual relative to) what we get under random shuffling under the null model, we would have a tough time concluding that there is any real difference between the groups based on our observed data set.

2.4 Permutation testing for the 2 sample mean situation

In any testing situation, you must define some function of the observations that gives us a single number that addresses our question of interest. This quantity is called a ***test statistic***. These often take on complicated forms and have names like t or z statistics that relate to their parametric (named) distributions so we know where to look up ***p-values***[10]. In randomization settings, they can have simpler forms because we use the data set to find the distribution of the statistic and don't need to rely on a named distribution. We will label our test statistic $\boldsymbol{T}$ (for **T** statistic) unless the test statistic has a commonly used name. Since we are interested in comparing the means of the two groups, we can define

$$T = \bar{x}_{Unattractive} - \bar{x}_{Average},$$

which coincidentally is what the`diffmean` function provided us previously. We label our ***observed test statistic*** (the one from the original data set) as

$$T_{obs} = \bar{x}_{Unattractive} - \bar{x}_{Average},$$

which happened to be 1.84 years here. We will compare this result to the results for the test statistic that we obtain from permuting the group labels. To denote permuted results, we will add a * to the labels:

$$T^* = \bar{x}_{Unattractive} - \bar{x}_{Average^*}.$$

We then compare the $T_{obs} = \bar{x}_{Unattractive} - \bar{x}_{Average} = 1.84$ to the distribution of results that are possible for the permuted results (T^*) which corresponds to assuming the null hypothesis is true.

We need to consider lots of permutations to do a permutation test. In contrast to your introductory statistics course where, if you did this, it was just a click away, we are going to learn what was going on under the hood. Specifically, we need a ***for loop*** in R to be able to repeatedly generate the permuted data sets and record T^* for each one. Loops are a basic programming task that make randomization methods possible as well as potentially simplifying any repetitive computing task. To write a "for loop", we need to choose how many times we want to do the loop (call that `B`) and decide on a counter to keep track of where we are at in the loops (call that `b`, which goes from 1 up to `B`). The simplest loop just involves printing out the index, `print(b)` at each step. This is our first use of curly braces, { and}, that are used to group the code we want to repeatedly run as we proceed through the loop. By typing the following code in the script window and then highlighting it all and hitting the run button, R will go through the loop 5 times, printing out the counter:

```
B <- 5
for (b in (1:B)){
  print(b)
}
```

Note that when you highlight and run the code, it will look about the same with "+" printed after the first line to indicate that all the code is connected when it appears in the console, looking like this:

```
> for(b in (1:B)){
+   print(b)
+}
```

When you run these three lines of code, the console will show you the following output:

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

[10] P-values are the probability of obtaining a result as extreme as or more extreme than we observed given that the null hypothesis is true.

```
[1] 4
[1] 5
```

Instead of printing the counter, we want to use the loop to repeatedly compute our test statistic across B random permutations of the observations. The `shuffle` function performs permutations of the group labels relative to responses and the `diffmean` difference in the two group means in the permuted data set. For a single permutation, the combination of shuffling `Attr` and finding the difference in the means, storing it in a variable called `Ts` is:

```
Ts <- diffmean(Years ~ shuffle(Attr), data=MockJury2)
Ts
```

```
##  diffmean
## -0.616643
```

And putting this inside the `print` function allows us to find the test statistic under 5 different permutations easily:

```
B <- 5
for (b in (1:B)){
  Ts <- diffmean(Years ~ shuffle(Attr), data=MockJury2)
  print(Ts)
}
```

```
##   diffmean
## -0.8300142
##   diffmean
## -0.1365576
##    diffmean
## -0.08321479
##  diffmean
## 0.5035562
## diffmean
## 1.677098
```

Finally, we would like to store the values of the test statistic instead of just printing them out on each pass through the loop. To do this, we need to create a variable to store the results, let's call it `Tstar`. We know that we need to store `B` results so will create a vector of length B, which contains B elements, full of missing values (NA) using the `matrix` function:

```
Tstar <- matrix(NA, nrow=B)
Tstar
```

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

Now we can run our loop B times and store the results in `Tstar`.

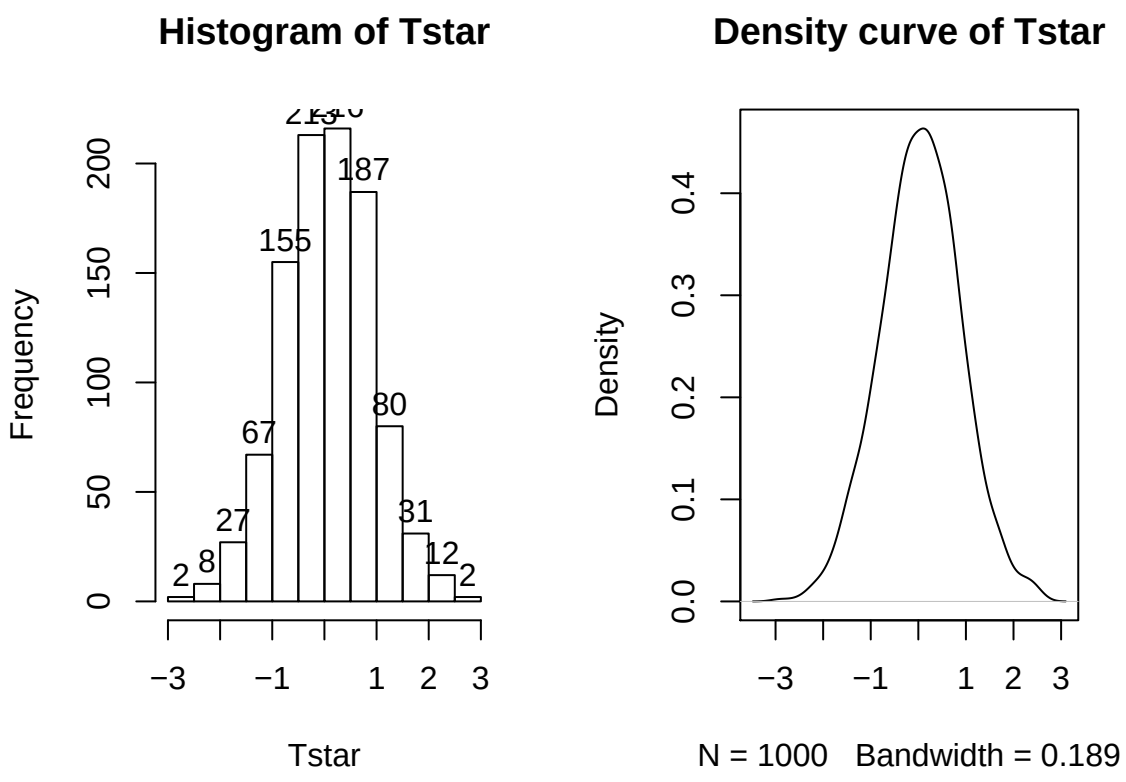
```
for (b in (1:B)){
  Tstar[b] <- diffmean(Years ~ shuffle(Attr), data=MockJury2)
}
Tstar
```

```
##             [,1]
## [1,] -0.08321479
```

```
## [2,]  0.23684211
## [3,] -0.24324324
## [4,] -0.61664296
## [5,]  0.66358464
```

Five permutations are still not enough to assess whether our T_{obs} of 1.84 is unusual and we need to do many permutations to get an accurate assessment of the possibilities under the null hypothesis. It is common practice to consider something like 1,000 permutations. The `Tstar` vector when we set B the permutation distribution for the selected test statistic under[11] the null hypothesis – what is called the ***null distribution*** of the statistic. The null distribution is the distribution of possible values of a statistic under the null hypothesis. We want to visualize this distribution and use it to assess how unusual our T_{obs} result of 1.84 years was relative to all the possibilities under permutations (under the null hypothesis). So we repeat the loop, now with $B = 1000$ and generate a histogram, density curve and summary statistics of the results:

```
B <- 1000
Tstar <- matrix(NA, nrow=B)
for (b in (1:B)){
  Tstar[b] <- diffmean(Years ~ shuffle(Attr), data=MockJury2)
}
hist(Tstar, label=T)
plot(density(Tstar), main="Density curve of Tstar")
```




```
favstats(Tstar)
```

```
##        min         Q1     median         Q3      max       mean        sd
##  -2.910384 -0.5099573 0.07681366 0.6102418 2.530583 0.04694168 0.8497364
```

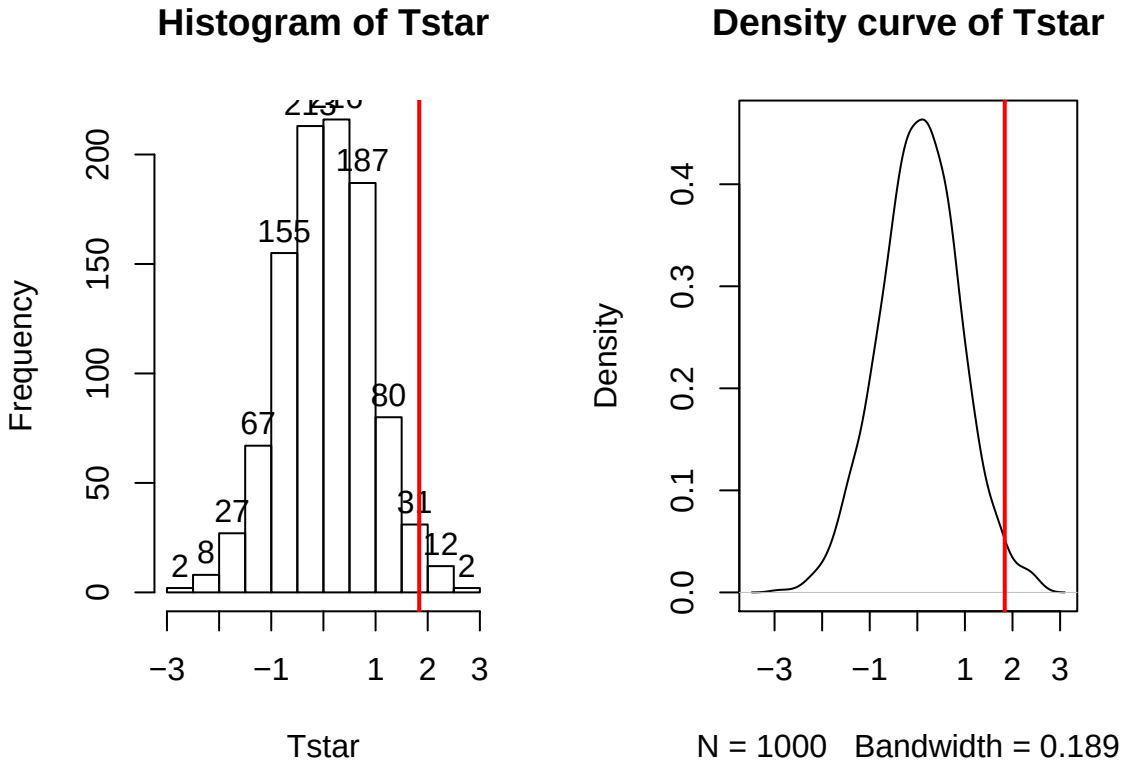
[11] We often say "under" in statistics and we mean "given that the following is true".

```
##     n missing
##  1000       0
```

Figure **??** contains visualizations of T^* and the `favstats` summary provides the related numerical summaries. Our observed T_{obs} of 1.84 seems fairly unusual relative to these results with only 20 T^* values over 2 based on the histogram. We need to make more specific comparisons of the permuted results versus our observed result to be able to clearly decide whether our observed result is really unusual.

To make the comparisons more concrete, first we can enhance the previous graphs by adding the value of the test statistic from the real data set, as shown in Figure **??**, using the `abline` function.

```
Tobs <- 1.837
hist(Tstar, labels=T)
abline(v=Tobs, lwd=2, col="red")
plot(density(Tstar),main="Density curve of Tstar")
abline(v=Tobs, lwd=2, col="red")
```



Second, we can calculate the exact number of permuted results that were larger than what we observed. To calculate the proportion of the 1,000 values that were larger than what we observed, we will use the `pdata` function. To use this function, we need to provide the distribution of values to compare to the cut-off (`Tstar`), the cut-off point (`Tobs`), and whether we want calculate the proportion that are below (left of) or above (right of) the cut-off (`lower.tail=F` option provides the proportion of values above the cutoff of interest).

```
pdata(Tstar, Tobs, lower.tail=F)
```

```
## [1] 0.02
```

The proportion of 0.02 tells us that 20 of the 1,000 permuted results (2%) were larger than what we observed. This type of work is how we can generate ***p-values*** using permutation distributions. P-values, as you should

remember, are the probability of getting a result as extreme as or more extreme than what we observed, given that the null is true. Finding only 20 permutations of 1,000 that were larger than our observed result suggests that it is hard to find a result like what we observed if there really were no difference, although it is not impossible.

When testing hypotheses for two groups, there are two types of alternative hypotheses, one-sided or two-sided. ***One-sided tests*** involve only considering differences in one-direction (like $\mu_1 > \mu_2$) and are performed when researchers can decide ***a priori***[12] which group should have a larger mean if there is going to be any sort of difference. In this situation, we did not know enough about the potential impacts of the pictures to know which group should be larger than the other so should do a two-sided test. It is important to remember that you can't look at the responses to decide on the hypotheses. It is often safer and more ***conservative***[13] to start with a ***two-sided alternative*** ($\mathbf{H_A}: \mu_1 \neq \mu_2$). To do a 2-sided test, find the area larger than what we observed as above. We also need to add the area in the other tail (here the left tail) similar to what we observed in the right tail. Some people suggest doubling the area in one tail but we will collect information on the number that were more extreme than the same value in the other tail. In other words, we count the proportion over 1.84 and below -1.84. So we need to also find how many of the permuted results were smaller than -1.84years to add to our previous proportion. Using `pdata -Tobs` as the cut-off and `lower.tail=T` provides this result:

```
pdata(Tstar, -Tobs, lower.tail=T)
```

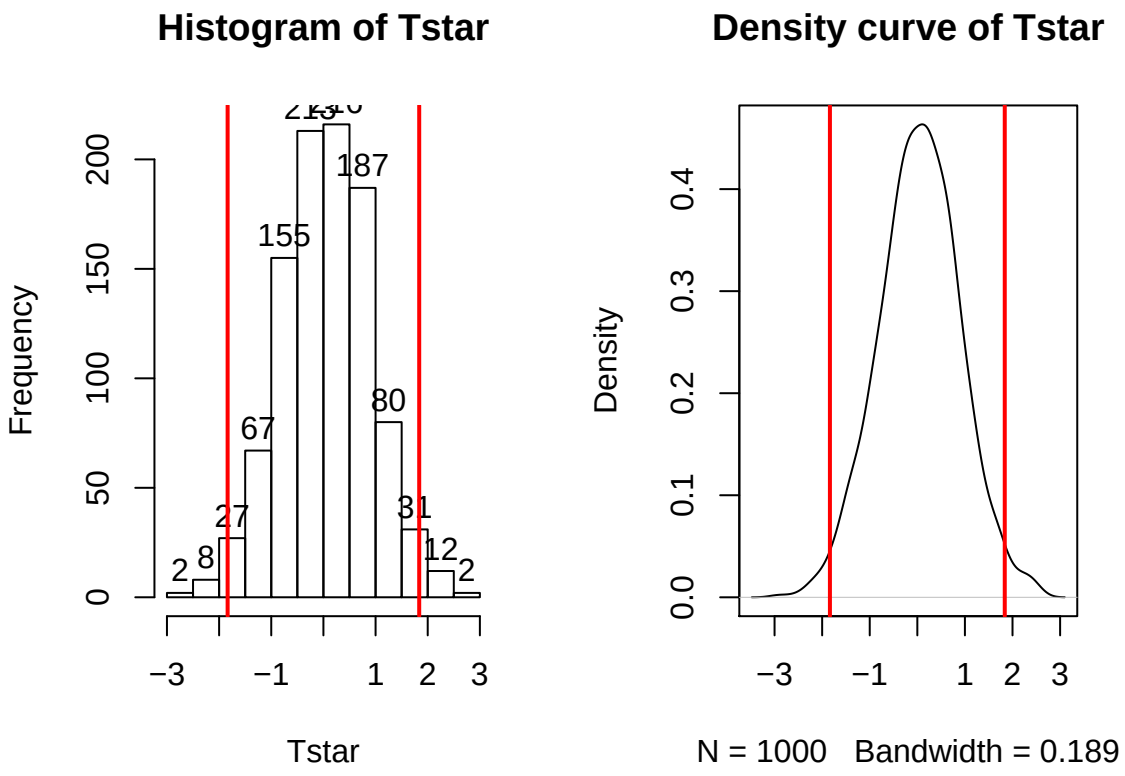
```
## [1] 0.014
```

So the p-value to test our null hypothesis of no difference in the true means between the groups is 0.02 + 0.014, providing a p-value of 0.034. Figure **??** shows both cut-offs on the histogram and density curve.

```
hist(Tstar, labels=T)
abline(v=c(-1,1)*Tobs, lwd=2, col="red")
plot(density(Tstar),main="Density curve of Tstar")
abline(v=c(-1,1)*Tobs, lwd=2, col="red")
```

[12] This is a fancy way of saying "in advance", here in advance of seeing the observations.

[13] Statistically, a conservative method is one that provides less chance of rejecting the null hypothesis in comparison to some other method or less than some pre-defined standard.



In general, the ***one-sided test p-value*** is the proportion of the permuted results that are more extreme than observed in the direction of the *alternative* hypothesis (lower or upper tail, remembering that this also depends on the direction of the difference taken). For the 2-sided test, the p-value is the proportion of the permuted results that are *less than the negative version of the observed statistic and greater than the positive version of the observed statistic*. Using absolute values (| |), we can simplify this: the ***two-sided p-value*** is the *proportion of the |permuted statistics| that are larger than |observed statistic|*. This will always work and finds areas in both tails regardless of whether the observed statistic is positive or negative. In R, the `abs` function provides the ***absolute value*** and we can again use `pdata` to find our p-value in one line of code:

```
pdata(abs(Tstar), abs(Tobs), lower.tail=F)
```

```
## [1] 0.034
```

We will discuss the choice of ***significance level*** below, but for the moment, assume that α is chosen to be our standard value of 0.05. Since the p-value is smaller than α, this suggests that we can ***reject the null hypothesis*** and conclude that there is evidence of some difference in the true mean sentences given between the two types of pictures.

Before we move on, let's note some interesting features of the permutation distribution of the difference in the sample means shown in Figure **??**.

1. It is basically centered at 0. Since we are performing permutations assuming the null model is true, we are assuming that $\mu_1 = \mu_2$ which implies that $\mu_1 - \mu_2 = 0$. This also suggests that 0 should be the center of the permutation distribution and it was.
2. It is approximately normally distributed. This is due to the ***Central Limit Theorem***[14], where the ***sampling distribution*** (distribution of all possible results for samples of this size) of the difference in

[14]We'll leave the discussion of the CLT to your previous stat coursework or an internet search. Remember that it has something to do with distributions looking more normal as the sample size increases.

sample means ($\bar{x}_1 - \bar{x}_2$) becomes more and normally distributed as the sample sizes increase. With 38 and 37 observations in the groups, we are likely to have a relatively normal looking distribution of the difference in the sample means. This result will allow us to use a parametric method to approximate this sampling distribution under the null model if some assumptions are met, as we'll discuss below.

3. Our observed difference in the sample means (1.84 years) is a fairly unusual result relative to the rest of these results but there are some permuted data sets that produce more extreme differences in the sample means. When the observed differences are really large, we may not see any permuted results that are as extreme as what we observed. When `pdata` gives you 0, the p-value should be reported to be smaller than 0.0001 (**not 0!**) since it happened in less than 1 in 1000 tries but does occur once – in the actual data set.

4. Since our null model is not specific about the direction of the difference, considering a result like ours but in the other direction (-1.84 years) needs to be included. The observed result seems to put about the same area in both tails of the distribution but it is not exactly the same. The small difference in the tails is a useful aspect of this approach compared to the parametric method discussed below as it accounts for slight asymmetry in the sampling distribution.

Earlier, we decided that the p-value was small enough to reject the null hypothesis since it was smaller than our chosen level of significance. In this course, you will often be allowed to use your own judgment about an appropriate significance level in a particular situation (in other words, if we forget to tell you an α -level, you can still make a decision using a reasonably selected significance level). Remembering that the p-value is the probability you would observe a result like you did (or more extreme), assuming the null hypothesis is true, this tells you that the smaller the p-value is, the more evidence you have against the null. The next section provides a more formal review of the hypothesis testing infrastructure, terminology, and some of things that can happen when testing hypotheses.

2.5 Hypothesis testing (general)

In hypothesis testing, it is formulated to answer a specific question about a population or true parameter(s) using a statistic based on a data set. In your previous statistics course, you (hopefully) considered one-sample hypotheses about population means and proportions and the two sample mean situation we are focused on here. Our hypotheses relate to trying to answer the question about whether the population mean sentences between the two groups are different, with an initial assumption of no difference.

Hypothesis testing is much like a criminal trial where you are in the role of a jury member or judge, if no jury is present. Initially, the defendant is assumed innocent. In our situation, the true means are assumed to be equal between the groups. Then evidence is presented and, as a juror, you analyze it. In statistical hypothesis testing, data are collected and analyzed. Then you have to decide if we had "enough" evidence to reject the initial assumption ("innocence" that is initially assumed). To make this decision, you want to have previously decided on the standard of evidence required to reject the initial assumption. In criminal cases, "beyond a reasonable doubt" is used. Wikipedia's definition (https://en.wikipedia.org/wiki/Reasonable_doubt) suggests that this standard is that "there can still be a doubt, but only to the extent that it would not affect a reasonable person's belief regarding whether or not the defendant is guilty". In civil trials, a lower standard called a "preponderance of evidence" is used. Based on that defined and pre-decided (*a priori*) measure, you decide that the defendant is guilty or not guilty. In statistics, we compare our p-value to a significance level, α, which is most of the time selected to be 5%. If our p-value is less than α, we reject the null hypothesis. The choice of the significance level is like the variation in standards of evidence between criminal and civil trials – and in all situations everyone should know the standards required for rejecting the initial assumption before any information is "analyzed". Once someone is found guilty, then there is the matter of sentencing which is related to the impacts ("size") of the crime. In statistics, this is similar to the estimated size of differences and the related judgments about whether the differences are practically important or not. If the crime is proven beyond a reasonable doubt but it is a minor crime, then the sentence will be small. With the same level of evidence and a more serious crime, the sentence will be more dramatic.

Table 2.2: Scope of inference summary.

Random Sampling/Random Assignment	**
Random Sampling (RS) – Yes (or some method that results in a representative sample of population of interest)	Be
Random Sampling (RS) – No (usually a convenience sample)	Ca

There are some important aspects of the testing process to note that inform how we interpret statistical hypothesis test results. When someone is found "not guilty", it does not mean "innocent", it just means that there was not enough evidence to find the person guilty "beyond a reasonable doubt". Not finding enough evidence to reject the null hypothesis does not imply that the true means are equal, just that there was not enough evidence to conclude that they were different. There are many potential reasons why we might fail to reject the null, but the most common one is that our sample size was too small (which is related to having too little evidence).

Throughout this material, we will continue to re-iterate the distinctions between parameters and statistics and want you to be clear about the distinctions between estimates based on the sample and inferences for the population or true values of the parameters of interest. Remember that statistics are summaries of the sample information and parameters are characteristics of populations (which we rarely know). In the two-sample mean situation, the sample means are always at least a little different – that is not an interesting conclusion. What is interesting is whether we have enough evidence to prove that the population or true means differ "beyond a reasonable doubt".

The scope of any inferences is constrained based on whether there is a
random sample (RS) and/or ***random assignment*** (RA). Table 2.2 contains the four possible combinations of these two characteristics of a given study. Random assignment allows for causal inferences for differences that are observed – the difference in treatment levels causes differences in the mean responses. Random sampling (or at least some sort of representative sample) allows inferences to be made to the population of interest. If we do not have RA, then causal inferences cannot be made. If we do not have a representative sample, then our inferences are limited to the sampled subjects.

A simple example helps to clarify how the scope of inference can change. Suppose we are interested in studying the GPA of students. If we had taken a random sample from, say, the STAT 217 students in a given semester, our scope of inference would be the population of 217 students in that semester. If we had taken a random sample from the entire MSU population, then the inferences would be to the entire MSU population in that semester. These are similar types of problems but the two populations are very different and the group you are trying to make conclusions about should be noted carefully in your results – it does matter! If we did not have a representative sample, say the students could choose to provide this information or not, then we can only make inferences to volunteers. These volunteers might differ in systematic ways from the entire population of STAT 217 students so we cannot safely extend our inferences beyond the group that volunteered.

To consider the impacts of RA versus observational studies, we need to be comparing groups. Suppose that we are interested in differences in the mean GPAs for different sections of STAT 217 and that we take a random sample of students from each section and compare the results and find evidence of some difference. In this scenario, we can conclude that there is some difference in the population of STAT 217 students but we can't say that being in different sections caused the differences in the mean GPAs. Now suppose that we randomly assigned every 217 student to get extra training in one of three different study techniques and found evidence of differences among the training methods. We could conclude that the training methods caused the differences in these students. These conclusions would only apply to STAT 217 students and could not be generalized to a larger population of students. If we took a random sample of STAT 217 students (say only 10 from each section) and then randomly assigned them to one of three training programs and found evidence of differences, then we can say that the training programs caused the differences. We can also say that we have evidence that those differences pertain to the population of STAT 217 students. This seems similar to the scenario where all 217 students participated in the training programs except that by using random sampling, only a fraction of the population needs to actually be studied to make inferences to the

Table 2.3: Table of decisions and truth scenarios in a hypothesis testing situation. But we never know the truth in a real situation.

	$\mathbf{H_0}$ **True**	$\mathbf{H_0}$ **False**
FTR $\mathbf{H_0}$	Correct decision	Type II error
Reject $\mathbf{H_0}$	Type I error	Correct decision

entire population of interest – saving time and money.

A quick summary of the terminology of hypothesis testing is useful at this point. The ***null hypothesis*** (H_0) states that there is no difference or no relationship in the population. This is the statement of no effect or no difference and the claim that we are trying to find evidence against. In this chapter, H_0: $\mu_1 = \mu_2$. When doing two-group problems, you always need to specify which group is 1 and which one is 2 because the order does matter. The ***alternative hypothesis*** (H_1 or H_A) states a specific difference between parameters. This is the research hypothesis and the claim about the population that we hope to demonstrate is more reasonable to conclude than the null hypothesis. In the two-group situation, we can have ***one-sided alternatives*** H_A:$\mu_1 > \mu_2$ (greater than) or H_A:$\mu_1 < \mu_2$ (less than) or, the more common, ***two-sided alternative*** H_A:$\mu_1 \neq \mu_2$ (not equal to). We usually default to using two-sided tests because we often do not know enough to know the direction of a difference in advance, especially in more complicated situations. The ***sampling distribution under the null*** is the distribution of all possible values of a statistic under H_0 is true. It is used to calculate the ***p-value*** , the probability of obtaining a result as extreme or more extreme than what we observed given that the null hypothesis is true. We will find sampling distributions using ***nonparametric*** approaches (like the permutation approach used above) and ***parametric*** methods (using "named" distributions like the t, F, and χ^2).

Small p-values are evidence against the null hypothesis because the observed result is unlikely due to chance if H_0 is true. Large p-values provide no evidence against H_0 but do not allow us to conclude that there is no difference. The ***level of significance*** is an *a priori* definition of how small the p-value needs to be to provide "enough" (sufficient) evidence against H_0. This is most useful to prevent sliding the standards after the results are found. We compare the p-value to the level of significance to decide if the p-value is small enough to constitute sufficient evidence to reject the null hypothesis. We use α to denote the level of significance and most typically use 0.05 which we refer to as the 5% significance level. We compare the p-value to this level and make a decision. The two options for *decisions* are to either *reject the null hypothesis* if the p-value $\leq \alpha$ or *fail to reject the null hypothesis* if the p-value $> \alpha$. When interpreting hypothesis testing results, remember that the p-value is a measure of how unlikely the observed outcome was, assuming that the null hypothesis is true. It is **NOT** the probability of the data or the probability of either hypothesis being true. The p-value, simply, is a measure of evidence against the null hypothesis.

The specific definition of α is that it is the probability of rejecting H_0 when H_0 is true, the probability of what is called a ***Type I error***. Type I errors are also called ***false rejections***. In the two-group mean situation, a Type I error would be concluding that there is a difference in the true means between the groups when none really exists in the population. In the courtroom setting, this is like falsely finding someone guilty. We don't want to do this very often, so we use small values of the significance level, allowing us to control the rate of Type I errors at α. We also have to worry about **Type II errors**, which are failing to reject the null hypothesis when it's false. In a courtroom, this is the same as failing to convict a truly guilty person. This most often occurs due to a lack of evidence. You can use the Table 2.3 to help you remember all the possibilities.

In comparing different procedures, there is an interest in studying the rate or probability of Type I and II errors. The probability of a Type I error was defined previously as α, the significance level. The ***power*** of a procedure is the probability of rejecting the null hypothesis when it is false. Power is defined as

$$\text{Power} = 1 - \text{Probability(Type II error)} \ = \text{Probability(Reject) } H_0 | H_0 \text{ is false}),$$

or, in words, the probability of detecting a difference when it actually exists. We want to use a statistical procedure that controls the Type I error rate at the pre-specified level and has high power to detect false null alternatives. Increasing the sample size is one of the most commonly used methods for increasing the power in a given situation. Sometimes we can choose among different procedures and use the power of the procedures to help us make that selection. Note that there are many ways H_0 false and the power changes based on how false the null hypothesis actually is. To make this concrete, suppose that the true mean sentences differed by either 1 or 20 years in previous example. The chances of rejecting the null hypothesis are much larger when the groups actually differ by 20 years than if they differ by just 1 year.

After making a decision (was there enough evidence to reject the null or not), we want to make the conclusions specific to the problem of interest. If we reject H_0, then we can conclude that there was sufficient evidence at the α-level that the null hypothesis is wrong (and the results point in the direction of the alternative). If we fail to reject H_0 (FTR H_0), then we can conclude that there was insufficient evidence at the α-level to say that the null hypothesis is wrong. We are **NOT** saying that the null is correct and we **NEVER** accept the null hypothesis. We just failed to find enough evidence to say it's wrong. If we find sufficient evidence to reject the null, then we need to revisit the method of data collection and design of the study to discuss scope of inference. Can we discuss causality (due to RA) and/or make inferences to a larger group than those in the sample (due to RS)?

To perform a hypothesis test, there are some steps to remember to complete to make sure you have thought through all the aspects of the results.

Outline of 6+ steps to perform a Hypothesis Test
Isolate the claim to be proved, method to use (define a test statistic T), and significance level.
1. Write the null and alternative hypotheses,
2. Assess the "Validity Conditions" for the procedure being used (discussed below),
3. Find the value of the appropriate test statistic,
4. Find the p-value,
5. Make a decision, and
6. Write a conclusion specific to the problem, including scope of inference discussion.

2.6 Connecting randomization (nonparametric) and parametric tests

In developing statistical inference techniques, we need to define the test quantity of interest. To compare the means of two groups, a statistic is needed that measures their differences. In general, for comparing two groups, the choices are simple – a difference in the means often works well and is a natural choice. There are other options such as tracking the ratio of means or possibly the difference in medians. Instead of just using the difference in the means, we also could "standardize" the difference in the means by dividing by an appropriate quantity that reflects the variation in the difference in the means. All of these are valid and can sometimes provide similar results - it ends up that there are many possibilities for testing using the randomization (nonparametric) techniques introduced previously. Parametric statistical methods focus on means because the statistical theory surrounding means is quite a bit easier (not easy, just easier) than other options but there are just a couple of test statistics that you can use and end up with named distributions to use for generating inferences. Randomization techniques allow inference for other quantities but our focus here will be on using randomization for inferences on means to see the similarities with the more traditional parametric procedures.

In two-sample mean situations, instead of working just with the difference in the means, we often calculate a test statistic that is called the
equal variance two-independent samples t-statistic. The test statistic is

$$t = \frac{\bar{x}_1 - \bar{x}_2}{s_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}},$$

where s_1^2 and s_2^2 are the sample variances for the two groups, n_1 and n_2 are the sample sizes for the two groups, and the ***pooled sample standard deviation*** ,

$$s_p = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}}.$$

The t-statistic keeps the important comparison between the means in the numerator that we used before and standardizes (re-scales) that difference so that t will follow a t-distribution (a parametric "named"distribution) if certain assumptions are met. But first we should see if standardizing the difference in the means had an impact on our permutation test results. Instead of using the `diffmean` function, we will use the `t.test` function (see its full use below) and have it calculate the formula for t for us. The R code "`$statistic`" is basically a way of extracting just the number we want to use for T from a larger set of output the `t.test` function wants to provide you. We will see below that `t.test` switches the order of the difference (now it is *Average - Unattractive*) – always carefully check for the direction of the difference in the results. Since we are doing a two-sided test, the code resembles the permutation test code in Section 2.4 with the new t-statistic replacing the difference in the sample means that we used before.

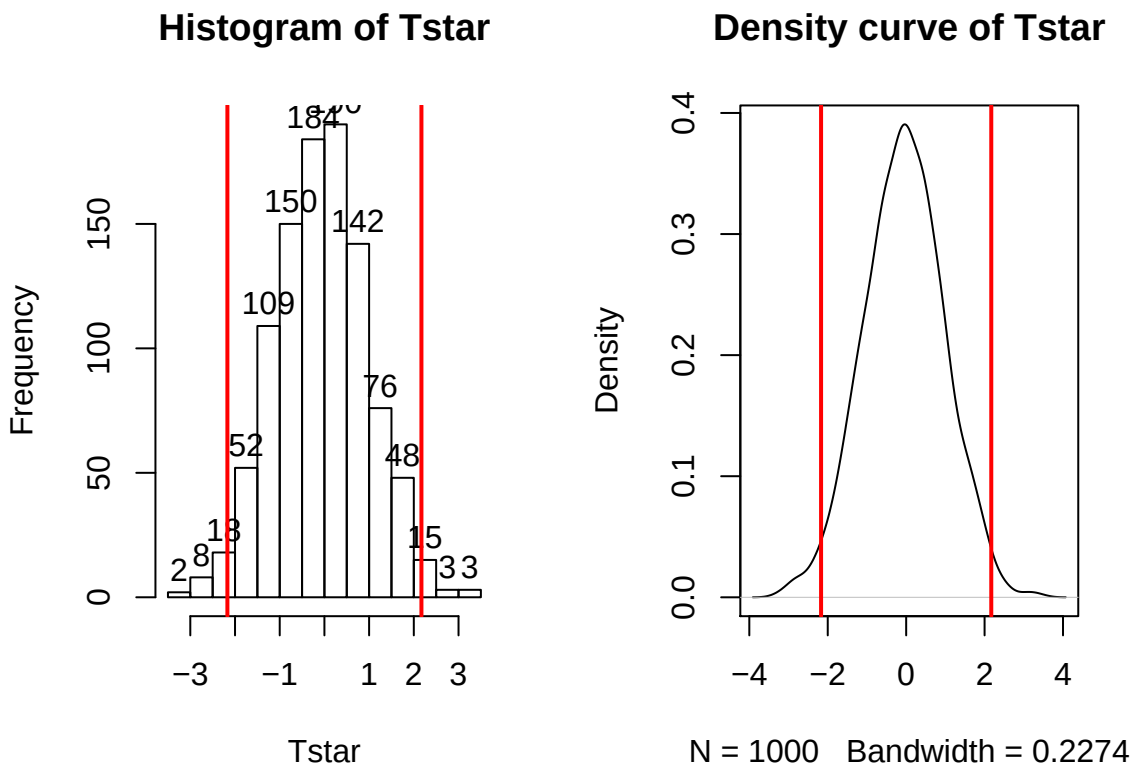
The permutation distribution in Figure **??** looks similar to the previous results with slightly different x-axis scaling. The the proportion of permuted results that were more extreme than the observed result was 0.031. This difference is due to a different set of random permutations being selected. If you run permutation code, you will often get slightly different results each time you run it. If you are uncomfortable with the variation in the results, you can run more than $B = 1{,}000$ permutations (say 10,000) and the variability in the resulting p-values will be reduced further. Usually this uncertainty will not cause any substantive problems – but do not be surprised if your results vary from a colleagues if you are both analyzing the same data set or if you re-run your permutation code.

```
Tobs <- t.test(Years ~ Attr, data=MockJury2, var.equal=T)$statistic
Tobs
```

```
##        t
## -2.17023
```

```
Tstar <- matrix(NA, nrow=B)
for (b in (1:B)){
  Tstar[b] <- t.test(Years ~ shuffle(Attr), data=MockJury2, var.equal=T)$statistic
}
hist(Tstar, labels=T)
abline(v=c(-1,1)*Tobs, lwd=2, col="red")
plot(density(Tstar), main="Density curve of Tstar")
abline(v=c(-1,1)*Tobs, lwd=2, col="red")
pdata(abs(Tstar),abs(Tobs),lower.tail=F)
```

```
##     t
## 0.031
```



The parametric version of these results is based on using what is called the ***two-independent sample t-test***. There are actually two versions of this test, one that assumes that variances are equal in the groups and one that does not. There is a rule of thumb that if the **ratio of the larger standard deviation over the smaller standard deviation is less than 2, the equal variance procedure is OK**. It ends up that this assumption is less important if the sample sizes in the groups are approximately equal and more important if the groups contain different numbers of observations. In comparing the two potential test statistics, the procedure that assumes equal variances has a complicated denominator (see the formula above for t involving s_p) but a simple formula for ***degrees of freedom*** (***df***) for the t-distribution ($df = n_1 + n_2 - 2$) that approximates the distribution of the test statistic, t, under the null hypothesis. The procedure that assumes unequal variances has a simpler test statistic and a very complicated degrees of freedom formula. The equal variance procedure is most similar to the ANOVA methods we will consider in Chapters 2 and 3 so that will be our focus for the two group problem. Fortunately, both of these methods are readily available in the `t.test` function in R if needed.

If the assumptions for the equal variance t-test are met and the null hypothesis is true, then the sampling distribution of the test statistic should follow a t-distribution with $n_1 + n_2 - 2$ degrees of freedom. The ***t-distribution*** is a bell-shaped curve that is more spread out for smaller values of degrees of freedom as shown in Figure 2.6. The t-distribution looks more and more like a ***standard normal distribution*** ($N(0, 1)$) as the degrees of freedom increase.

To get the p-value for the parametric t-test, we need to calculate the test statistic and df, then look up the areas in the tails of the t-distribution relative to the observed t-statistic. We'll learn how to use R to do this below, but for now we will allow the `t.test` function to take care of this for us. The `t.test` function uses our formula notation (`Years ~ Attr`) and then `data=...` as we saw before for making plots. To get the equal-variance test result, the `var.equal=T` option needs to be turned on. Then `t.test` provides us with lots of useful output. The three results we've been discussing are highlighted in the output below – the test statistic value (-2.17), $df = 73$, and the p-value, from the t-distribution with 73 degrees of freedom, of 0.033.

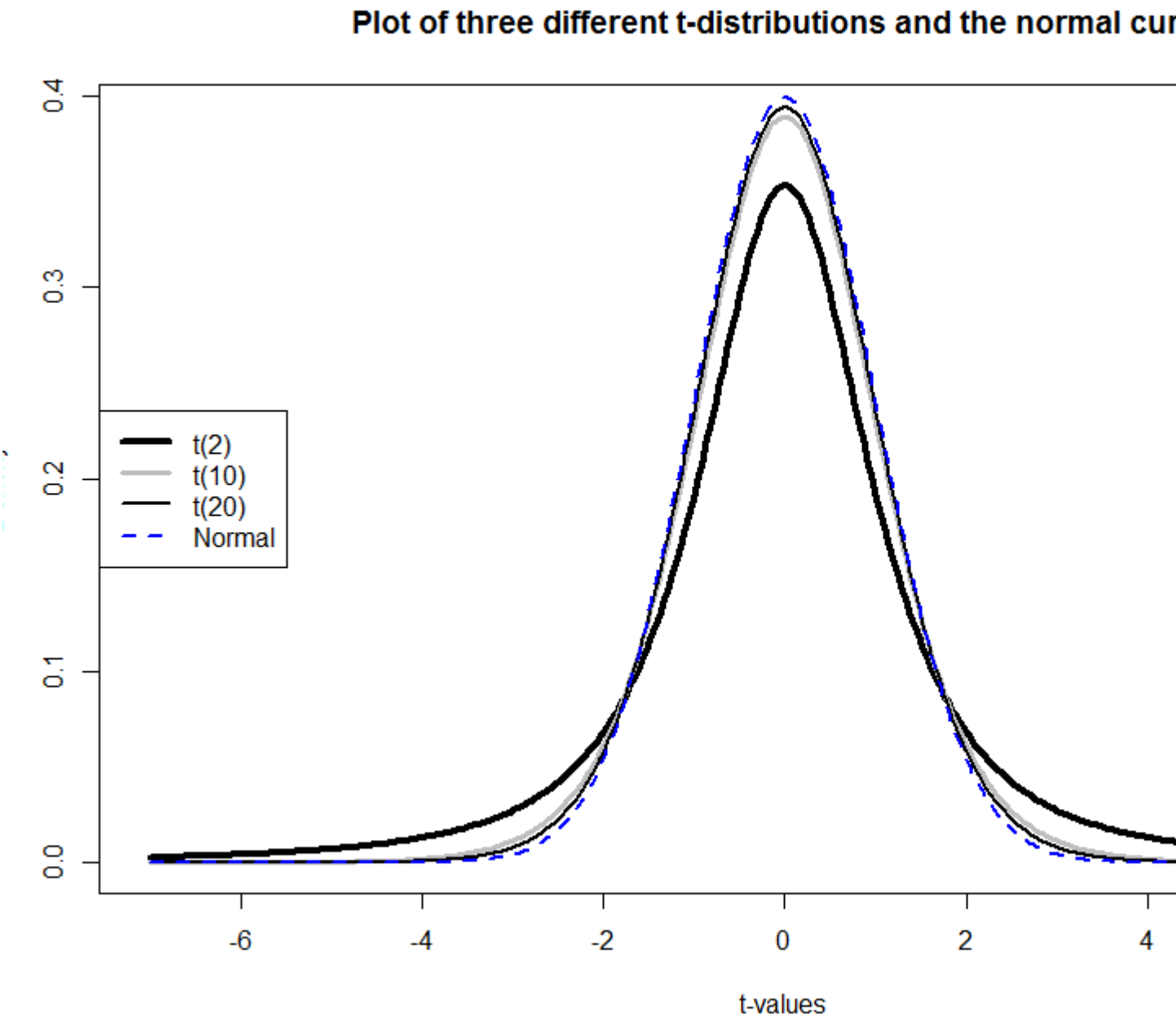


Figure 2.6: Plots of t and normal distributions

```
t.test(Years ~ Attr, data=MockJury2, var.equal=T)
```

```
##
##  Two Sample t-test
##
## data:  Years by Attr
## t = -2.1702, df = 73, p-value = 0.03324
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.5242237 -0.1500295
## sample estimates:
##      mean in group Average mean in group Unattractive
##                   3.973684                   5.810811
```

So the parametric t-test gives a p-value of 0.033 from a test statistic of -2.1702. The negative sign on the test statistic occurred because the function took *Average - Unattractive* which is the opposite direction as `diffmean`. The p-value is very similar to the two permutation results found before. The reason for this similarity is that the permutation distribution with 73 degrees of freedom. Figure 2.7 shows how similar the two distributions happened to be here.

In your previous statistics course, you might have used an applet or a table to find p-values such as what was provided in the previous R output. When not directly provided in the output of a function, R can be used to look up p-values[15] from named distributions such as the t-distribution. In this case, the distribution of the test statistic under the null hypothesis is a $t(73)$ or a t with 73 degrees of freedom. The `pt` function is used to get p-values from the t-distribution in the same manner that `pdata` could help us to find p-values from the permutation distribution. We need to provide the `df=...` and specify the tail of the distribution of interest using the `lower.tail` option along with the cutoff of interest. If we want the area to the left of -2.17:

```
pt(-2.1702, df=73, lower.tail=T)
```

```
## [1] 0.01662286
```

And we can double it to get the p-value that `t.test` provided earlier, because the t-distribution is symmetric:

```
2*pt(-2.1702, df=73, lower.tail=T)
```

```
## [1] 0.03324571
```

More generally, we could always make the test statistic positive using the absolute value, find the area to the right of it, and then double that for a two-sided test p-value:

```
2*pt(abs(-2.1702), df=73, lower.tail=T)
```

```
## [1] 1.966754
```

Permutation distributions do not need to match the named parametric distribution to work correctly, although this happened in the previous example. The parametric certain conditions to be met for the sampling distribution of the statistic to follow the named distribution and provide accurate p-values. The conditions for the equal variance t-test are:

1. **Independent observations**: Each observation obtained is unrelated to all other observations. To assess this, consider whether anything in the data collection might lead to clustered or related observations that are un-related to the differences in the groups. For example, was the same person measured more than once?[16]

[15] On exams, you will be asked to describe the area of interest, sketch a picture of the area of interest, and/or note the distribution you would use.

[16] In some studies, the same subject might be measured in both conditions and this violates the assumptions of this procedure.

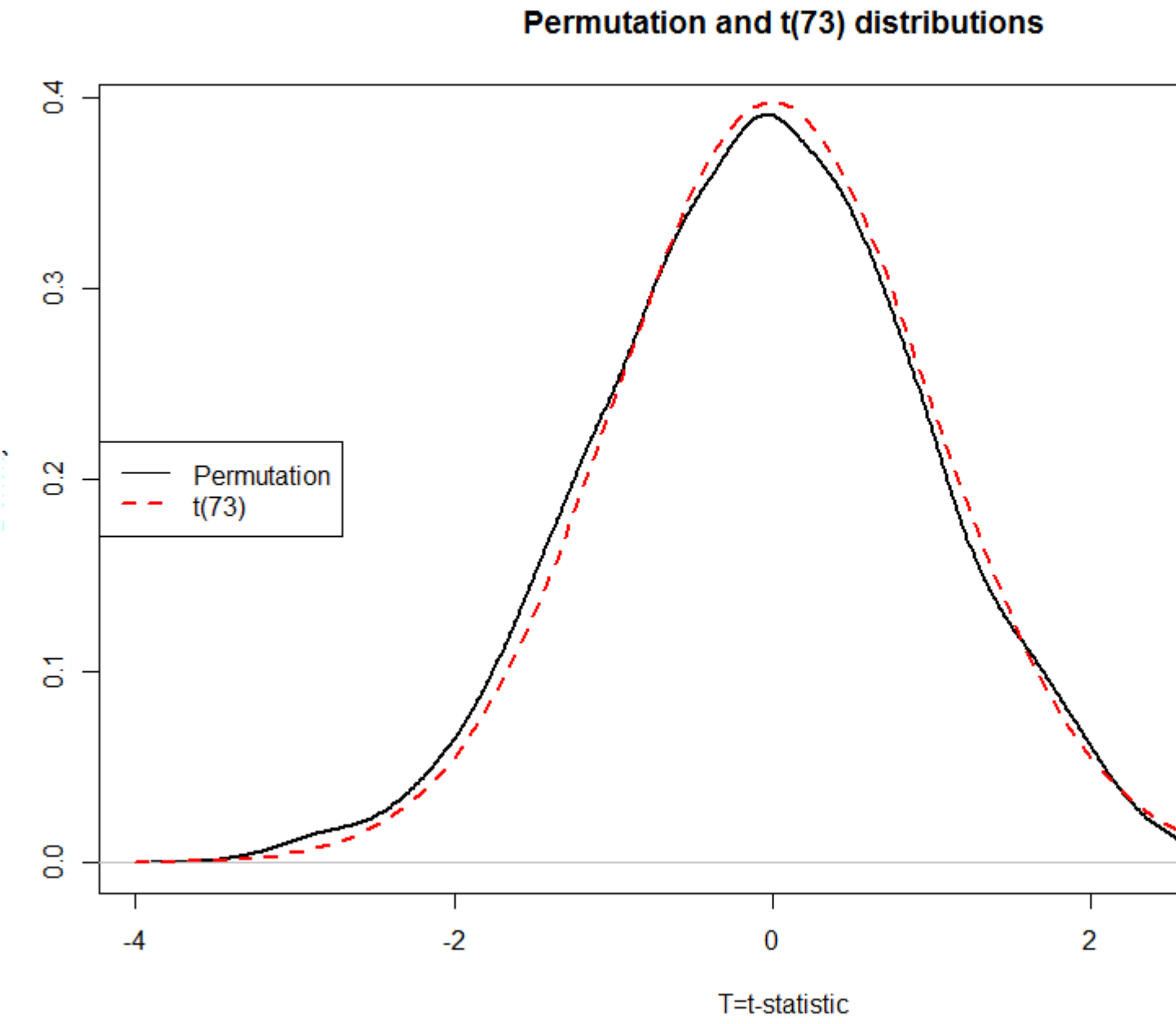


Figure 2.7: Plot of permutation and t distribution with $df = 73$.

2. **Equal variances** in the groups (because we used a procedure that assumes equal variances! – there is another procedure that allows you to relax this assumption if needed...). To assess this, compare the standard deviations and variability in the beanplots and see if they look noticeably different. Be particularly critical of this assessment if the sample sizes differ greatly between groups.
3. **Normal distributions** of the observations in each group. We'll learn more diagnostics later, but the boxplots and beanplots are a good place to start to help you look for skews or outliers, which were both present here. If you find skew and/or outliers, that would suggest a problem with the assumption of normality as normal distributions are symmetric and extreme observations occur very rarely.

For the permutation test, we relax the third condition and replace it with:

3. ***Similar distributions between the groups:*** The permutation approach allows valid inferences as long as the two groups have similar shapes and only possibly differ in their centers. In other words, the distributions need not look normal for the procedure to work well, but they do need to look similar.

In the prisoner "juror" study, we can assume that the independent observation condition is met because there is no information suggesting that the same subjects were measured more than once or that some other type of grouping in the responses was present (like the subjects were divided in groups and placed in rooms to discuss their responses prior to submitting them). The equal variance condition might be violated. The variances need not be equal as the procedure can still provide reasonable results with some violation of this assumption. The standard deviations are 2.8 vs 4.4, so this difference is not "large" according to the rule of thumb noted above. It is, however, close to being considered problematic. It would be difficult to reasonably assume that the normality condition is met here (Figure **??** with clear right skews in both groups and potential outliers which causes concerns for (3) for the parametric procedure. The shapes look similar for the two groups so there is less reason to be concerned with using the permutation approach based on its version of (3) above.

The permutation approach is resistant to impacts of violations of the normality assumption. It is not resistant to impact of violations of any of the other assumptions. In fact, it can be quite sensitive to unequal variances as it will detect differences in the variances of the groups instead of differences in the means. Its scope of inference is the same as the parametric approach and can lead to similarly inaccurate conclusions in the presence of non-independent observations as for the parametric approach. In this example, we discover that parametric and permutation approaches provide very similar inferences.

2.7 Second example of permutation tests

In every chapter, we will follow the first example used to motivate and explain the methods with a "worked" example where we focus just on the results. In a previous semester, some of the STAT 217 students ($\boldsymbol{n}$=79) provided information on their *Sex*, *Age*, and current cumulative *GPA*. We might be interested in whether Males and Females had different average GPAs. First, we can take a look at the difference in the responses by groups based on the output and as displayed in Figure 2.8.

```
s217 <- read.csv("http://www.math.montana.edu/courses/s217/documents/s217.csv")
require(mosaic)
par(mfrow=c(1,2))
boxplot(GPA~Sex, data=s217)
require(beanplot)
beanplot(GPA~Sex, data=s217, log="", col="lightblue", method="jitter")
```

```
mean(GPA~Sex, data=s217)
```

```
##        F        M
## 3.338378 3.088571
```

```
favstats(GPA~Sex, data=s217)
```

```
##   Sex  min  Q1 median   Q3 max     mean       sd  n missing
```

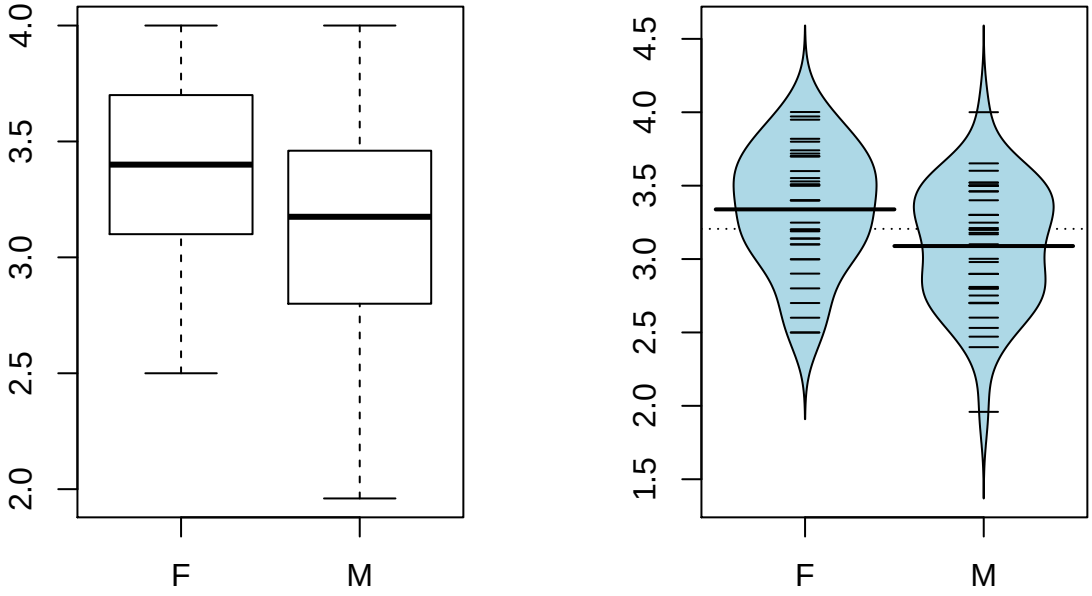


Figure 2.8: Side-by-side boxplot and beanplot of GPAs of STAT 217 students by sex.

```
## 1   F 2.50 3.1  3.400 3.70   4 3.338378 0.4074549 37       0
## 2   M 1.96 2.8  3.175 3.46   4 3.088571 0.4151789 42       0
```

In these data, the distributions of the GPAs look to be left skewed but maybe not as dramatically as the responses were right-skewed in the previous example. The Female GPAs look to be slightly higher than for Males (0.25 GPA difference in the means) but is that a "real" difference? We need our inference tools to more fully assess these differences.

```
diffmean(GPA~Sex, data=s217)
```

```
##   diffmean
## -0.2498069
```

First, we can try the parametric approach:

```
t.test(GPA~Sex, data=s217, var.equal=T)
```

```
##
##  Two Sample t-test
##
## data:  GPA by Sex
## t = 2.6919, df = 77, p-value = 0.008713
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.06501838 0.43459552
## sample estimates:
## mean in group F mean in group M
##        3.338378        3.088571
```

So the test statistic was observed to be $t = 2.69$ and it hopefully follows a $t(77)$ distribution under the null hypothesis. This provides a p-value of 0.008713 that we can trust if all the conditions to use this procedure are met. Compare these results to the permutation approach, which relaxes that normality assumption, with the results that follow. In the permutation test, $T = 2.692$ and the p-value is 0.005 which is a little smaller than the result provided by the parametric approach. The agreement of the two approaches, again, provides some re-assurance about the use of either approach.

```
Tobs <- t.test(GPA~Sex, data=s217, var.equal=T)$statistic
Tstar <- matrix(NA, nrow=B)
for (b in (1:B)){
  Tstar[b] <- t.test(GPA~shuffle(Sex), data=s217, var.equal=T)$statistic
}
```

```
par(mfrow=c(1,2))
hist(Tstar,labels=T)
abline(v=c(-1,1)*Tobs, lwd=2, col="red")
plot(density(Tstar), main="Density curve of Tstar")
abline(v=c(-1,1)*Tobs, lwd=2, col="red")
```

```
pdata(abs(Tstar),abs(Tobs),lower.tail=F)
```

```
##     t
## 0.005
```

Here is a full write-up of the results using all 6+ hypothesis testing steps, using the permutation results:

0. *Isolate the claim to be proved and method to use (define a test statistic T)* We want to test for a difference in the means between males and females and will use the equal-variance two-sample t-test statistic to compare them, making a decision at the 5% significance level.

1. Write the null and alternative hypotheses

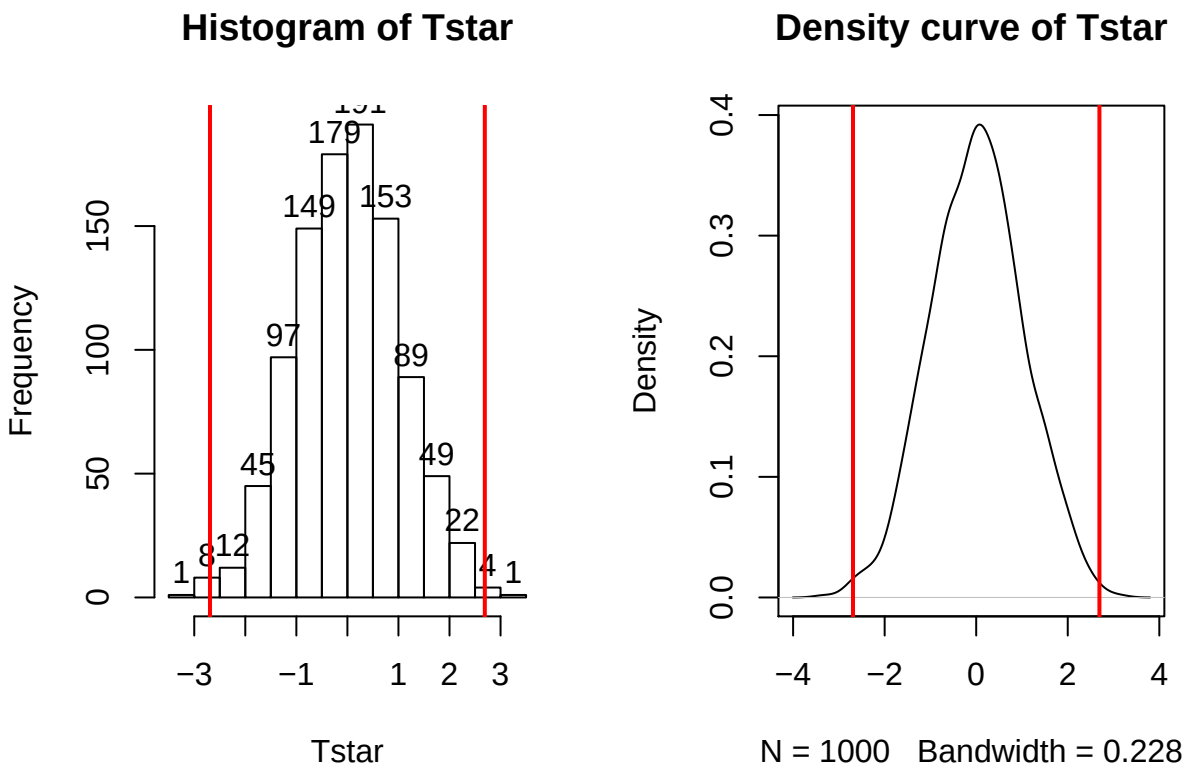


Figure 2.9: Histogram and density curve of permutation distribution of test statistic for STAT 217 GPAs.

- $H_0 : \mu_{male} = \mu_{female}$
  - where μ_{male} is the true mean GPA for males and μ_{female} is true mean GPA for females.
- $H_A : \mu_{male} \neq \mu_{female}$

2. Check conditions for the procedure being used
   - **Independent observations condition**: It appears that this assumption is met because there is no reason to assume any clustering or grouping of responses that might create dependence in the observations. The only possible consideration is that the observations were taken from different sections and there could be some differences between the sections. However, for overall GPA this not likely to be a big issue. The only way this could create a violation here is if certain sections tended to attract students with different GPA levels (such as the 9 am section had the best/worst GPA students...).
   - **Equal variance condition** : There is a small difference in the range of the observations in the two groups but the standard deviations are very similar so there is no evidence that this condition is violated.
   - **Similar distribution condition**: Based on the side-by-side boxplots and beanplots, it appears that both groups have slightly left-skewed distributions, which could be problematic for the parametric approach, but the permutation approach condition is not violated since the distributions look to have fairly similar shapes.
3. Find the value of the appropriate test statistic
   - $T = 2.69$ from the previous R output.
4. Find the p-value
   - p-value=0.005 from the permutation distribution results.
   - This means that there is about a 0.5% chance we would observe a difference in mean GPA (female-male or male-female) of 0.25 points or more if there in fact no difference in true mean GPA between females and males in STAT 217 in a particular semester.
5. Decision
   - Since the p-value is "small" (*a priori* 5% significance level selected), we can reject the null hypothesis.
6. Conclusion and scope of inference, specific to the problem
   - There is strong evidence against the null hypothesis of no difference in the true mean GPA between males and females for the STAT 217 students in this semester and so we conclude that there is evidence of a difference in the mean GPAs between males and females in STAT 217 students.
   - Because this was not a randomized experiment, we can't say that the difference in sex causes the difference in mean GPA and because it was not a random sample from a larger population, our inferences only pertain the STAT 217 students that responded to the survey in that semester.

2.8 Confidence intervals and bootstrapping

Randomly shuffling the treatments between the observations is like randomly sampling the treatments without replacement. In other words, we randomly sample one observation at a observations. This provides us with a technique for testing hypotheses because it provides new splits of the observations into groups that are as interesting as what we observed if the null hypothesis is assumed true. In most situations, we also want to estimate parameters of interest and provide ***confidence intervals*** for those parameters (an interval where we are ___% ***confident*** that the true parameter lies). As before, there are two options we will consider

– a parametric and a nonparametric approach. The nonparametric approach will be using what is called ***bootstrapping*** and draws its name from "pull yourself up by your bootstraps" where you improve your situation based on your own efforts. In statistics, we make our situation or inferences better by re-using the observations we have by assuming that the sample represents the population. Since each observation represents other similar observations in the population that we didn't get to measure, if we ***sample with replacement*** to generate a new data set of size n from our data set (also of size n) it mimics the process of taking our population of interest. This process also ends up giving us useful sampling distributions of statistics even when our standard normality assumption is violated, similar to what we encountered in the permutation tests. Bootstrapping is especially useful in situations where we are interested in statistics other than the mean (say we want a confidence interval for a median or a standard deviation) or when we consider functions of more than one parameter and don't want to derive the distribution of the statistic (say the difference in two medians). In this text, bootstrapping is used to provide more trustworthy inferences when some of our assumptions (especially normality) might be violated for our parametric procedure.

To perform bootstrapping, we will use the `resample` function from the `mosaic` package. We can apply this function to a data set and get a new version of the data set by sampling new observations *with replacement* from the original one. The new, bootstrapped version of the data set (called `MockJury_BTS` below) contains a new variable called `orig. id` which is the number of the subject from the original data set. By summarizing how often each of these id's occurred in a bootstrapped data set, we can see how the re-sampling works. The `table` function will count up how many times each observation was used in the bootstrap sample, providing a row with the id followed by a row with the count[17]. In the first bootstrap sample shown, the 2nd, 7th, and 9th observations were sampled one time each, the 4th observation was sampled three times, and the 1st, 3rd, 5th, and many others were not sampled at all. Bootstrap sampling thus picks some observations multiple times and to do that it has to ignore some observations.

```
MockJury_BTS <- resample(MockJury2)
table(as.numeric(MockJury_BTS$orig.id))
```

```
##
##  1  2  3  4  5  6 10 11 12 14 15 17 18 19 20 22 24 26 29 30 32 35 36 37 39
##  1  2  2  1  3  2  1  2  1  1  3  1  2  1  2  1  2  1  2  2  2  2  1  2  2
## 40 42 43 44 45 46 47 48 49 55 58 59 60 61 69 70 71 72 74 75
##  2  1  1  4  2  2  1  2  1  2  1  1  2  2  2  2  2  1  1  1
```

Like in permutations, one randomization isn't enough. A second bootstrap sample is also provided to help you get a sense of what it is doing to generate a data set. It did not select subject 7 but did select 2, 4, 6, and 8 two times. You can see other variations in the resulting re-sampling of subjects with the most sampled subject being the chance of selecting any observation for any slot in the new data set is $1/75$ and the expected or mean number of appearances we expect to see for an observation is the number of tries times the probably of selection on each so $75 * 1/75 = 1$.

```
MockJury_BTS2 <- resample(MockJury2)
table(as.numeric(MockJury_BTS2$orig.id))
```

```
##
##  1  2  3  5  6  8 11 12 13 14 15 18 19 20 21 23 24 26 27 28 29 31 32 34 36
##  1  1  1  1  4  1  1  1  1  3  1  1  1  1  3  2  2  1  1  1  2  1  2  1  2
## 37 38 40 42 46 48 50 51 52 56 58 59 61 62 63 66 67 68 69 72 73 74 75
##  1  2  1  1  1  2  4  1  1  1  3  2  1  1  1  1  1  1  2  3  1  4  2
```

We can use the two results to get an idea of distribution of results in terms of number of times observations might be re-sampled when sampling with replacement and the variation in those results, as shown in Figure 2.10. We could also derive the expected counts for each number of times of re-sampling when we start with all observations having an equal chance and sampling with replacement but this isn't important for using bootstrapping methods.

[17]The `as.numeric` function is also used here. It really isn't important but makes sure the output of `table` is sorted by observation number by first converting the *orig.id* variable into a numeric vector.

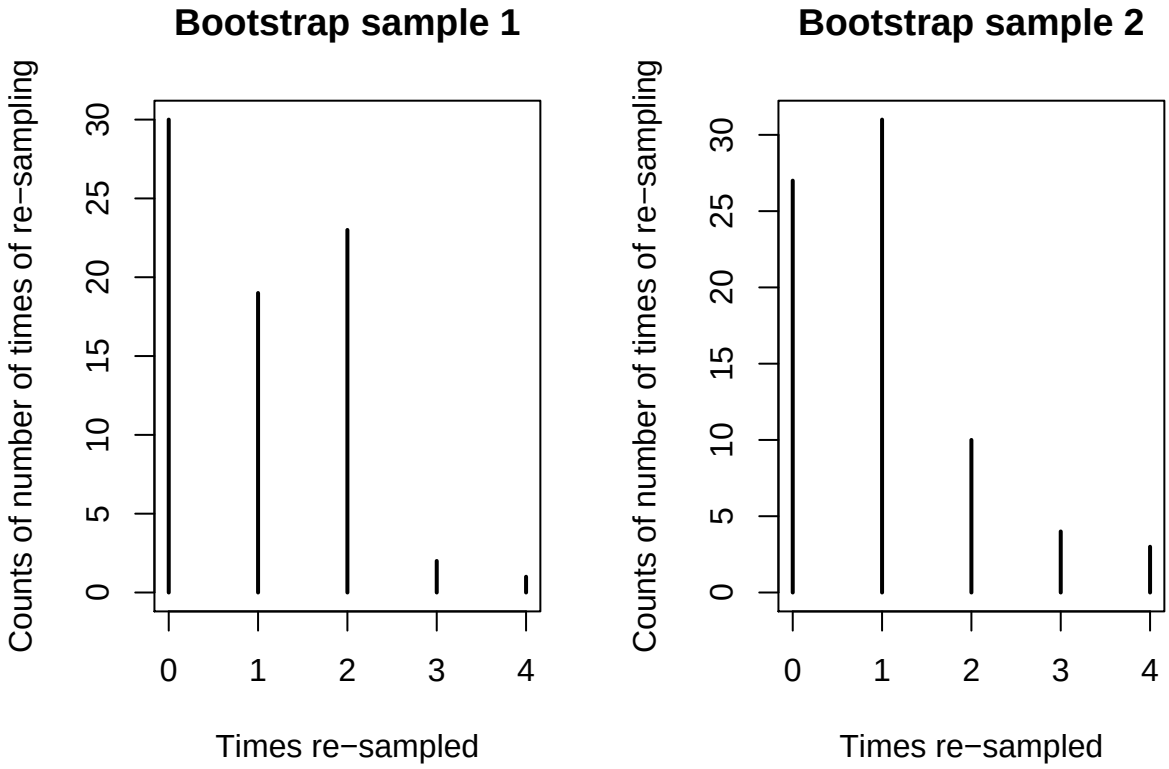


Figure 2.10: Counts of number of times of observation (or not observed for times re-sampled of 0) for two bootstrap samples.

The main point of this exploration was to see that each run of the `resample` function provides a new version of the data set. Repeating this B times using another `for` loop, we will track our quantity of interest, say T, in all these new "data sets" and call those results T^*. The distribution of the bootstrapped T^* statistics will tell us about the range of results to expect for the statistic and the middle ___% of the T^*'s provides a ***bootstrap confidence interval***[18] for the true parameter – here the *difference in the two population means*.

To make this concrete, we can revisit our previous examples, starting with the `MockJury2` data created before and our interest in comparing the mean sentences for the *Average*and *Unattractive* picture groups. The bootstrapping code is very similar to the permutation code except that we apply the `resample` function to the entire data set as opposed to the `shuffle` function being applied to the explanatory variable.

```
par(mfrow=c(1,2))
Tobs <- diffmean(Years ~ Attr, data=MockJury2); Tobs
B <- 1000
Tstar <- matrix(NA,nrow=B)
for (b in (1:B)){
  Tstar[b] <- diffmean(Years ~ Attr, data=resample(MockJury2))
  }
hist(Tstar, labels=T)
abline(v=Tobs, col="red", lwd=2)
plot(density(Tstar), main="Density curve of Tstar")
abline(v=Tobs, col="red", lwd=2)
```

```
favstats(Tstar)
```

```
## diffmean
## 1.837127
##         min       Q1   median       Q3      max     mean        sd    n
##  -0.3627312 1.305773 1.833091 2.385281 4.988756 1.854428 0.8438987 1000
##  missing
##        0
```

In this situation, the observed difference in the mean sentences is 1.84 years (Unattractive-Average), which is the vertical line in Figure 2.11. The bootstrap distribution shows the results for the difference in the sample means when fake data sets are re-constructed by sampling from the data set with replacement. The bootstrap distribution is approximately centered at the observed value (difference in the sample means) and is relatively symmetric.

The permutation distribution in the same situation (Figure **??**) had a similar shape but was centered at 0. Permutations create sampling distributions based on assuming the null hypothesis is true, which is useful for hypothesis testing. Bootstrapping creates distributions centered at the observed result, which is the sampling distribution "under the alternative" or when no null hypothesis is assumed; bootstrap distributions are useful for generating confidence intervals for the true parameter values.

To create a 95% bootstrap confidence interval for the difference in the true mean sentences ($\mu_{Unattr} - \mu_{Avg}$), select the middle 95% of results from the bootstrap distribution. Specifically, find the 2.5th percentile and the 97.5th percentile (values that put 2.5 and 97.5% of the results to the left) in the bootstrap distribution, which leaves 95% in the middle for the confidence interval. To find percentiles in a distribution in R, functions are of the form `q[Name of distribution]`, with the function `qt` extracting percentiles from a t-distribution (examples below). From the bootstrap results, use the `qdata` function on the `Tstar` results that contain the bootstrap distribution of the statistic of interest.

```
qdata(Tstar, 0.025)
```

```
##         p  quantile
## 0.0250000 0.2414232
```

[18] There are actually many ways to use this information to make a confidence interval. We are using the simplest method that is called the "percentile" method.

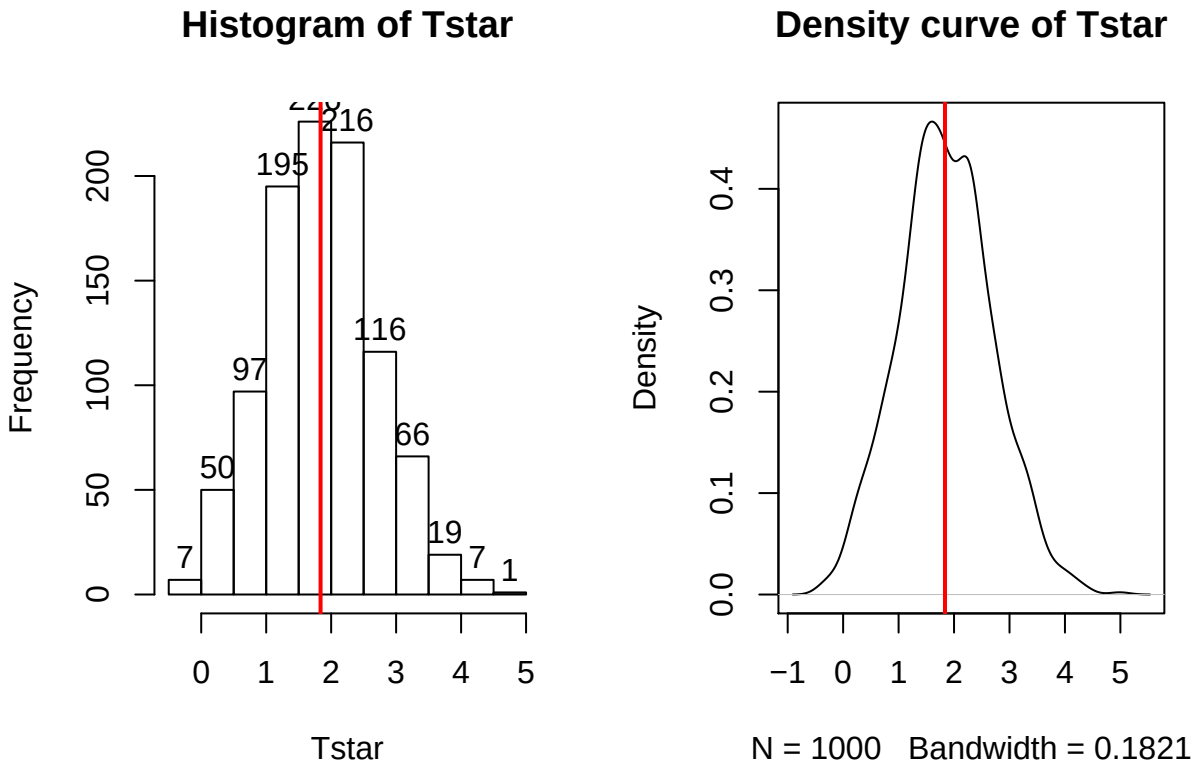


Figure 2.11: Histogram and density curve of bootstrap distributions of difference in sample mean `Years` with vertical line for the observed difference in the means of 1.84 years.

```
qdata(Tstar, 0.975)
```

```
##        p quantile
## 0.975000 3.521528
```

These results tell us that the 2.5th percentile of the bootstrap distribution is at 0.26 years and the 97.5th percentile is at 3.50 years. We can combine these results to provide a 95% confidence for $\mu_{Unattr} - \mu_{Avg}$ that is between 0.26 and 3.50. We can interpret this as with any confidence interval, that we are 95% confident that the difference in the true mean suggested sentences (Unattractive minus Average group) is between 0.26 and 3.50 years. We can also obtain both percentiles in one line of code using:

```
quantiles <- qdata(Tstar, c(0.025,0.975))
quantiles
```

```
##        quantile     p
## 2.5%  0.2414232 0.025
## 97.5% 3.5215278 0.975
```

Figure 2.12 displays those same percentiles on the bootstrap distribution residing in `Tstar`.

```
par(mfrow=c(1,2))
hist(Tstar, labels=T)
abline(v=quantiles$quantile, col="blue", lwd=3)
plot(density(Tstar), main="Density curve of Tstar")
abline(v=quantiles$quantile, col="blue", lwd=3)
```

Although confidence intervals can exist without referencing hypotheses, we can revisit our previous H_0 : $\mu_{Unattr} = \mu_{Avg}$. This null hypothesis is equivalent to testing $H_0 : \mu_{Unattr} - \mu_{Avg} = 0$, that the difference in the true means is equal to 0 years. And the difference in the means was the scale for our confidence interval, which did not contain 0 years. We will call 0 an interesting ***reference value*** for the confidence interval, because here it is the value where the true means are equal to each other (have a difference of 0 years). In general, if our confidence interval does not contain 0, then it is saying that 0 is not one of our likely values for the difference in the true means. This implies that we should reject a claim that they are equal. This provides the same inferences for the hypotheses that we considered previously using both a parametric and permutation approach. The general summary is that we can use confidence intervals to test hypotheses by assessing whether the reference value under the null hypothesis is in the confidence interval (FTR H_0) or outside the confidence interval (Reject H_0). P-values are more informative about hypotheses but confidence intervals are more information about the size of differences, so both offer useful information and, as shown here, can provide consistent conclusions about hypotheses.

As in the previous situation, we also want to consider the parametric approach for comparison purposes and to have that method available, especially to help us understand some methods where we will only consider parametric inferences in later chapters. The parametric confidence interval is called the ***equal variance, two-sample t confidence interval*** and assumes that the populations being sampled from are normally distributed and leads to using a t-distribution to form the interval. The output from the `t.test` function provides the parametric 95% confidence interval calculated for you:

```
t.test(Years ~ Attr, data=MockJury2, var.equal=T)
```

```
##
##  Two Sample t-test
##
## data:  Years by Attr
## t = -2.1702, df = 73, p-value = 0.03324
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.5242237 -0.1500295
```

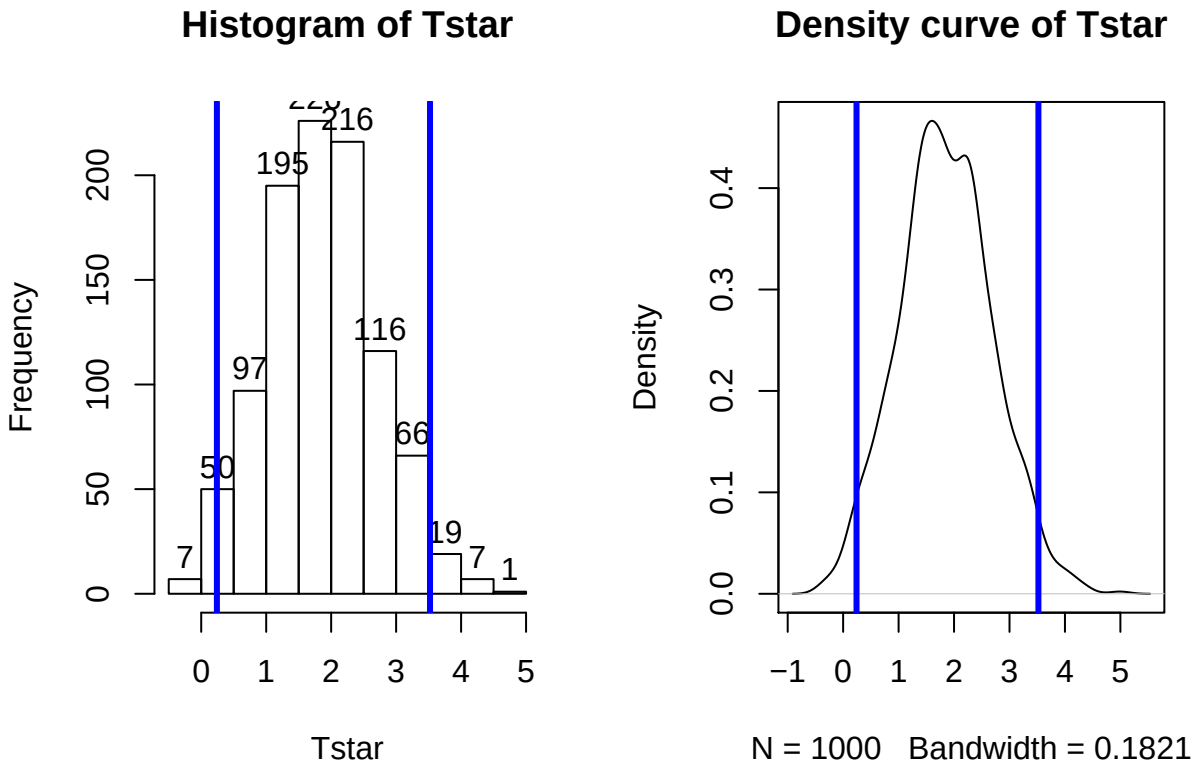


Figure 2.12: Histogram and density curve of bootstrap distribution with 95% bootstrap confidence intervals displayed (vertical lines).

```
## sample estimates:
##      mean in group Average mean in group Unattractive
##                   3.973684                   5.810811
```

The `t.test` function again switched the order of the groups and provides slightly different end-points than our bootstrap confidence interval (both are made at the 95% confidence level though), which was slightly narrower. Both intervals have the same interpretation, only the methods for calculating the intervals and the assumptions differ. Specifically, the bootstrap interval can tolerate different distribution shapes other than normal and still provide intervals that work well[19]. The other assumptions are all the same as for the hypothesis test, where we continue to assume that we have independent observations with equal variances for the two groups.

The formula that `t.test` is using to calculate the parametric ***equal-variance two-sample t confidence interval*** is:

$$\bar{x}_1 - \bar{x}_2 \mp t^*_{df} s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

In this situation, the df is again $n_1 + n_2 - 2$ and $s_p = \sqrt{\frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2}}$. The t^*_{df} is a multiplier that comes from finding the percentile from the t-distribution that puts C% in the middle of the distribution with C being the confidence level. It is important to note that this t^* has nothing to do with the previous test statistic t. It is confusing and many of you will, at some point, happily take the result from a test statistic calculation and use it for a multiplier in a t-based confidence interval. Figure 2.13 shows the t-distribution with 73 degrees of freedom and the cut-offs that put 95% of the area in the middle.

```
par(mfrow=c(1,1))
x<-seq(from=-4,to=4,length.out=200)
plot(x,dt(x,df=73),col="red",lty=2,lwd=3,type="l",xlab="t-values",ylab="Density",
     main="Plot of t(73) distribution" )
abline(v=-2.1702,lwd=3)
abline(v=2.1702,lwd=3)
```

For 95% confidence intervals, the multiplier is going to be close to 2 and anything else is a sign of a mistake. We can use R to get the multipliers for confidence intervals using the `qt` function in a similar fashion to how `qdata` was used in the bootstrap results, except that this new value must be used in the previous confidence interval formula. This function produces values for requested percentiles, so if we want to put 95% in the middle, we place 2.5% in each tail of the distribution and need to request the 97.5th percentile. Because the t-distribution is always symmetric around 0, we merely need to look up the value for the 97.5th percentile and know that the multiplier for the 2.5th percentile is just $-t^*$. The t^* multiplier to form the confidence interval is 1.993 for a 95% confidence interval when the $df = 73$ based on the results from `qt`:

```
qt(0.975, df=73)
```

```
## [1] 1.992997
```

Note that the 2.5th percentile is just the negative of this value due to symmetry and the real source of the minus in the minus/plus in the formula for the confidence interval.

```
qt(0.025, df=73)
```

```
## [1] -1.992997
```

We can also re-write the confidence interval formula into a slightly more general form as

[19] When hypothesis tests "work well" they have high power to detect differences while having Type I error rates that are close to what we choose a priori. When confidence intervals "work well", they contain the true parameter value in repeated random samples at around the selected confidence level

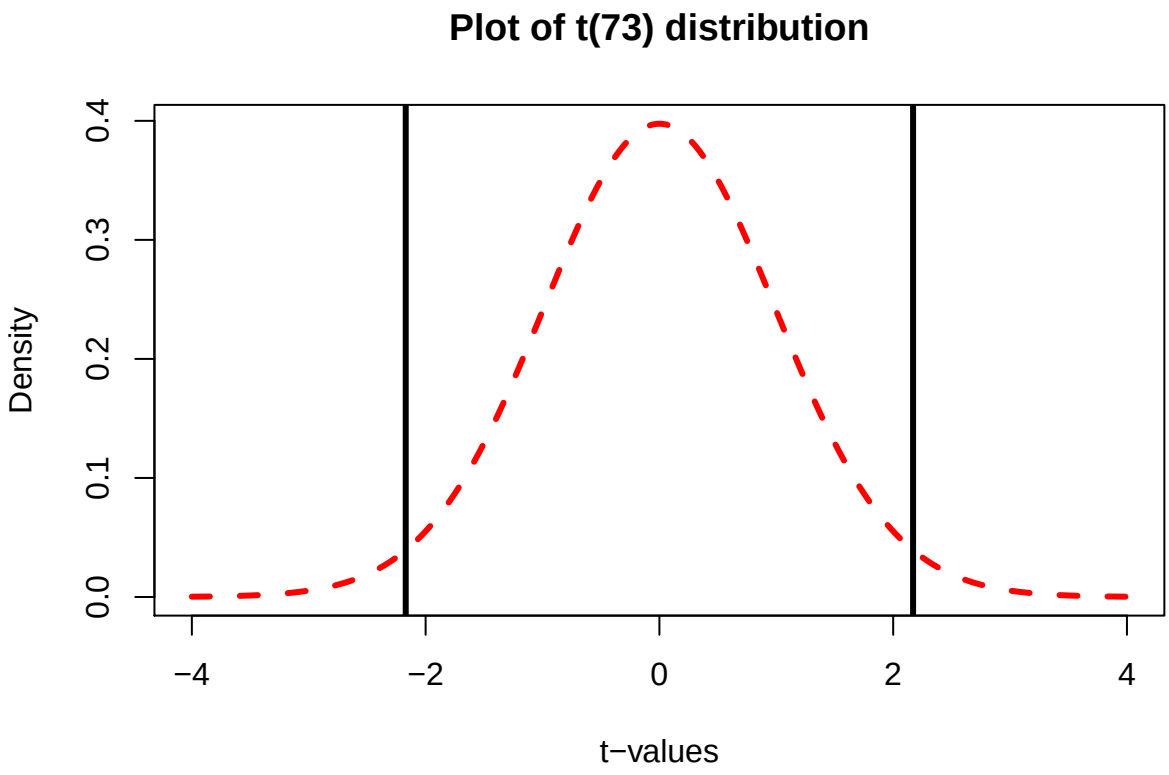


Figure 2.13: Plot of $t(73)$ with cut-offs for putting 95% of distributions in the middle.

$$\bar{x}_1 - \bar{x}_2 \mp t^*_{df} SE_{\bar{x}_1 - \bar{x}_2} \text{ OR } \bar{x}_1 - \bar{x}_2 \mp ME$$

where $SE_{\bar{x}_1-\bar{x}_2} = s_p\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}$ and $ME = t^*_{df} SE_{\bar{x}_1-\bar{x}_2}$. In some situations, researchers will report the ***standard error*** (SE) or ***margin of error*** (ME) as a method of quantifying the uncertainty in a statistic. The SE is an estimate of the standard deviation of the statistic (here $\bar{x}_1 - \bar{x}_2$) and the ME is an estimate of the precision of a statistic that can be used to directly form a confidence interval. The ME depends on the choice of confidence level although 95% is almost always selected.

To finish this example, R can be used to help you do calculations much like a calculator except with much more power "under the hood". You have to make sure you are careful with using `( )` to group items and remember that the asterisk (*) is used for multiplication in R. We need the pertinent information which is available from the `favstats` output. Two versions of this output are provided below, the first is how the output appears directly from R, the second is a formatted table has the relevant information bolded which is needed to calculate the confidence interval "by hand" using R.

```
favstats(Years~Attr, data=MockJury2)
```

```
##           Attr min Q1 median Q3 max     mean       sd  n missing
## 1      Average   1  2      3  5  12 3.973684 2.823519 38       0
## 2 Unattractive   1  2      5 10  15 5.810811 4.364235 37       0
```

Attr	min	Q1	median	Q3	max	mean	sd	n	missing
Average	1	2	3	5	12	**3.97**	**2.82**	**38**	0
Unattractive	1	2	5	10	15	**5.81**	**4.36**	**37**	0

Start with typing the following command to calculate s_p and store it in a variable named `sp`:

```
sp <- sqrt(((38-1)*(2.8235^2)+(37-1)*(4.364^2))/(38+37-2))
sp
```

```
## [1] 3.665036
```

Then calculate the confidence interval that `t.test` provided using:

```
3.974-5.811+c(-1,1)*qt(.975,df=73)*sp*sqrt(1/38+1/37)
```

```
## [1] -3.5240302 -0.1499698
```

The previous code uses `c(-1, 1)` times the margin of error to subtract and add the ME to the difference in the sample means ($3.974 - 5.811$), which generates the lower and then upper bounds of the confidence interval. If desired, we can also use just the last portion of the previous calculation to find the margin of error, which is 1.69 here.

```
qt(.975,df=73)*sp*sqrt(1/38+1/37)
```

```
## [1] 1.68703
```

2.9 Bootstrap confidence intervals for difference in GPAs

We can now apply the new confidence interval methods on the STAT 217 grade data. This time we start with the parametric 95% confidence interval "by hand" in R and then use `t.test` to verify our result. The `favstats` output provides us with the required information to calculate the confidence interval:

```
favstats(GPA~Sex,data=s217)
```

```
##   Sex  min  Q1 median   Q3 max     mean        sd  n missing
## 1   F 2.50 3.1  3.400 3.70   4 3.338378 0.4074549 37       0
## 2   M 1.96 2.8  3.175 3.46   4 3.088571 0.4151789 42       0
```

The df are $37 + 42 - 2 = 77$. Using the SDs from the two groups and their sample sizes, we can calculate s_p:

```
sp <- sqrt(((37-1)*(0.4075^2)+(42-1)*(0.41518^2))/(37+42-2))
sp
```

```
## [1] 0.4116072
```

The margin of error is:

```
qt(.975,df=77)*sp*sqrt(1/37+1/42)
```

```
## [1] 0.1847982
```

All together, the 95% confidence interval is:

```
3.338-3.0886+c(-1,1)*qt(.975,df=77)*sp*sqrt(1/37+1/42)
```

```
## [1] 0.0646018 0.4341982
```

So we are 95% confident that the difference in the true mean GPAs between females and males (females minus males) is between 0. 065 and 0. 434 GPA points. We get a similar[20] result from the `t.test` output:

```
t.test(GPA~Sex,data=s217,var.equal=T)
```

```
##
##  Two Sample t-test
##
## data:  GPA by Sex
## t = 2.6919, df = 77, p-value = 0.008713
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.06501838 0.43459552
## sample estimates:
## mean in group F mean in group M
##        3.338378        3.088571
```

Note that we can easily switch to 90% or 99% confidence intervals by simply changing the percentile in `qt` or changing `conf. level` in the `t.test` function. In the following two lines of code, we added octothorpes[21](#) and then some text after function calls to explain what is being calculated. In computer code, octothorpes provide a way of adding comments that tell the software (here R) to ignore any text after a "#" on a given line. In the color version of the text, comments are also clearly distinguished.

```
qt(.95,df=77) # For 90% confidence and 77 df
```

```
## [1] 1.664885
```

```
qt(.995,df=77) #For 99% confidence and 77 df
```

```
## [1] 2.641198
```

```
t.test(GPA~Sex,data=s217,var.equal=T,conf.level=0.90)
```

[20] We rounded the means a little and that caused the small difference in results.

[21] You can correctly call octothorpes *number* symbols or, in the twitter verse, *hashtags*. For more on this symbol, see "http://blog.dictionary.com/octothorpe/". I usually call them number symbols too.

```
##
##  Two Sample t-test
##
## data:  GPA by Sex
## t = 2.6919, df = 77, p-value = 0.008713
## alternative hypothesis: true difference in means is not equal to 0
## 90 percent confidence interval:
##  0.09530553 0.40430837
## sample estimates:
## mean in group F mean in group M
##        3.338378        3.088571
```

```
t.test(GPA~Sex,data=s217,var.equal=T,conf.level=0.99)
```

```
##
##  Two Sample t-test
##
## data:  GPA by Sex
## t = 2.6919, df = 77, p-value = 0.008713
## alternative hypothesis: true difference in means is not equal to 0
## 99 percent confidence interval:
##  0.004703598 0.494910301
## sample estimates:
## mean in group F mean in group M
##        3.338378        3.088571
```

As a review of some basic ideas with confidence intervals make sure you can answer the following questions:

1. What is the impact of increasing the confidence level in this situation?
2. What happens to the width of the confidence interval if the size of the SE increases or decreases?
3. What about increasing the sample size – should that increase or decrease the width of the interval?

All the general results you learned before about impacts to widths of CIs hold in this situation whether we are considering the parametric or bootstrap methods...

To finish this example, we will generate the comparable bootstrap 90% confidence interval using the bootstrap distribution in Figure 2.14.

```
Tobs <- diffmean(GPA ~ Sex, data=s217); Tobs
```

```
##   diffmean
## -0.2498069
```

```
par(mfrow=c(1,2))
B<- 1000
Tstar<-matrix(NA,nrow=B)
for (b in (1:B)){
  Tstar[b]<-diffmean(GPA ~ Sex, data=resample(s217))
  }
qdata(Tstar,.05)
```

```
##          p   quantile
##  0.0500000 -0.4032273
```

```
qdata(Tstar,.95)
```

```
##           p    quantile
##  0.95000000 -0.09521925
```

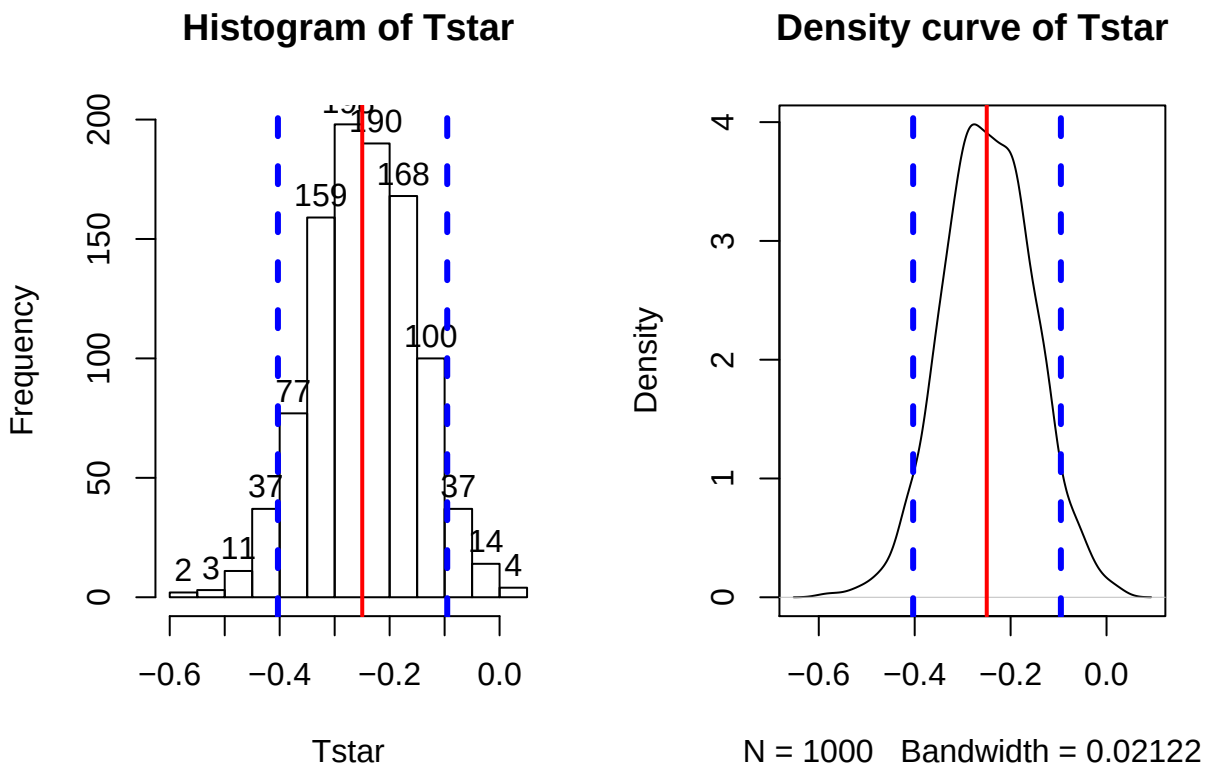


Figure 2.14: Histogram and density curve of bootstrap distribution of difference in sample mean GPAs (male minus female) with observed difference (solid vertical line) and quantiles that delineate the 90% confidence intervals (dashed vertical lines).

```
quantiles<-qdata(Tstar,c(.05,.95))
quantiles
```

```
##         quantile    p
## 5%  -0.40322729 0.05
## 95% -0.09521925 0.95
```

The output tells us that the 90% confidence interval is from -0.393 to -0.094 GPA points. The bootstrap distribution with the observed difference in the sample means and these cut-offs is displayed in Figure 2.14 using this code:

```
par(mfrow=c(1,2))
hist(Tstar,labels=T)
abline(v=Tobs,col="red",lwd=2)
abline(v=quantiles$quantile,col="blue",lwd=3,lty=2)
plot(density(Tstar),main="Density curve of Tstar")
abline(v=Tobs,col="red",lwd=2)
abline(v=quantiles$quantile,col="blue",lwd=3,lty=2)
```

In the previous output, the parametric 90% confidence interval is from 0.095 to 0.404, suggesting similar results again from the two approaches once you account for the two different orders of differencing of the groups. Based on the bootstrap CI, we can say that we are 90% confident that the difference in the true

mean GPAs for STAT 217 students is between -0.393 to -0.094 GPA points (male minus females). Because sex cannot be assigned to the subjects, we cannot infer that sex is causing this difference and because this was a voluntary response sample of STAT 217 students in a given semester, we cannot infer that a difference of this size would apply to all STAT 217 students or even students in another semester.

Throughout the semester, pay attention to the distinctions between parameters and statistics, focusing on the differences between estimates based on the sample and inferences for the population of interest in the form of the parameters of interest. Remember that statistics are summaries of the sample information and parameters are characteristics of populations (which we rarely know). And that our inferences are limited to the population that we randomly sampled from, if we randomly sampled.

2.10 Chapter summary

In this chapter, we reviewed basic statistical inference methods in the context of a two-sample mean problem. You were introduced to using R to do permutation testing and generate bootstrap confidence intervals as well as obtaining parametric *t*-test and confidence intervals in this same situation. You should have learned how to use a `for` loop for doing the nonparametric inferences and the `t.test` function for generating parametric inferences. In the two examples considered, the parametric and nonparametric methods provided similar results, suggesting that the assumptions were at least close to being met for the parametric procedures. When parametric and nonparametric approaches disagree, the nonparametric methods are likely to be more trustworthy since they have less restrictive assumptions but can still have problems.

When the noted conditions are not met in a hypothesis testing situation, the Type I error rates can be inflated, meaning that we reject the null hypothesis more often than we have allowed to occur by chance. Specifically, we could have a situation where our assumed 5% significance level test might actually reject the null when it is true 20% of the time. If this is occurring, we call a procedure ***liberal*** (it rejects too easily) and if the procedure is liberal, how could we trust a small p-value to be a "real" result and not just an artifact of violating the assumptions of the procedure? Likewise, for confidence intervals we hope that our 95% confidence level procedure, when repeated, will contain the true parameter 95% of the time. If our assumptions are violated, we might actually have an 80% confidence level procedure and it makes it hard to trust the reported results for our observed data set. Statistical inference relies on a belief in the methods underlying our inferences. If we don't trust our assumptions, we shouldn't trust the conclusions to perform the way we want them to. As sample sizes increase and/or violations of conditions lessen, then the procedures will perform better. In Chapter 3, we'll learn some new tools for doing diagnostics to help us assess how much those conditions are violated.

2.11 Summary of important R code

The main components of R code used in this chapter follow with components to modify in red, remembering that any R packages mentioned need to be installed and loaded for this code to have a chance of working:

- summary(DATASETNAME)
  - Provides numerical summaries of all variables in the data set.
- t.test(Y ~ X, data=DATASETNAME, conf.level=0.95)
  - Provides two-sample t-test test statistic, df, p-value, and 95% confidence interval.
- 2*pt(abs(Tobs), df=DF, lower.tail=F)
  - Finds the two-sided test p-value for an observed 2-sample t-test statistic of `Tobs`.
- hist(DATASETNAME$Y)
  - Makes a histogram of a variable named `Y` from the data set of interest.

- boxplot(Y~X, data=DATASETNAME)
  - Makes a boxplot of a variable named Y for groups in X from the data set.
- beanplot(Y~X, data=DATASETNAME)
  - Makes a beanplot of a variable named Y for groups in X from the data set.
  - Requires the `beanplot` package is loaded.
- mean(Y~X, data=DATASETNAME); sd(Y~X, data=DATASETNAME)
  - Provides the mean and sd of responses of Y for each group described in X.
  - This usage of `mean` and `sd` requires the `mosaic` package.
- favstats(Y~X, data=DATASETNAME)
  - Provides numerical summaries of Y by groups described in X.
- Tobs <- t.test(Y~X, data=DATASETNAME, var.equal=T)$statistic; Tobs
  B <- 1000
  Tstar <- matrix(NA, nrow=B)
  for (b in (1:B)){
  Tstar[b] <- t.test(Y~shuffle(X), data=DATASETNAME, var.equal=T)$statistic
  }
  - Code to run a `for` loop to generate 1000 permuted versions of the test statistic using the `shuffle` function and keep track of the results in `Tstar`
- pdata(Tstar, abs(Tobs, lower.tail=F)
  - Finds the proportion of the permuted test statistics in Tstar that are less than -|Tobs| or greater than |Tobs|, useful for finding the two-sided test p-value.
- Tobs <- diffmean(Y~X, data=DATASETNAME, var.equal=T)$statistic; Tobs
  B <- 1000
  Tstar <- matrix(NA, nrow=B)
  for (b in (1:B)){
  Tstar[b] <- diffmean(Y~X, data=resample(DATASETNAME))
  }
  - Code to run a `for` loop to generate 1000 bootstrapped versions of the data set using the `resample` function and keep track of the results of the statistic in `Tstar`.
- qdata(Tstar, c(0. 025, 0. 975))
  - Provides the values that delineate the middle 95% of the results in the bootstrap distribution (`Tstar`)

2.12 Practice problems

Load the `HELPrct` data set from the `mosaicData` package (you need to install the `mosaicData` package once to be able to load it). The HELP study was a clinical trial for adult inpatients recruited from a detoxification unit. Patients with no primary care physician were randomly assigned to receive a multidisciplinary assessment and a brief motivational intervention or usual care and various outcomes were observed. Two of the variables in the data set are `sex`, a factor with levels male and female and `daysanysub` which is the time (in days) to first use of any substance post-detox. We are interested in the difference in mean number of days to first use of any substance post-detox between males and females. There are some missing responses and the following code will produce `favstats` with the missing values and then provide a data set that for complete observations by applying the `na.omit` function that removes any observations with missing values.

```
require(mosaicData)
data(HELPrct)
HELPrct <- HELPrct[, c("daysanysub", "sex")] #Just focus on two variables
HELPrct <- na.omit(HELPrct2) #Removes subjects with missing
favstats(daysanysub~sex, data=HELPrct2)
favstats(daysanysub~sex, data=HELPrct3)
```

2.1. Based on the results provided, how many observations were missing for males and females? Missing values here likely mean that the subjects didn't use any substances post-detox in the time of the study but might have at a later date – the study just didn't run long enough. This is called ***censoring***. What is the problem with the numerical summaries here if the missing responses were all something larger than the largest observation?

2.2. Make a beanplot and a boxplot of `daysanysub` ~ `sex` using the `HELPrct3` data set created above. Compare the distributions, recommending parametric or nonparametric inferences.

2.3. Generate the permutation results and write out the 6+ steps of the hypothesis test, making sure to note the numerical value of observed test statistic you are using and include a discussion of the scope of inference.

2.4. Interpret the p-value for these results.

2.5. Generate the parametric `t.test` results, reporting the test-statistic, its distribution under the null hypothesis, and compare the p-value to those observed using the permutation approach.

2.6. Make and interpret a 95% bootstrap confidence interval for the difference in the means.

Chapter 3

One-Way ANOVA

In Chapter 2, tools for comparing the means of two groups were considered. More generally, these methods are used for a quantitative response and a categorical explanatory variable (group) which had two and only two levels. The full prisoner rating data set actually contained three groups (Figure 3.1 with *Beautiful*, *Average*, and *Unattractive* rated pictures randomly assigned to the subjects for sentence ratings. In a situation with more than two groups, we have two choices. First, we could rely on our two group comparisons, performing tests for every possible pair (*Beautiful* vs*Average*, *Beautiful* vs *Unattractive*, and *Average* vs *Unattractive*). We spent Chapter 2 doing inferences for differences between *Average* and *Unattractive*. The other two comparisons would lead us to initially end up with three p-values and no direct answer about our initial question of interest – is there some overall difference in the average sentences provided across the groups? In this chapter, we will learn a new method, called ***Analysis of Variance***, or ***One-Way ANOVA*** since there is just one[1] grouping variable. After we perform our One-Way ANOVA test for overall evidence of a difference, we will revisit the comparisons similar to those considered in Chapter 2 to get more details on specific differences among *all* the pairs of groups – what we call ***pair-wise comparisons***. An issue is created when you perform many tests simultaneously and we will augment our previous methods with an adjusted method for pairwise comparisons to make our results valid called ***Tukey's Honest Significant Difference***.

To make this more concrete, we return to the original MockJury data, making side-by-side boxplots and beanplots (Figure 3.1 as well summarizing the sentences for the three groups using `favstats`.

```
require(heplots)
require(mosaic)
data(MockJury)
par(mfrow=c(1,2))
boxplot(Years~Attr,data=MockJury)
require(beanplot)
beanplot(Years~Attr,data=MockJury,log="",col="bisque",method="jitter")
```

```
favstats(Years~Attr,data=MockJury)
```

```
##           Attr min Q1 median   Q3 max     mean       sd  n missing
## 1    Beautiful   1  2      3  6.5  15 4.333333 3.405362 39       0
## 2      Average   1  2      3  5.0  12 3.973684 2.823519 38       0
## 3 Unattractive   1  2      5 10.0  15 5.810811 4.364235 37       0
```

There are slight differences in the sample sizes in the three groups with 37 *Unattractive*, 38 *Average* and 39 *Beautiful* group responses, providing a data set has a total sample size of $N = 114$. The *Beautiful* and

[1]In Chapter 4, methods are discussed for when there are two categorical explanatory variables that is called the Two-Way ANOVA and related ANOVA tests are used in Chapter 8 for working with extensions of these models.

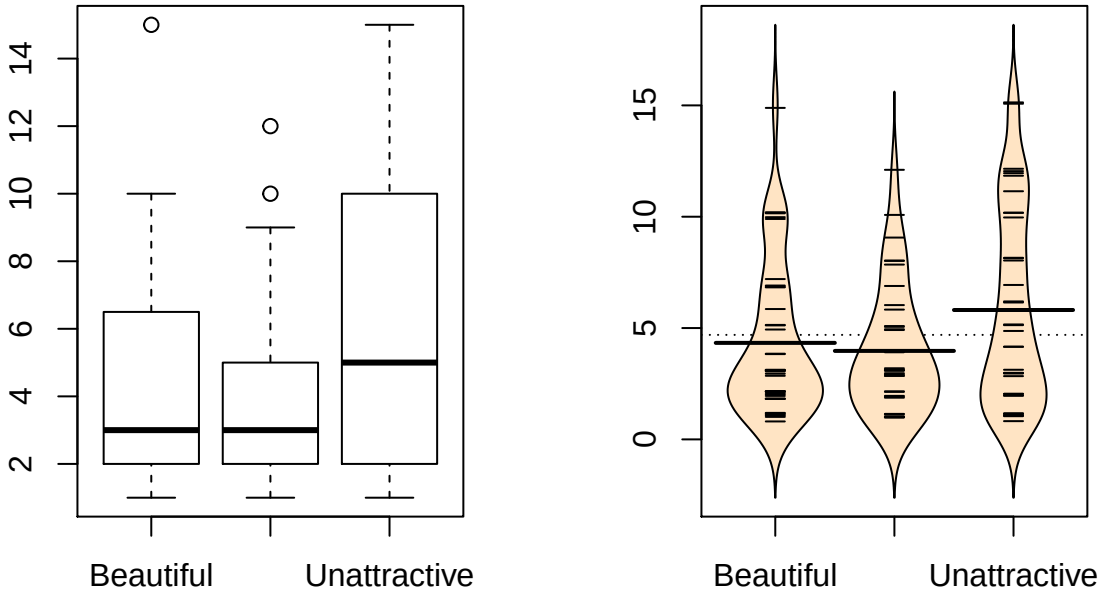


Figure 3.1: Boxplot and beanplot of the sentences (years) for the three treatment groups.

Average groups do not appear to be very different with means of 4.33 and 3.97 years. In Chapter 2, we found moderate evidence regarding the difference in *Average*and *Unattractive*. It is less clear whether we might find evidence of a difference between *Beautiful* and *Unattractive* groups since we are comparing means of 5.81 and 4.33 years. All the distributions appear to be right skewed with relatively similar shapes. The variability in *Average* and *Unattractive* groups seems like it could be slightly different leading to an overall concern of whether the variability is the same in all the groups.

3.1 Situation

We introduced the statistical model $y_{ij} = \mu_j + \epsilon_j$ in Chapter 2 for the situation with $j = 1$ or 2 to denote a situation where there were two groups and, for the model that is consistent with the alternative hypothesis, the means differed. Now we have three groups and the previous model can be extended to this new situation by allowing j to be 1, 2, or 3. Now that we have more than two groups, we need to admit that what we were doing in Chapter 2 was actually fitting what is called a ***linear model***. The linear model assumes that the responses follow a normal distribution with the linear model defining the mean, all observations have the same variance, and the parameters for the mean in the model enter linearly. This last condition is hard to explain at this level of material – it is sufficient to know that there models where the parameters enter the model nonlinearly and that they are beyond the scope of this course. The result of this constraint is that we will be able to use the same general modeling framework for the methods introduced in Chapters 3, 4, 6, 7, and 8.

As in Chapter 2, we have a null hypothesis that defines a situation (and model) where all the groups have the same mean. Specifically, the ***null hypothesis*** in the general situation with J groups ($J \geq 2$) is to have all the **true** group means equal,

$$H_0 : \mu_1 = \ldots \mu_J.$$

This defines a model where all the groups have the same mean so it can be defined in terms of a single mean, μ, for the i^{th} observation from the j^{th} group as $y_{ij} = \mu + \epsilon_{ij}$. This is not the model that most researchers want to be the final description of their study as it implies no difference in the groups. There is more caution required to specify the alternative hypothesis with more than two groups. The ***alternative hypothesis*** needs to be the logical negation of this null hypothesis of all groups having equal means; to make the null hypothesis false, we only need one group to differ but more than one group could differ from the others. Essentially, there are many ways to "violate" the null hypothesis so we choose some delicate wording for the alternative hypothesis when there are more than 2 groups. Specifically, we state the alternative as

$$H_A : \text{ Not all } \mu_j \text{ are equal}$$

or, in words, **at least one of the true means differs among the J groups**. You will be attracted to trying to say that all means are different in the alternative but we do not put this strict a requirement in place to reject the null hypothesis. The alternative model allows all the true group means to differ but does require that they differ with

$$\mu_j + \epsilon_{ij}.$$

This linear model states that the response for the i^{th} observation in the j^{th} group, $\mathbf{y_{ij}}$, is modeled with a group j ($j = 1, \ldots, J$) population mean, μ_j, and a random error for each subject in each group ϵ_{ij}, that we assume follows a normal distribution and that all the random errors have the same variance, σ^2. We can write the assumption about the random errors, often called the ***normality assumption***, as $\epsilon_{ij} \sim N(0, \sigma^2)$. There is a second way to write out this model that allows extension to more complex models discussed below, so we need a name for this version of the model. The model written in terms of the μ_j's is called the **cell means model** and is the easier version of this model to understand.

One of the reasons we learned about beanplots is that it helps us visually consider all the aspects of this model. In the right panel of Figure 3.1, we can see the wider, bold horizontal lines that provide the estimated group means. The bigger the differences in the sample means, the more likely we are to find evidence against the null hypothesis. You can also see the null model on the plot that assumes all the groups have the same as displayed in the dashed horizontal line at 4.7 years (the R code below shows the overall mean of *Years* is 4.7). While the hypotheses focus on the means, the model also contains assumptions about the distribution of the responses – specifically that the distributions are normal and that all the groups have the same variability. As discussed previously, it appears that the distributions are right skewed and the variability might not be the same for all the groups. The boxplot provides the information about the skew and variability but since it doesn't display the means it is not directly related to the linear model and hypotheses we are considering.

```
mean(MockJury$Years)
```

```
## [1] 4.692982
```

There is a second way to write out the One-Way ANOVA model that provides a framework for extensions to more complex models described in Chapter 4 and beyond. The other ***parameterization*** (way of writing out or defining) of the model is called the **reference-coded model** since it writes out the model in terms of a ***baseline group*** and deviations from that baseline or reference level. The reference-coded model for the i^{th} subject in the j^{th} group is $y_{ij} = \boldsymbol{\alpha} + \boldsymbol{\tau_j} + \epsilon_{ij}$ where $\boldsymbol{\alpha}$ (alpha) is the true mean for the baseline group (first alphabetically) and the $\boldsymbol{\tau_j}$ (tau j) are the deviations from the baseline group for group j. The deviation for the baseline group, $\boldsymbol{\tau_1}$, is always set to 0 so there are really just deviations for groups 2 through J. The equivalence between the two models can be seen by considering the mean for the first, second, and J^{th} groups in both models:

	Cell means:	Reference-coded:
Group 1:	${\color{red}{\mu_1}}$	${\color{purple}{\boldsymbol{\alpha}}}$
Group 2:	${\color{red}{\mu_2}}$	${\color{red}{\boldsymbol{\tau_2}}}$
$\ldots$	$\ldots$	$\ldots$
Group J:	${\color{red}{\mu_J}}$	${\color{purple}{\boldsymbol{\tau_J}}}$

The hypotheses for the reference-coded model are similar to those in the cell-means coding except that they are defined in terms of the deviations, $\boldsymbol{\tau_j}$. The null hypothesis is that there is no deviation from the baseline for any group – that all the $\boldsymbol{\tau_j}$'s $= 0$,

$$\boldsymbol{H_0 : \tau_2 = \ldots = \tau_J = 0}.$$

The alternative hypothesis is that at least one of the deviations is not 0,

$$\boldsymbol{H_A}: \textbf{ Not all } \boldsymbol{\tau_j} \textbf{ equal } \textbf{0}.$$

In this chapter, you are welcome to use either version (unless we instruct you otherwise) but we have to use the reference-coding in subsequent chapters. The next task is to learn how to use R's linear model `lm` function to get estimates of the parameters in each model, but first a quick review of these new ideas:

Cell Means Version

- $H_0 : \mu_1 = \ldots \mu_J$ $\qquad$ $H_A :$ Not all μ_j equal
- Null hypothesis in words: No difference in the true means between the groups.
- Null model $y_{ij} = \mu_j + \epsilon_{ij}$
- Alternative hypothesis in words: At least one of the true means differs between the groups.
- Alternative model: $y_{ij} = \mu_j + \epsilon_{ij}$.

Reference-coded Version

- $H_0 : \boldsymbol{\tau_2 \ldots \tau_J = 0}$ $H_A :$ Not all τ_j equal
- Null hypothesis in words: No deviation of the true mean for any groups from the baseline group.
- Null model: $y_{ij} = \boldsymbol{\alpha} + \tau_j + \epsilon_{ij}$
- Alternative hypothesis in words: At least one of the true deviations is different from 0 or that at least one group has a different true mean than the baseline group.
- Alternative model: $y_{ij} = \boldsymbol{\alpha + \tau_j} + \epsilon_{ij}$

In order to estimate the models discussed above, the `lm` function is used. If you look closely in the code for the rest of the book, any model for a quantitative response will use this function, suggesting a common thread in the most commonly used statistical models. The `lm` function continues to use the same format as previous functions, `lm(Y~X, data=datasetname)`. It ends up that this code will give you the reference-coded version of the model by default (R thinks it is that important!). We want to start with the cell-means version of the model, so we have to override the standard technique and add a "`-1`" to the formula interface to tell R that we want to the cell-means coding. Generally, this looks like `lm(Y~X-1 , data=datasetname)`. Once we fit a model in R, the `summary` function run on the model provides a useful "summary" of the model coefficients and a suite of other potentially interesting information. When fitting this version of the One-Way ANOVA model, you will find a row of output for each group relating the μ_j's. The output contains columns for an estimate (`Estimate`), standard error (`Std.Error`), t-value (`t value`), and p-value (`Pr(>|t|)`). We'll learn to use all the output in the following material, but for now just focus on the estimates of the parameters that the function provides that we put in bold.

```
lm1 <- lm(Years ~ Attr-1, data=MockJury)
summary(lm1)
```

```
##
## Call:
## lm(formula = Years ~ Attr - 1, data = MockJury)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8108 -2.8108 -0.9737  2.1892 10.6667
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## AttrBeautiful      4.3333     0.5730   7.563 1.23e-11 ***
## AttrAverage        3.9737     0.5805   6.845 4.41e-10 ***
## AttrUnattractive   5.8108     0.5883   9.878  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.578 on 111 degrees of freedom
## Multiple R-squared:  0.6449, Adjusted R-squared:  0.6353
## F-statistic: 67.21 on 3 and 111 DF,  p-value: < 2.2e-16
```

	Estimate	Std. Error	t value	Pr(>|t|)
AttrBeautiful	**4.33**	0.573	7.56	1.23e-11
AttrAverage	**3.97**	0.58	6.85	4.41e-10
AttrUnattractive	**5.81**	0.588	9.88	6.86e-17

In general, we denote estimated parameters with a hat over the parameter of interest to show that it is an estimate. For the true mean of group j, μ_j, we estimate it with $\hat{\mu}_j$, which is just the sample mean for group j, $\bar{x}_j$. The model suggests an estimate for each observation that we denote as $\hat{y}_{ij}$ that we will also call a

fitted value based on the model being considered. The three estimates are bolded in the previous output, with the same estimate used for all observations in the same group. R tries to help you to sort out which row of output corresponds to which group by appending the group name l with the variable name. Here, the variable name was `Attr` and the first group alphabetically was *Beautiful*, so R provides a row labeled `AttrBeautiful` with an estimate of 4.3333. The sample means from the three groups can be seen to directly match that and the other two results.

```
mean(Years ~ Attr, data=MockJury)
```

```
##    Beautiful      Average Unattractive
##     4.333333     3.973684     5.810811
```

The reference-coded version of the same model is more complicated but ends up giving the same results once we understand what it is doing. It uses a different parameterization to accomplish this so has different model output. Here is the model summary:

```
lm2 <- lm(Years ~ Attr, data=MockJury)
summary(lm2)
```

```
##
## Call:
## lm(formula = Years ~ Attr, data = MockJury)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8108 -2.8108 -0.9737  2.1892 10.6667
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        4.3333     0.5730   7.563 1.23e-11 ***
## AttrAverage       -0.3596     0.8157  -0.441   0.6601
## AttrUnattractive   1.4775     0.8212   1.799   0.0747 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.578 on 111 degrees of freedom
## Multiple R-squared:  0.04754,    Adjusted R-squared:  0.03038
## F-statistic:  2.77 on 2 and 111 DF,  p-value: 0.067
```

The estimated model coefficients are $\hat{\alpha} = 4.333$ years, $\hat{tau}_2 = -0.3596$ years, $\hat{\tau}_3 = 1.4775$ years where R selected group 1 for *Beautiful*, 2 for *Average*, and 3 for *Unattractive*. The way you can figure out the baseline group (group 1 is *Beautiful* here) is to see which category label is *not present* in the output. **The baseline level is typically the first group label alphabetically**, but you should always check this. Based on these definitions, there are interpretations available for each coefficient. For $\hat{\alpha} = 4.333$ years, this is an estimate of the mean sentencing time for the *Beautiful*group. $\hat{\tau}_2 = -0.3596$ years is the deviation of the *Average* group's mean from the *Beautiful* groups mean (specifically, it is 0.36 years lower). Finally, $\hat{\tau}_3 = 1.4775$ years tells us that the *Unattractive* group mean sentencing time is 1.48 years higher than time. These interpretations lead directly to reconstructing the estimated means for each group by combining the baseline and pertinent deviations as shown in Table 3.2.

We can also visualize the results of our linear models using what are called ***term-plots*** or ***effect-plots*** (from the `effects` package; Fox, 2003) as displayed in Figure 3.2. We don't want to use the word "effect" for these model components unless we have random assignment in the study design so we generically call these ***term-plots*** as they display terms or components from the model in hopefully useful ways to aid in model interpretation even in the presence of complicated model parameterizations. Specifically, these plots take an estimated model and show you its estimates along with 95% confidence intervals generated by the linear model. To make this plot, you need to install and load the `effects` package and then use `plot(allEffects(...))`

Table 3.2: Constructing group mean estimates from the reference-coded linear model estimates.

Group	Formula	Estimates
Beautiful	$\hat{\alpha}$	**4.3333** years
Average	$\hat{\alpha}+\hat{\tau}_2$	4.3333 - 0.3596 = **3.974** years
Unattractive	$\hat{\alpha}+\hat{\tau}_3$	4.3333 + 1.4775 = **5.811** years

functions together on the `lm` object called `lm2` that was estimated above. You can find the correspondence between the displayed means and the estimates that were constructed in Table 3.2.

```
require(effects)
plot(allEffects(lm2))
```

In order to assess evidence for having different means for the groups, we will compare either of the previous models (cell-means or reference-coded) to a null model based on the null hypothesis ($H_0 : \mu_1 = \ldots = \mu_J$) which implies a model of $y_{ij} = \mu_j + \epsilon_{ij}$ in the cell-means version where μ is a common mean for all the observations. We will call this the **mean-only** model since it only has a single mean in it. In the reference-coding version of the model, we have a null hypothesis that $H_0 : \tau_2 = \ldots = \tau_J = 0$, so the "mean-only" model is $y_{ij} = \boldsymbol{\alpha} + \epsilon_{ij}$ with $\boldsymbol{\alpha}$ having the same definition as μ for the cell means model – it forces a common value for the mean for all the groups. Moving from the *reference-coded* model to the *mean-only* model is also an example of a situation where we move from a "full" model to a "reduced" model by setting some coefficients in the "full" model to 0 and, by doing this, get a simpler or "reduced" model. Simple models can be good as they are easier to groups that suggests no difference in the groups is not a very exciting result in most, but not all, situations[2]. In order for R to provide results for the mean-only model, we remove the grouping variable, `Attr`, from the model formula and just include a "1". The `(Intercept)` row of the output provides the estimate for the mean-only model as a reduced model from either the cell-means or reference-coded models when we assume that the mean is the same for all groups:

```
lm3 <- lm(Years ~ 1, data=MockJury)
summary(lm3)
```

```
## $coefficients
##             Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 4.692982  0.3403532 13.78857 5.765681e-26
```

This model provides an estimate of the common mean for all observations of $4.693 = \hat{\mu} = \hat{\alpha}$ years. This value also is the dashed, horizontal line in the beanplot in Figure 3.1. Some people call this mean-only estimate the grand or overall mean.

3.2 Linear model for One-Way ANOVA (cell-means and reference-coding)

The previous discussion showed two ways of parameterizing models for the One-Way ANOVA model and getting estimates from output but still hasn't addressed how to assess evidence related to whether the observed differences in the means among the groups is "real". In this section, we develop what is called the ***ANOVA F-test*** that provides a method of aggregating the differences among the means of 2 or more groups and testing our null hypothesis of no difference in the means vs the alternative. In order to develop the test, some additional notation is needed. The sample size in each group is denoted n_j and the total sample size is $\boldsymbol{N = \Sigma n_j = n_1 + n_2 + \ldots + n_J}$ where Σ (capital sigma) means "add up over whatever follows". An estimated ***residual*** (e_{ij}) is the difference between an observation, y_{ij}, and the model estimate, $\hat{y}_{ij} = \hat{\mu}_j$, for

[2]Suppose we were doing environmental monitoring and were studying asbestos levels in soils. We might be hoping that the mean-only model were reasonable to use if the groups being compared were in remediated areas and in areas known to have never been contaminated.

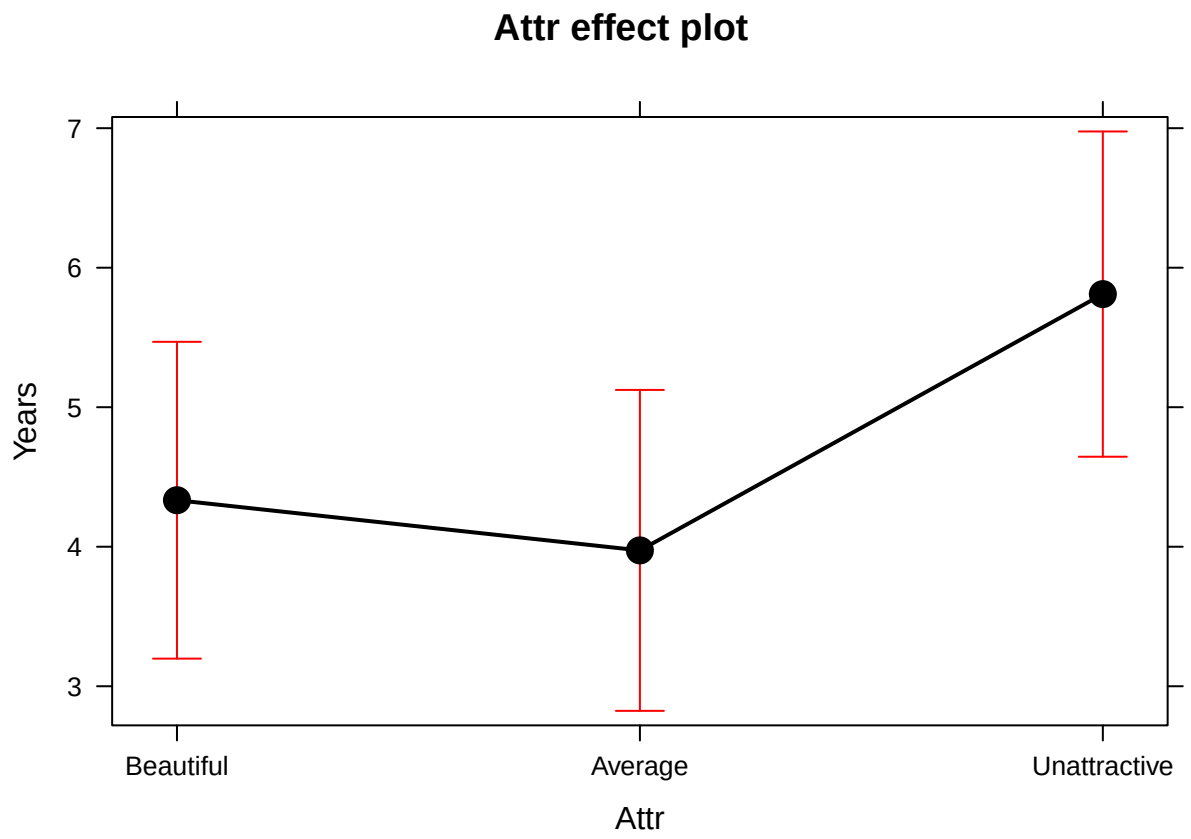


Figure 3.2: Plot of the estimated group mean sentences from the reference-coded model for the MockJury data.

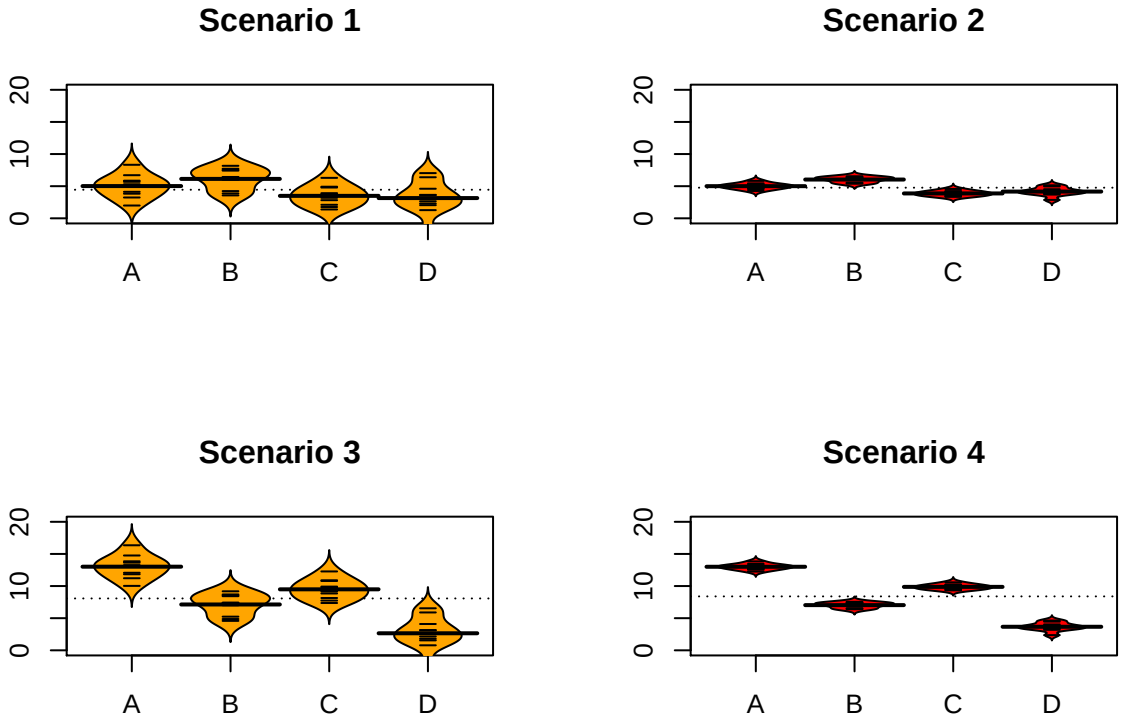


Figure 3.3: Demonstration of different amounts of difference in means relative to variability. Scenarios have same means in rows and same variance around means in columns of plot.

that observation, $y_{ij} - \hat{y}_{ij} = e_{ij}$. It is basically what is left over that the mean part of the model ($\hat{\mu}_j$) does not explain. It is also a window into how "good" the model might be.

Consider the four different fake results for a situation with four groups ($J = 4$) displayed in Figure 3.3. Which of the different results shows the most and least evidence of differences in the means? In trying to answer this, think about both how different the means are (obviously important) and how variable the results are around the mean. These situations were created to have the same means in Scenarios 1 and 2 as well as matching means in Scenarios 3 and 4. The variability around the means matches by shading (lighter or darker). In Scenarios 1 and 2, the differences in the means is smaller than in the other two results. But Scenario 2 should provide more evidence of what little difference in present than Scenario 1 because it has less variability around the means. The best situation for finding group differences here is Scenario 4 since it has the largest difference in the means and the least variability around those means. Our test statistic somehow needs to allow a comparison of the variability in the means to the overall variability to help us get results that reflect that Scenario 4 has the strongest evidence of a difference and Scenario 1 would have the least.

The statistic that allows the comparison of relative amounts of variation is called the ***ANOVA F-statistic*** . It is developed using ***sums of squares*** which are measures of total variation like are used in the numerator of the standard deviation ($\Sigma_1^N (y_i - \bar{y})^2$) that took all the observations, subtracted the mean, squared the differences, and then added up the results over all the observations to generate a measure of total variability. With multiple groups, we will focus on decomposing that total variability (***Total Sums of Squares***) into variability among the means (we'll call this ***Explanatory Variable*** **A's** ***Sums of Squares***) and variability in the residuals or errors (***Error Sums of Squares***). We define each of these quantities in the One-Way ANOVA situation as follows:

- $\mathbf{SS_{Total}}$ = Total Sums of Squares = $\Sigma_{j=1}^{J}\Sigma_{i=1}^{n_j}(y_{ij}-\bar{\bar{y}})^2$
  - This is the total variation in the responses around the overall or ***grand mean*** ($\bar{\bar{y}}$, the estimated mean for all the observations and available from the mean-only model).
  - By summing over all n_j observations in each group, $\Sigma_{i=1}^{n_j}(\ )$, and then adding those results up across the groups, $\Sigma_{j=1}^{J}(\ )$, we accumulate the variation across all N observations.
  - Note: this is the residual variation if the null model is used, so there is no further decomposition possible for that model.
  - This is also equivalent to the numerator of the sample variance, $\Sigma_1^N(y_i-\bar{y})^2$ which is what you get when you ignore the information on the potential differences in the groups.
- $\mathbf{SS_A}$ = Explanatory Variable A's Sums of Squares = $\Sigma_{j=1}^{J}\Sigma_{i=1}^{n_j}(\bar{y}_i-\bar{\bar{y}})^2 = \Sigma_{j=1}^{J}n_j(\bar{y}_i-\bar{\bar{y}})^2$
  - This is the variation in the group means around the grand mean based on the explanatory variable A.
  - Also called sums of squares for the treatment, regression, or model.
- $\mathbf{SS}_E$ = Error (Residual) Sums of Squares = $\Sigma_{j=1}^{J}\Sigma_{i=1}^{n_j}(y_{ij}-\bar{y})^2 = \Sigma_{j=1}^{J}\Sigma_{i=1}^{n_j}(e_{ij})^2$
  - This is the variation in the responses around the group means.
  - Also called the sums of squares for the residuals, with the second version of the formula showing that it is just the squared residuals added up across all the observations.

The possibly surprising result given the mass of notation just presented is that the total sums of squares is **ALWAYS** equal to the sum of explanatory variable A's sum of squares and the error sums of squares,

$$\mathbf{SS_{Total}}=\mathbf{SS_A}+\mathbf{SS_E}.$$

This equality means that if the $\mathbf{SS_A}$ goes up, then the $\mathbf{SS_E}$ must go down if $\mathbf{SS_{Total}}$ remains the same. This result is called the ***sums of squares decomposition formula***. We use these results to build our test statistic and organize this information in what is called an ***ANOVA table***. The ANOVA table is generated using the `anova` function applied to the reference-coded model, `lm2` :

```
lm2<-lm(Years ~ Attr, data=MockJury)
anova(lm2)
```

```
## Analysis of Variance Table
##
## Response: Years
##            Df  Sum Sq Mean Sq F value Pr(>F)
## Attr        2   70.94  35.469    2.77  0.067 .
## Residuals 111 1421.32  12.805
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that the ANOVA table has a row labelled `Attr`, which contains information for the grouping variable (we'll generally refer to this as explanatory variable A but here it is the picture group that was randomly assigned), and a row labelled `Residuals`, which is synonymous with "Error". The Sums of Squares (SS) are available in the `Sum Sq` column. It doesn't show a row for "Total" but the $\mathbf{SS_{Total}}=\mathbf{SS_A}+\mathbf{SS_E}$ = 1492.26.

```
70.94 + 1421.32
```

```
## [1] 1492.26
```

It may be easiest to understand the *sums of squares decomposition* by connecting it to our permutation ideas. In a permutation situation, the total variation (SS_{Total}) cannot change – it is the same responses varying around the grand mean. However, the amount of variation attributed to variation among the means and in

the residuals can change if we change which observations go with which group. In Figure 3.4 (panel a), the means, sums of squares, and 95% confidence intervals for each mean are displayed for the three treatment levels from the original prisoner rating data. Three permuted versions of the data set are summarized in panels (b), (c), and (d). The SS_A is 70.9 in the real data set and between 6.6 and 11 in the permuted data sets. If you had to pick among the plots for the one with the most evidence of a difference in the means, you hopefully would pick panel (a). This visual "unusualness" suggests that this observed result is unusual relative to the possibilities under permutations, which are, again, the possibilities tied to having the null hypothesis being true. But note that the differences here are not that great between these three permuted data sets and the real one. It is likely that at least some of you might have selected panel (d) as also looking like it shows some evidence of differences (maybe not the most?) as it also looks like it shows some evidence differences.

One way to think about $\mathbf{SS_A}$ is that it is a function that converts the variation in the group means into a single value. This makes it a reasonable test statistic in a permutation testing context. By comparing the observed $SS_A = 70.9$ to the permutation results of 6.5, 9.7, and 40.5 we see that the observed result is much more extreme than the three alternate versions. In contrast to our previous test statistics where positive and negative differences were possible, SS_A is always positive with a value of 0 corresponding to no variation in the means. The larger the SS_A, the more variation there is in the means. The permutation p-value for the alternative hypothesis of **some** (not of greater or less than!) difference in the true means of the groups will involve counting the number of permuted SS_A^* results that are larger than what we observed.

```
## [1] 70.93836
```

To do a permutation test, we need to be able to calculate and extract the SS_A value. In the ANOVA table, it is in the first row and is the second number and we can use the bracket, `[, ]`, referencing to extract that number from the ANOVA table that `anova` produces with `anova(lm(Years~Attr, data=MockJury))[1, 2]`. We'll store the observed value of SS_A in `Tobs`, reusing some ideas from Chapter @ref{chapter2}.

```
Tobs <- anova(lm(Years~Attr,data=MockJury))[1,2]; Tobs
```

```
## [1] 70.93836
```

The following code performs the permutations `B=1,000` times using the `shuffle` function, builds up a vector of results in `Tobs`, and then makes a plot of the resulting permutation distribution:

```
par(mfrow=c(1,2))
B<- 1000
Tstar<-matrix(NA,nrow=B)
for (b in (1:B)){
  Tstar[b]<-anova(lm(Years~shuffle(Attr),data=MockJury))[1,2]
  }
hist(Tstar,labels=T,ylim=c(0,550))
abline(v=Tobs,col="red",lwd=3)
plot(density(Tstar),main="Density curve of Tstar")
abline(v=Tobs,col="red",lwd=3)
```

The right-skewed distribution (Figure 3.5) contains the distribution of SS_A's under permutations (where all the groups are assumed to be equivalent under the null hypothesis). While the observed result is larger than many of the SS_A's, there are also many permuted results that are much larger than observed. The proportion of permuted results that exceed the observed value is found using `pdata` as before, except only for the area to the right of the observed result. We know that `Tobs` will always be positive so no absolute values are required here.

```
pdata(Tstar,Tobs,lower.tail=F)
```

```
## [1] 0.072
```

This provides a permutation-based p-value of 0.072 and suggests marginal evidence against the null hypothesis of no difference in the true means. We would interpret this p-value as saying that there is a 7.2% chance of

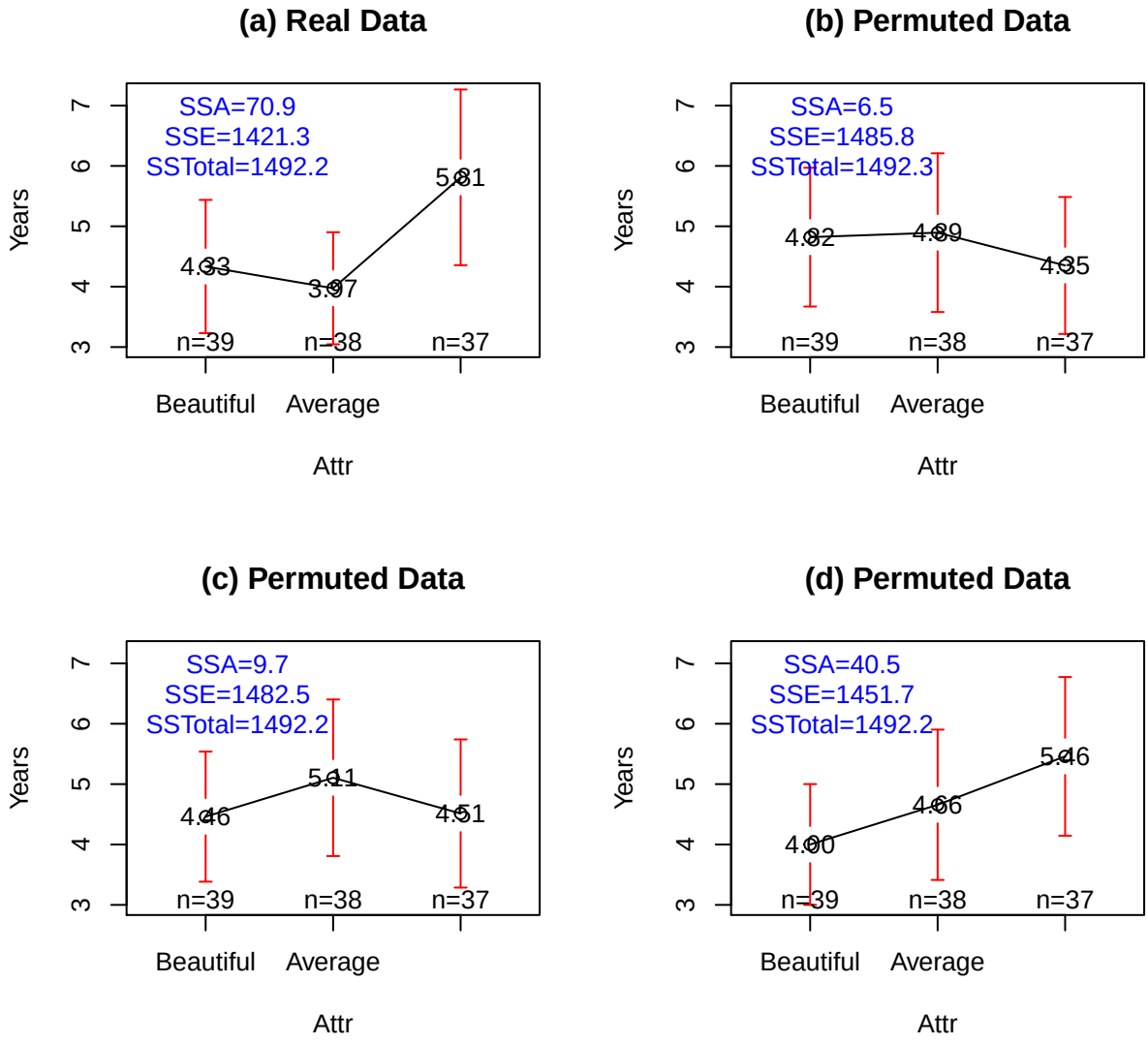


Figure 3.4: Plot of means and 95% confidence intervals for the three groups for the real data (a) and three different permutations of the treatment labels to the same responses in (b), (c), and (d). Note that SSTotal is always the same but the different amounts of variation associated with the means (SSA) or the errors (SSE) changes in permutation.

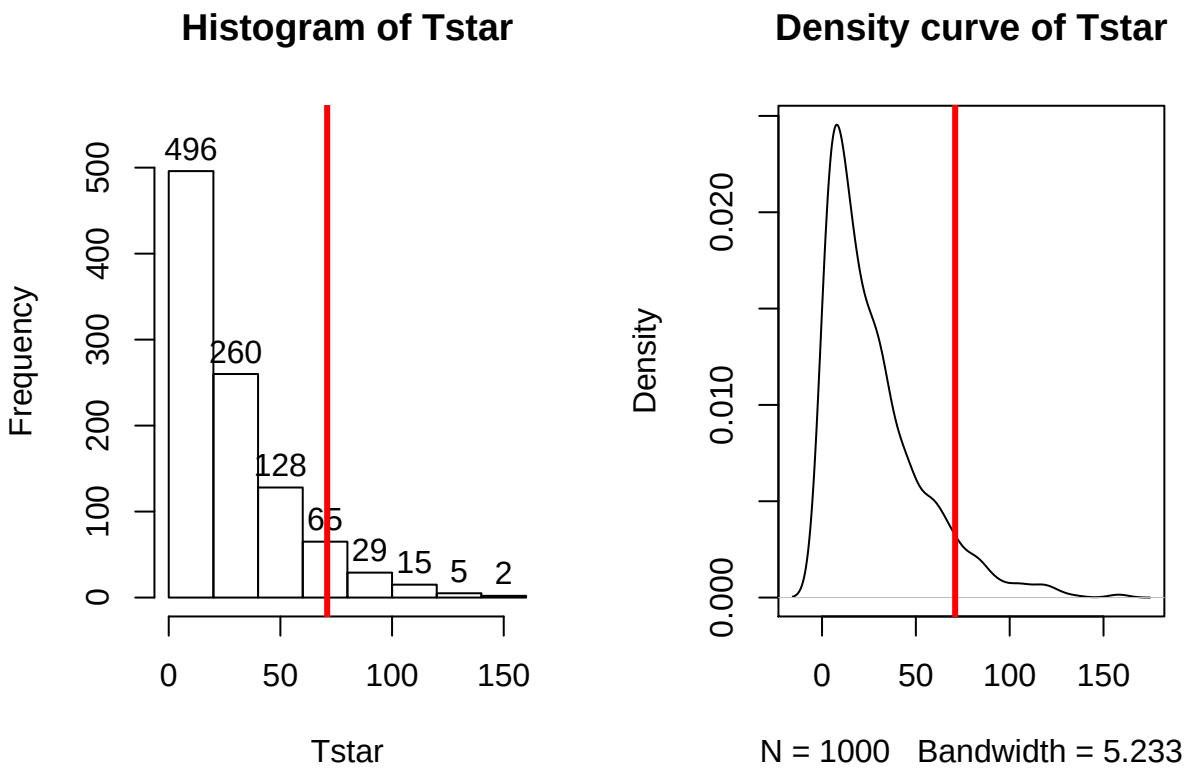


Figure 3.5: Histogram and density curve of permutation distribution of SS_A with the observed value of SS_A displayed as a bold, vertical line. The proportion of results that are larger than the observed value of SS_A provides an estimate of the p-value.

getting a SS_A as large or larger than we observed, given that the null hypothesis is true.

It ends up that some nice parametric statistical results are available (if our assumptions are met) for the ratio of estimated variances, which are called ***Mean Squares***. To turn sums of squares into mean square (variance) estimates, we divide the sums of squares by the amount of free information available. For example, remember the typical variance estimator introductory statistics, $\Sigma_1^N(y_i - \bar{y})^2/(N-1)$? Your instructor spent some time trying various approaches to explaining why we have a denominator of $N-1$. The most useful for our purposes moving forward is that we "lose" one piece of information to estimate the mean and there are N deviations around the single mean so we divide by $N-1$. The main point is that the sums of squares were divided by something and we got an estimator for the variance, here of the observations.

Now consider $\text{SS}_E = \Sigma_{j=1}^J \Sigma_{i=1}^{n_j}(y_i - \bar{y})^2$ which still has N deviations but it varies around the J means, so the

$$\textbf{Mean Square Error} = \text{MS}_E = \text{SS}_E/(N-J).$$

Basically, we lose J pieces of information in this calculation because we have to estimate J means.

The similar calculation of the ***Mean Square for variable*** **A** (MS_A) is harder to see in the formula ($\text{SS}_A = \Sigma_{j=1}^J n_j(\bar{y}_i - \bar{\bar{y}})^2$), but the same reasoning can be used to understand the denominator for forming MS_A: there are J means that vary around the grand mean so

$$\text{MS}_A = \text{SS}_A/(J-1).$$

In summary, the two mean squares are simply:

- $\text{MS}_A = \text{SS}_A/(J-1)$, which estimates the variance of the group means around the grand mean.
- $\text{MS}_{\text{Error}} = \text{SS}_{\text{Error}}/(N-J)$, which estimates the variation of the errors around the group means.

These results are put together using a ratio to define the ***ANOVA F-statistic*** (also called the ***F-ratio***) as

$$F = \text{MS}_A/\text{MS}_{\text{Error}}.$$

If the variability in the means is "similar" to the variability in the residuals, the statistic would have a value around 1. If that variability is similar then there be no evidence of a difference in the means. If the MS_A is much larger than the MS_E, the F-statistic will provide evidence against the null hypothesis. The "size" of the F-statistic is formalized by finding the p-value. The F-statistic, if assumptions discussed below are met and we assume the null hypothesis is true, follows what is called an F-distribution. The ***F-distribution*** is a right-skewed distribution whose shape is defined by what are called the ***numerator degrees of freedom*** ($J-1$) and the ***denominator degrees of freedom*** ($N-J$). These names correspond to the values that we used to calculate the mean squares and where in the F-ratio each mean square was used; F-distributions are denoted by their degrees of freedom using the convention of F (*numerator df*, *denominator df*). Some examples of different F-distributions are displayed for you in Figure 3.6.

The characteristics of the F-distribution can be summarized as:

- Right skewed,
- Nonzero probabilities for values greater than 0,
- Its shape changes depending on the **numerator** and **denominator DF**, and
- **Always use the right-tailed area for p-values.**

Now we are ready to discuss an ANOVA table since we know about each of its components. Note the general format of the ANOVA table is[3]:

[3]Make sure you can work from left to right and up and down to fill in the ANOVA table given just the necessary information to determine the other components – there is always a question like this on the exam...

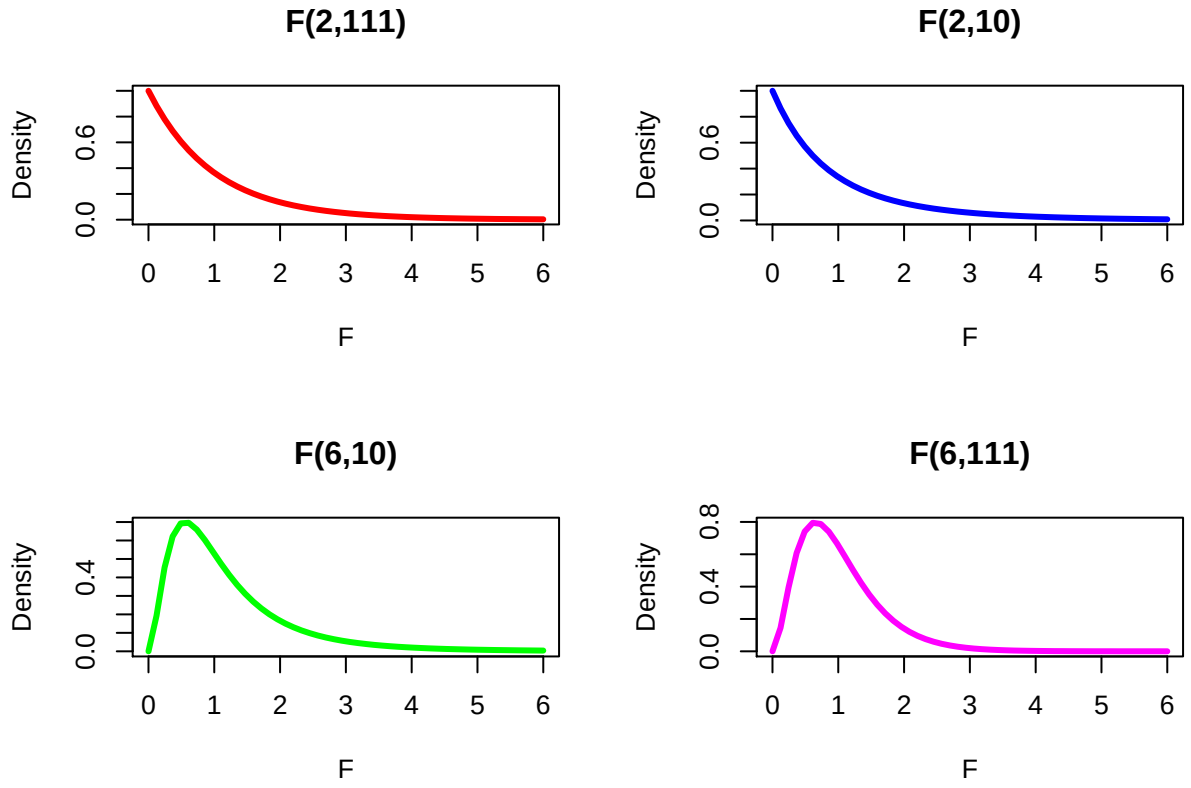


Figure 3.6: Density curves of four different F-distributions. Upper left is an $F(2, 111)$, upper right is $F(2, 10)$, lower left is $F(6, 10)$, and lower right is $F(6, 111)$. P-values are found using the areas to the right of the observed F-statistic value.

Table 3.3: General One-Way ANOVA table.

Source	DF	Sums of Squares	Mean Squares	F-ratio
Variable A	$J-1$	SS_A	$\text{MS}_A=\text{SS}_A/(J-1)$	$F=\text{MS}_A/\text{[illegible}
Residuals	$N-J$	SS_E	$\text{MS}_E = \text{SS}_E/(N-J)$	
Total	$N-1$	SS_{Total}		

The table is oriented to help you reconstruct the F-ratio from each of its components. The output from R is similar although it does not provide the last row and sometimes switches the order of columns. The R version of the table for the type of picture effect (`Attr`) with $J = 3$ levels and $N = 114$ observations, repeated from above, is:

```
anova(lm2)
```

```
## Analysis of Variance Table
##
## Response: Years
##            Df  Sum Sq Mean Sq F value Pr(>F)
## Attr        2   70.94  35.469    2.77  0.067 .
## Residuals 111 1421.32  12.805
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value from the F-distribution is 0.067. We can verify this result using the observed F-statistic of 2.77 (which came from taking the ratio of the two mean squares, F=35.47/12.8) which follows an $F(2, 111)$ distribution if the null hypothesis is true and some other assumptions are met. Using the `pf` function provides us with areas in the specified F-distribution with the `df1` provided to the function as the numerator *df* and `df2` as the denominator *df* and `lower.tail=F` reflecting our desire for a right tailed area.

```
pf(2.77,df1=2,df2=111,lower.tail=F)
```

```
## [1] 0.06699803
```

The result from the F-distribution using this parametric procedure is similar to the p-value obtained using permutations with the test statistic of the SS_A, which was 70.9. The F-statistic obviously is another potential test statistic to use as a test statistic in a permutation approach, now that we know about it. We should check that we get similar results from it with permutations as we did from using SS_A as a permutation test test statistic. The following code generates the permutation distribution for the F-statistic (Figure 3.7) and assesses how unusual the observed F-statistic of 2.77 was in this permutation distribution. The only change in the code involves moving from extracting SS_A to extracting the F-ratio which is in the 4th column of the `anova` output:

```
Tobs <- anova(lm(Years~Attr,data=MockJury))[1,4]; Tobs
```

```
## [1] 2.770024
```

```
par(mfrow=c(1,2))
B<- 1000
Tstar<-matrix(NA,nrow=B)
for (b in (1:B)){
  Tstar[b]<-anova(lm(Years~shuffle(Attr),data=MockJury))[1,4]
}

pdata(Tstar,Tobs,lower.tail=F)
```

```
## [1] 0.064
```

```
hist(Tstar,labels=T)
abline(v=Tobs,col="red",lwd=3)
plot(density(Tstar),main="Density curve of Tstar")
abline(v=Tobs,col="red",lwd=3)
```

The permutation-based p-value is 0.064 which, again, matches the other results closely. The first conclusion is that using a test statistic of either the F-statistic or the SS_A provide similar permutation results. However, we tend to favor using the F-statistic because it is more commonly used in reporting ANOVA results, not because it is any better in a permutation context.

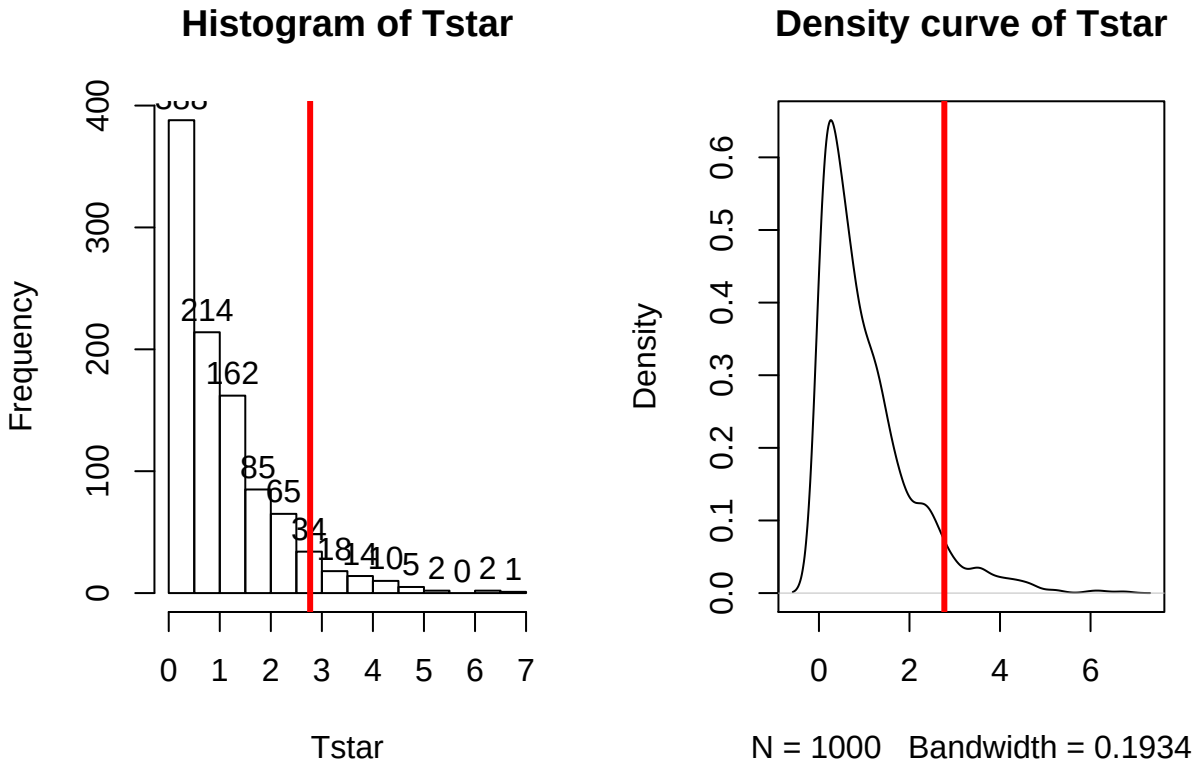


Figure 3.7: Histogram and density curve of the permutation distribution of the F-statistic with bold, vertical line for observed value of the test statistic of 2.77.

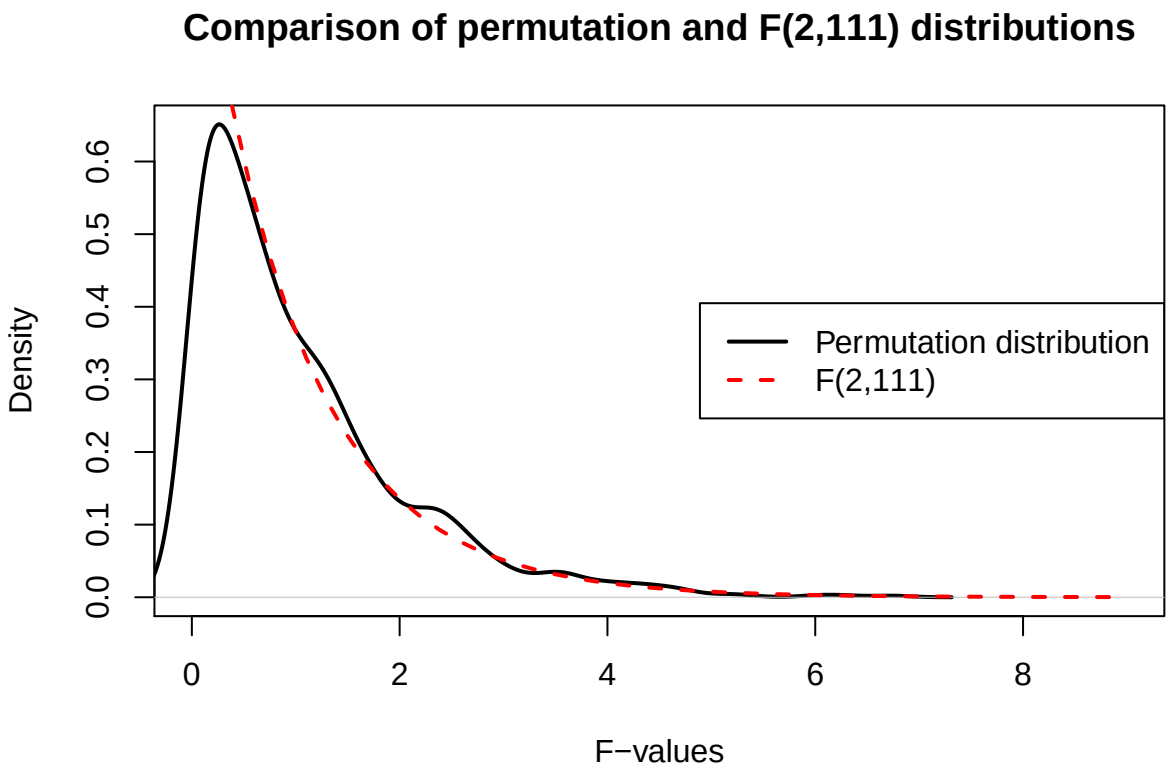


Figure 3.8: Comparison of $F(2, 111)$ (dashed line) and permutation distribution (solid line).

It is also interesting to compare the permutation distribution for the F-statistic and the parametric $F(2, 111)$ distribution (Figure 3.8). They do not match perfectly but are quite similar. Some the differences around 0 are due to the behavior of the method used to create the density curve and are not really a problem for the methods. The similarity in the two curves explains why both methods give similar results. In some situations, the correspondence will not be quite so close.

So how can we rectify this result (p-value ≈ 0.06) and the Chapter 2 result that detected a difference between *Average* and *Unattractive* with a p-value ≈ 0.03? I selected the two groups to compare in Chapter 2 because they were furthest apart. "Cherry-picking" the comparison that is likely to be most different creates a false sense of the real situation and inflates the Type I error rate because of the selection. If the entire suite of pairwise comparisons are considered, this result may lose some of its luster. In other words, if we consider the suite of three pair-wise differences (and the tests) implicit in comparing all of them, we may need stronger evidence in the most different pair than a p-value of 0.033 to suggest overall differences. In this situation, the *Beautiful* and *Average* groups are not that different from each other so their difference does not contribute much to the will revisit this topic and consider a method that is statistically valid for performing all possible pair-wise comparisons that is also consistent with our overall test results.

3.3 One-Way ANOVA Sums of Squares, Mean Squares, and F-test

The requirements for a One-Way ANOVA F-test are similar to those discussed in Chapter 2, except that there are now J groups instead of only 2. Specifically, the linear model assumes:

1. **Independent observations**,

2. **Equal variances**, and
3. **Normal distributions**.

For assessing equal variances across the groups, it is best to use plots to assess this. We can use boxplots and beanplots to compare the spreads of the groups, which were provided in Figure 3.1. The range and IQRs should be relatively similar across the groups if you do not find evidence of a problem with this assumption. You should start with noting how clear or big the violation of the assumption might be but remember that there will always be some differences in the variation among groups even if the true variability is exactly equal in the populations. In addition to our direct plotting, there are some diagnostic plots available from the `lm` function that can help us more clearly assess potential violations of the previous assumptions.

We can obtain a suite of four diagnostic plots by using the `plot` function on any linear model object that we have fit. To get all the plots together in four panels we need to add the `par(mfrow=c(2, 2))` command to tell R to make a graph with 4 panels[4].

```
par(mfrow=c(2,2))
plot(lm2,pch=16)
```

There are two plots in Figure 3.9 with useful information for the equal variance assumption. The "Residuals vs Fitted" panel in the top left displays the residuals ($e_{ij} = y_{ij} - \hat{y}_i j$) on the y-axis and the fitted values ($\hat{y}_{ij}$) on the x-axis. This allows you to see if the variability of the observations differs across the groups as a function of the mean of the groups because all the observations in the same group get the same fitted value, the mean of the group. In this plot, the points seem to have fairly similar spreads at the fitted values for the three groups with fitted values of 4, 4.3, and 6. The "Scale-Location" plot in the lower left panel has the same x-axis but the y-axis contains the square-root of the absolute value of the standardized residuals. The absolute value transforms all the residuals into a magnitude scale (removing direction) and the square-root helps you see differences in variability more accurately. The standardization scales them to have a variance of 1 so help you in other displays to get a sense of how many standard deviations you are away from the mean in the residual distribution. The visual assessment is similar in the two plots – you want to consider whether it appears that the groups have somewhat similar or noticeably different amounts of variability. If you see a clear funnel shape in the Residuals vs Fitted or an increase or decrease in the upper edge of points in the Scale-Location plot that may indicate a violation of the constant variance assumption. Remember that some variation across the groups is expected and is OK, but large differences in spreads are problematic for all the procedures that involve linear models. When discussing these results, you want to discuss how clearly the differences in variation are and whether that *shows a clear violation of the assumption* of equal variance for all observations. Like in hypothesis testing, you can't prove that you've met assumptions based on a plot "looking OK", but you can say that there is no clear evidence that the assumption is violated!

The linear model assumes that all the random errors (ϵ_{ij}) follow a normal distribution. To gain insight into the validity of this assumption, we can explore the original observations as displayed in the beanplots, mentally subtracting off the differences in the means and focusing on the shapes of the distributions of observations in each group. These plots are especially good for assessing whether there is there a skew or outliers present in each group. If so, by definition, the normality assumption is violated. But our assumption is about the distribution of all the errors after the remove the differences in the means and so we want an overall assessment technique to understand how reasonable our assumption is overall for our model. The residuals from the entire model provide us with estimates of the random errors and if the normality assumption is met, then the residuals all-together should approximately follow a normal distribution. The ***Normal Q-Q Plot*** in upper right panel of Figure 3.9 is a direct visual assessment of how well our residuals match what we would expect from a normal distribution. Outliers, skew, heavy and light-tailed aspects of distributions (all violations of normality) show up in this plot once you learn to read it – which is our next task. To make it easier to read QQ-plots, it is nice to start with just considering histograms and/or density plots of the residuals and to see how that maps into this new display. We can obtain the residuals from the linear model using the `residuals` function on any linear model object.

[4]We have been using this function quite a bit to make multi-panel graphs but did not show you that line of code. But you need to use this command for linear model diagnostics or you won't get the plots we want from the model. And you really just need `plot(lm2)` but the `pch=16` option makes it easier to see some of the points in the plots.

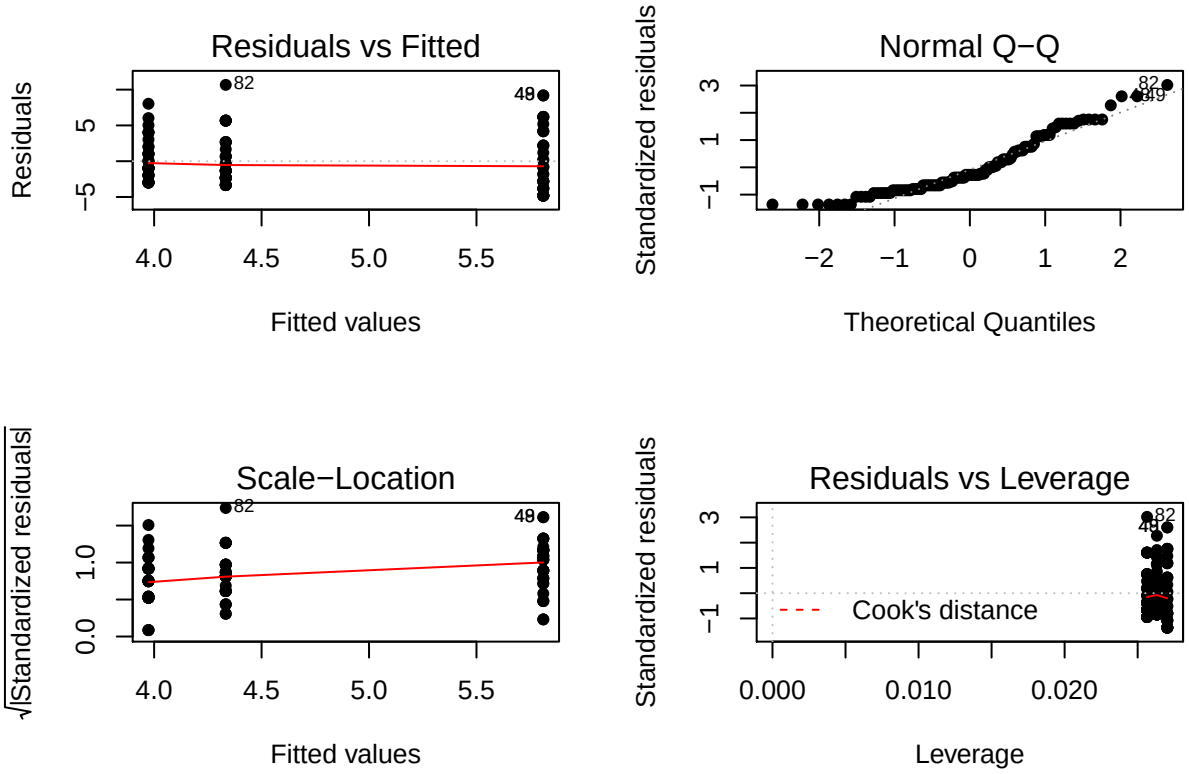


Figure 3.9: Default diagnostic plots for the linear model.

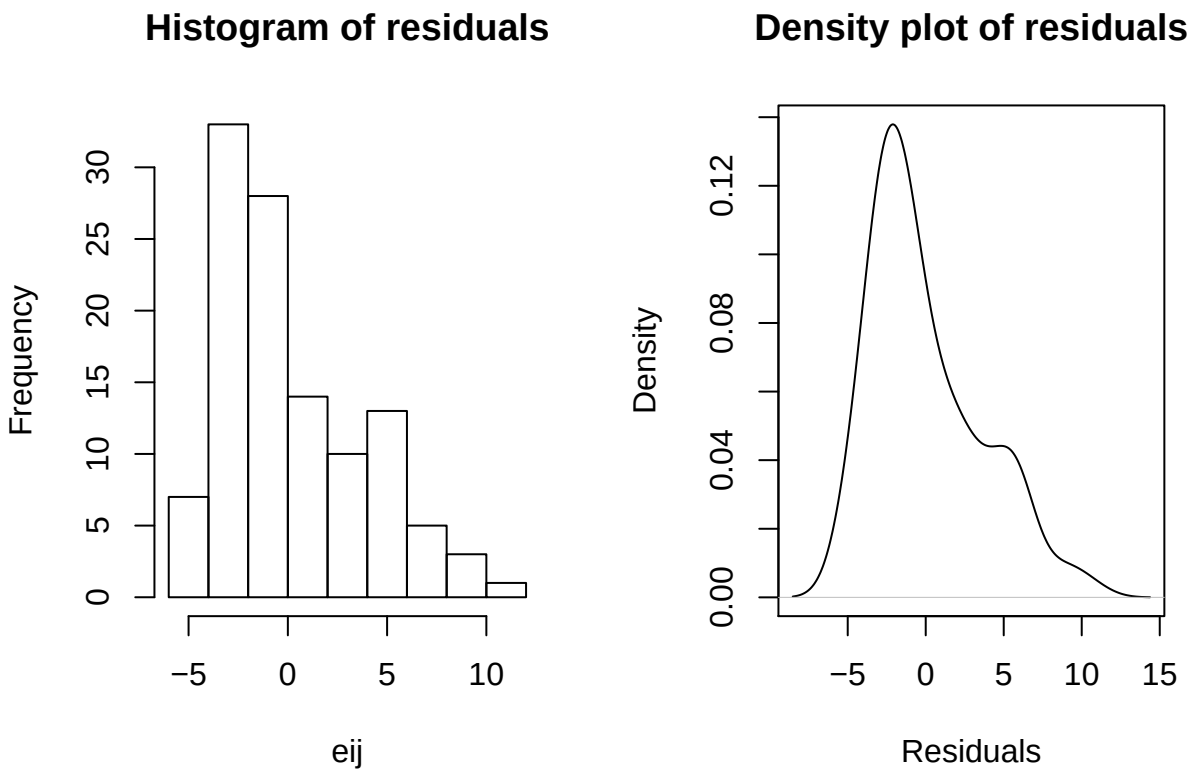


Figure 3.10: Histogram and density curve of the linear model raw residuals.

```
par(mfrow=c(1,2))
eij<-residuals(lm2)
hist(eij, main="Histogram of residuals")
plot(density(eij), main="Density plot of residuals", ylab="Density",
     xlab="Residuals")
```

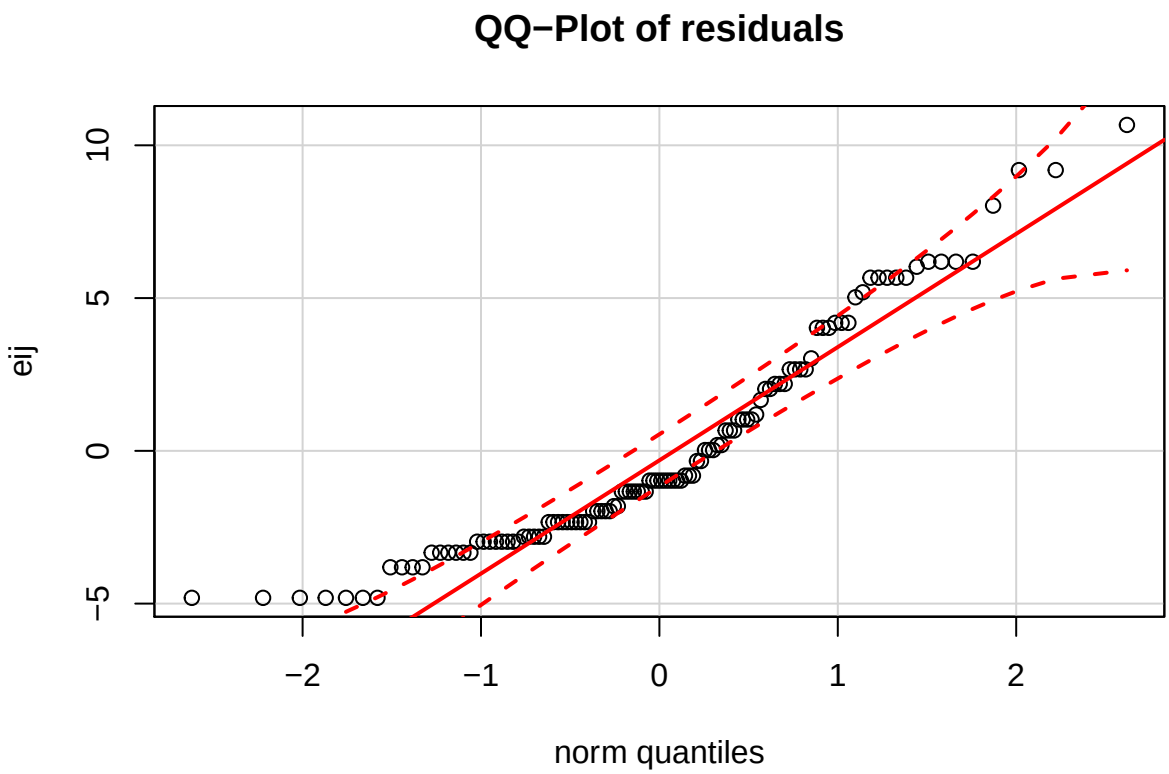
Figure 3.10 shows that there is a right skew present in the residuals for the prisoner rating data model that accounted for different means in the three groups, which is consistent with the initial assessment of some right skew in the plots of observations in each group.

A Quantile-Quantile plot (***QQ-plot***) shows the "match" of an observed distribution with a theoretical distribution, almost always the normal distribution. They are also known as Quantile Comparison, Normal Probability, or Normal Q-Q plots, with the last two names being specific to comparing results to a normal distribution. In this version[5], the QQ-plots display the value of observed percentiles in the residual distribution on the y-axis versus the percentiles of a theoretical normal distribution on the x-axis. If the observed **distribution of the residuals matches the shape of the normal distribution, then the plotted points should follow a 1-1 relationship.** If the points follow the displayed straight line then that suggests that the residuals have a similar shape to a normal distribution. Some variation is expected around the line and some patterns of deviation are worse than others for our models, so you need to go beyond saying "it does not match a normal distribution". It is best to be specific about the type of deviation you are detecting.

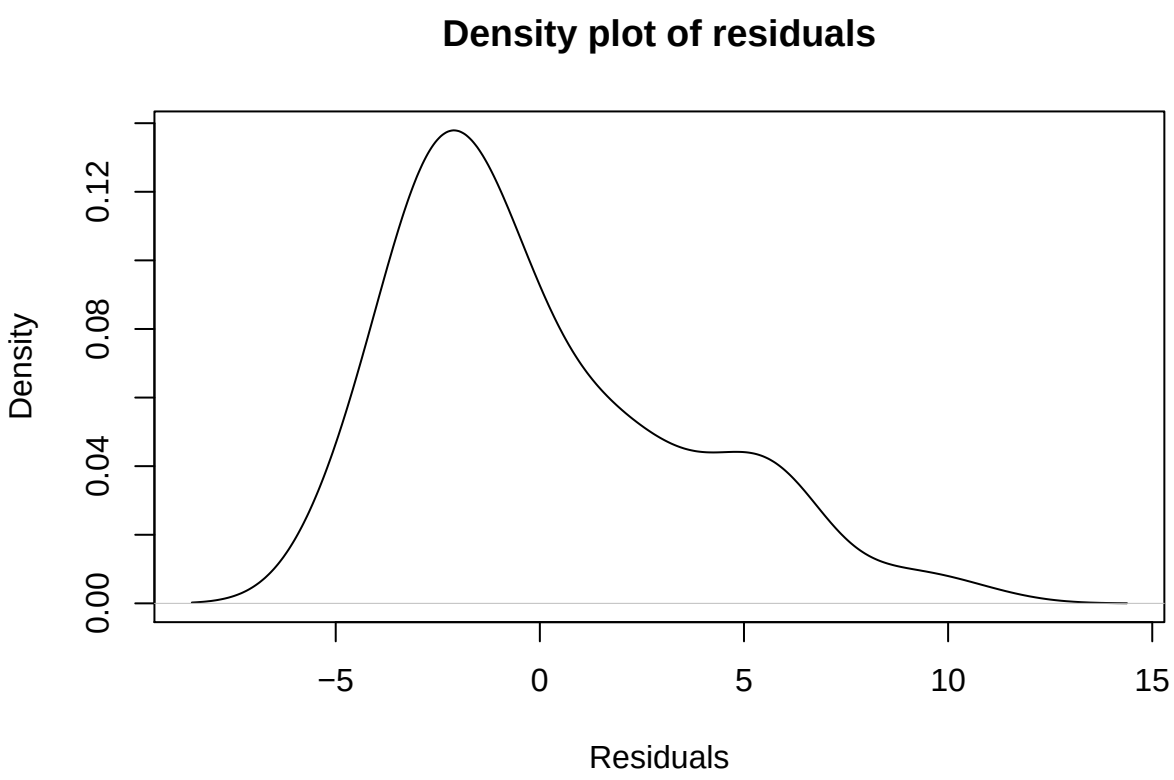
[5]Along with multiple names, there is variation of what is plotted on the x and y axes and the scaling of the values plotted, increasing the challenge of interpreting QQ-plots. We are consistent about the x and y axis choices but different functions that make these plots in R do switch the axes.

And to do that, we need to practice interpreting some QQ-plots.

The QQ-plot of the linear model residuals from Figure 3.9 is extracted and enhanced it a little to make Figure **??** so we can just focus on it. We know from looking at the histogram that this is a slightly right skewed distribution. The QQ-plot places the observed ***standardized***[6] ***residuals*** on the y-axis and the theoretical normal values on the x-axis. The most noticeable deviation from the 1-1 line is in the lower left corner of the plot. These are for the negative residuals (left tail) and there are many residuals at around the same value that are a little smaller than -1. If the distribution had followed the normal distribution here, the points would be on the 1-1 line and there would be some standardized residuals much smaller than -1.5. So we are not getting as much spread in the smaller residuals as we would expect in a normal distribution. If you go back to the histogram you can see that the smallest residuals are all stacked up and do not spread out like the left tail of a normal distribution should. In the right tail (positive) residuals, there is also a systematic lifting from the 1-1 line to larger values in the residuals than the normal would generate. For example, the point labeled as "82" (the 82nd observation in the data set) has a value of 3 in residuals but should actually be smaller (maybe 2.5) if the distribution was normal. Put together, this pattern in the QQ-plot suggests that the left tail is too compacted (too short) and the right tail is too spread out – this is the right skew we identified from the histogram and density curve!



[6]Here this means re-scaled so that they should have similar scaling to a standard normal with mean 0 and standard deviation 1. This does not change the shape of the distribution but can make outlier identification simpler – having a standardized residual more extreme than 5 or -5 would suggest a deviation from normality since we rarely see values that many standard deviations from the mean in a normal distribution. But mainly focus on the shape of the pattern in the QQ-plot.



Generally, when both tails deviate on the same side of the line (forming a sort of quadratic curve, especially in more extreme cases), that is evidence of a skew. To see some different potential shapes in QQ-plots, six different data sets are displayed in Figures 3.11 and 3.12. In each row, a QQ-plot and associated density curve are displayed. If the points are both above the 1-1 line in the lower and upper tails as in Figure 3.11(a), then the pattern is a right skew, here even more extreme than in the real data set. If the points are below the 1-1 line in both tails as in Figure 3.11(c), then the pattern is identified as a left skew. Skewed residual distributions (either direction) are problematic for models that assume normally distributed responses but not necessarily for our permutation approaches if all the groups have similar skewed shapes. The other problematic pattern is to have more spread than a normal curve as in Figure 3.11(e) and (f). This shows up with the points being below the line in the left tail (more extreme negative than expected by the normal) and the points being above the line for the right tail (more extreme positive than the normal predicts). We call these distributions ***heavy-tailed*** which can manifest as distributions with outliers in both tails or just a bit more spread out than a normal distribution. Heavy-tailed residual distributions can be problematic for our models as the variation is greater than what the normal distribution can account for and our methods might under-estimate the variability in the results. The opposite pattern with the left tail above the line and the right tail below the line suggests less spread (***lighter-tailed***) than a normal as in Figure 3.11(g) and (h). This pattern is relatively harmless and you can proceed with methods that assume normality safely as they will just be a little conservative.

Finally, to help you calibrate expectations for data that are actually normally distributed, two data sets simulated from normal distributions are displayed in Figure 3.12. Note how neither follows the line exactly but that the overall pattern matches fairly well. **You have to allow for some variation from the line in real data sets** and focus on when there are really noticeable issues in the distribution of the residuals such as those displayed above. Again, you will never be able to prove that you have normally distributed residuals even if the residuals are all exactly on the line, but if you see QQ-plots as in Figure 3.11 you can encounter situations that provide evidence of clear violations of the normality assumption.

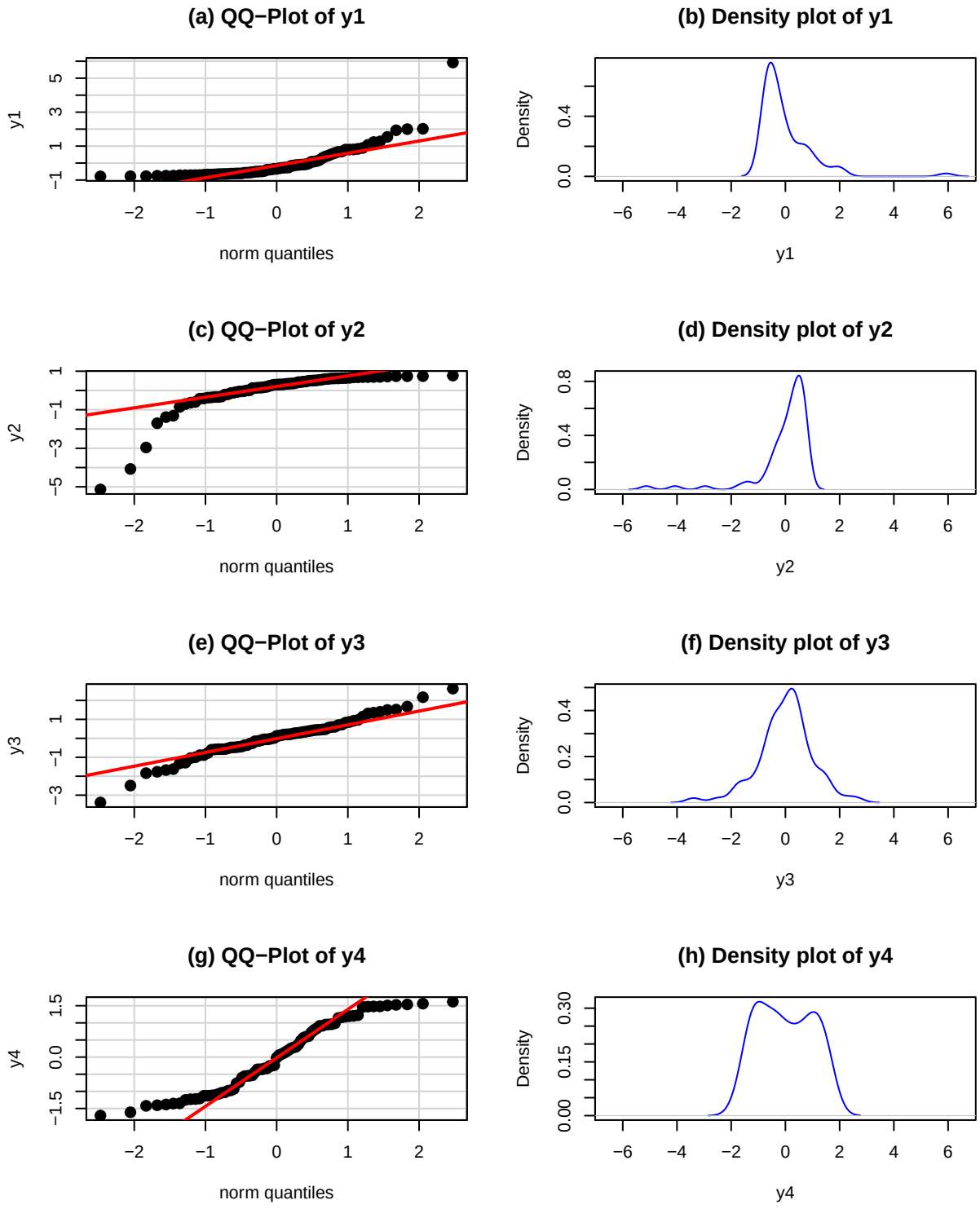


Figure 3.11: QQ-plots and density curves of four simulated distributions with different shapes.

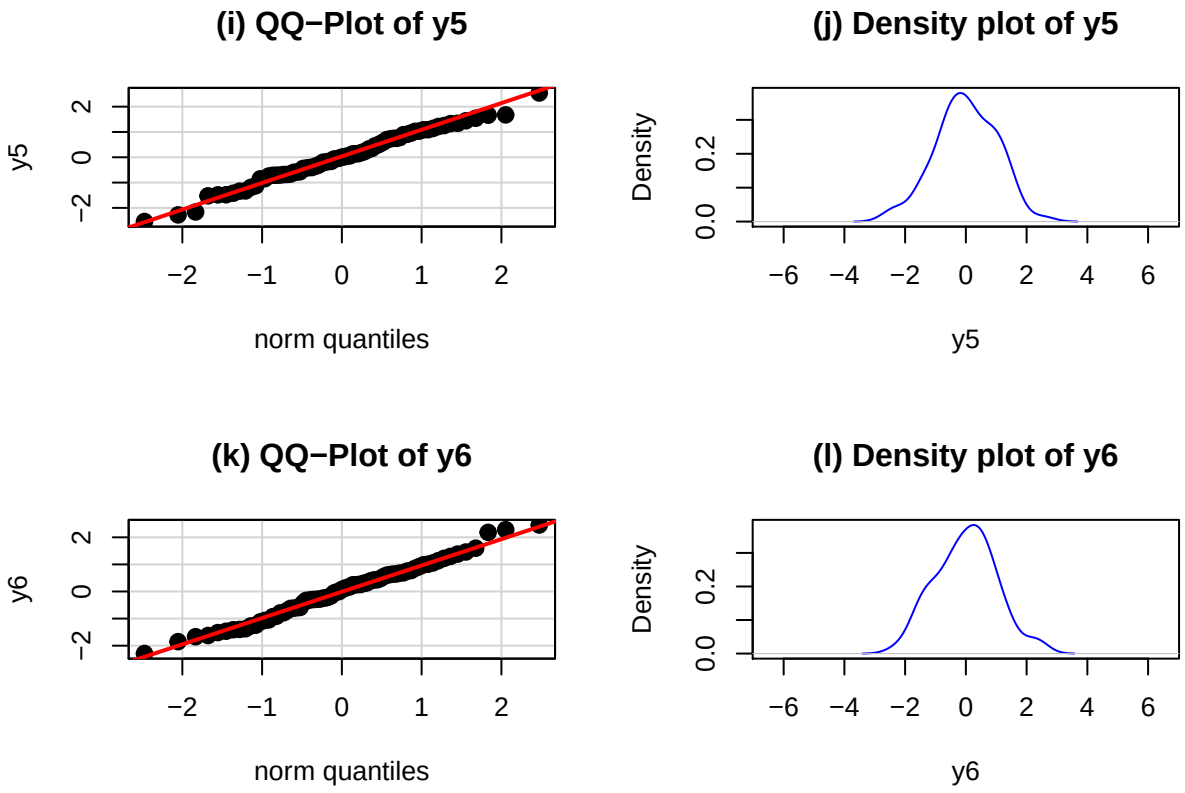


Figure 3.12: Two more simulated data sets, generated from normal distributions.

The last issues with assessing the assumptions in an ANOVA relates to situations where the methods are more or less ***resistant***[7] to violations of assumptions. For reasons beyond the scope of this book, the parametric ANOVA F-test is more resistant to violations of the assumptions of the normality and equal variance assumptions if the design is balanced. A ***balanced design*** occurs when each group is measured the same number of times. The resistance decreases as the data set becomes less balanced, as the sample sizes in the groups are more different, so having close to balance is preferred to a more imbalanced situation if there is a choice available. There is some intuition available here – it makes some sense that you would have better results in comparing groups if the information available is similar in all the groups and none are relatively under-represented. We can check the number of observations in each group to see if they are equal or similar using the `tally` function from the `mosaic` package. This function is useful for being able to get counts of observations, especially for cross-classifying observations on two variables that is used in Chapter 5. For just a single variable, we use `tally(~x, data=...)`:

```
require(mosaic)
tally(~Attr, data=MockJury)
```

```
## Attr
##    Beautiful      Average Unattractive
##           39           38           37
```

So the sample sizes do vary among the groups and the design is technically not balanced, but it is also very close to being balanced with only two more observations in the largest group compared to the smallest group size. This tells us that the F-test should have some resistance to violations of assumptions. This nearly balanced design, and the moderate sample size (over 37 per group is considered a good but not large sample), make the parametric and nonparametric approaches provide similar results in this data set even in the presence of the skewed residual error distribution.

3.4 ANOVA model diagnostics including QQ-plots

3.5 Guinea pig tooth growth One-Way ANOVA example

3.6 Multiple (pair-wise) comparisons using Tukey's HSD and the compact letter display

3.7 Pair-wise comparisons for Prisoner Rating data

3.8 Chapter Summary

3.9 Summary of important R code

3.10 Practice problems

[7] A resistant procedure is one that is not severely impacted by a particular violation of an assumption. For example, the median is resistant to the impact of an outlier.

Chapter 4

Two-Way ANOVA

Chapter 5

Chi-square tests

Chapter 6

Correlation and Simple Linear Regression

Chapter 7

Simple linear regression inference

Chapter 8

Multiple linear regression

Chapter 9

Case studies

9.1 Overview of material covered

9.2 The impact of simulated chronic nitrogen deposition on the biomass and N2-fixation activity of two boreal feather moss–cyanobacteria associations

9.3 Ants learn to rely on more informative attributes during decision-making

9.4 Multi-variate models are essential for understanding vertebrate diversification in deep time

9.5 General summary

Bibliography