A Second Semester Statistics Course with R

Mark Greenwood and Katherine Banner

2017-07-25

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

(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.

1.1 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:

Table 1.1: First 5 and last 6 rows of the Mock Jury 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 ***n*** for the number of subjects in the sample and denote the population size (if available) with ***N***. Here, the sample size is ***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.

[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.

Instead of loading this data set into R using the "Import Dataset" functionality, we can load an R package that contains the data, making for easy access to this data set. The package called `heplots` (Fox and Friendly, 2016) contains a data set called MockJury that contains the results of the study. We also rely the R package called `mosaic` (Pruim et al., 2016b) 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

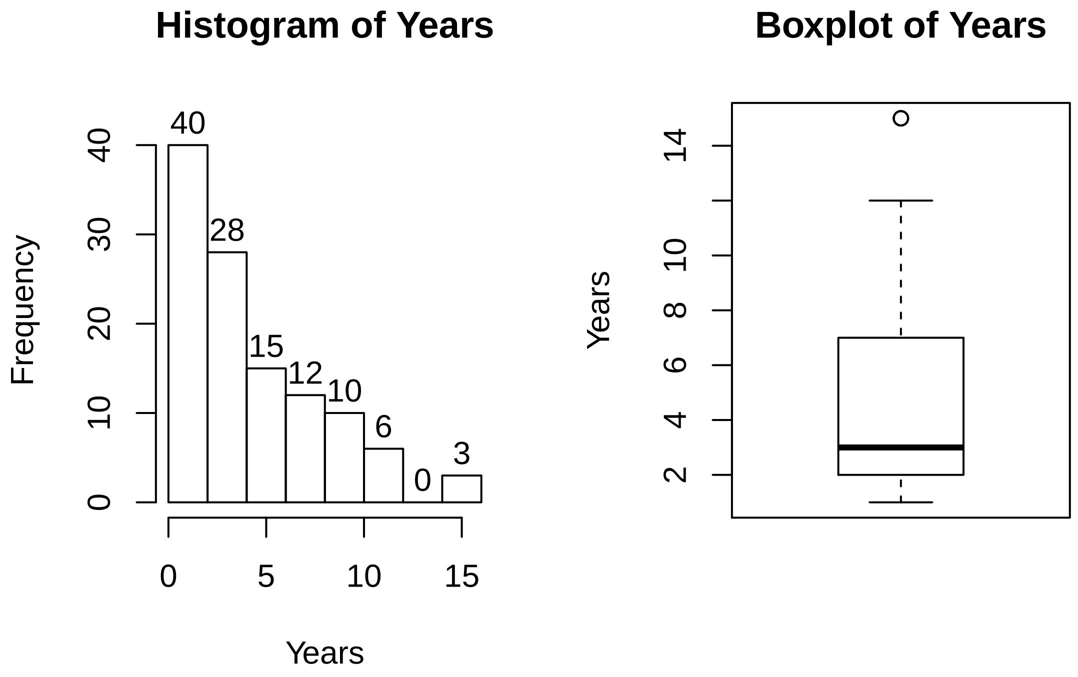

Figure 1.1: Histogram and boxplot of suggested sentences in years.

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 1.1 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.

```
par(mfrow=c(1,2))
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.

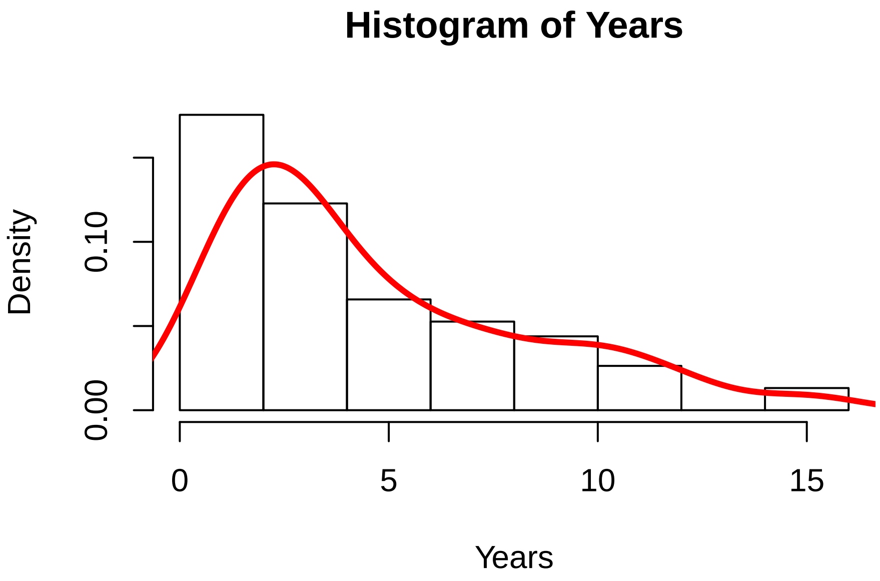

Figure 1.2: Histogram and density curve of Years data.

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 1.2 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.

```r
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 1.3, 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

[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.

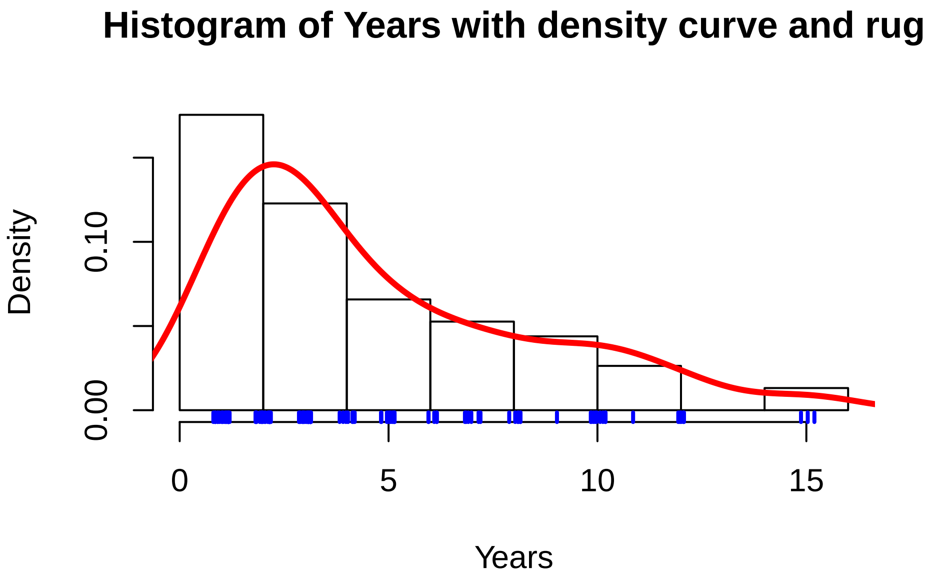

Figure 1.3: Histogram with density curve and rug plot of the jittered responses.

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 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 1.4 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

[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

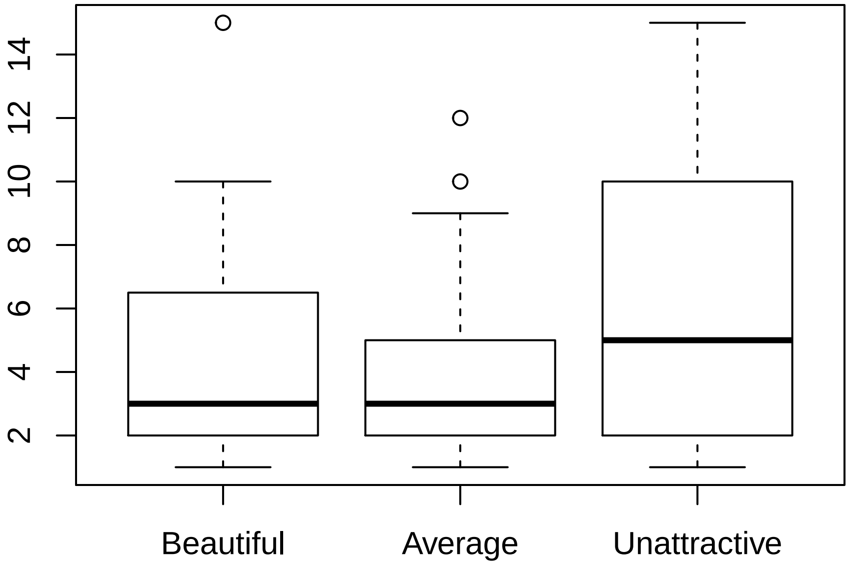

Figure 1.4: Side-by-side boxplot of Years based on picture groups.

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***.

1.2 Beanplots

The other graphical display for comparing multiple groups we will use is a newer display called a ***beanplot*** (Kampstra, 2008). Figure 1.5 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 (Kampstra, 2014). 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 1.5 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 *Unattractive* group. The *Unattractive* group seems to have more high observations than the other groups even

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

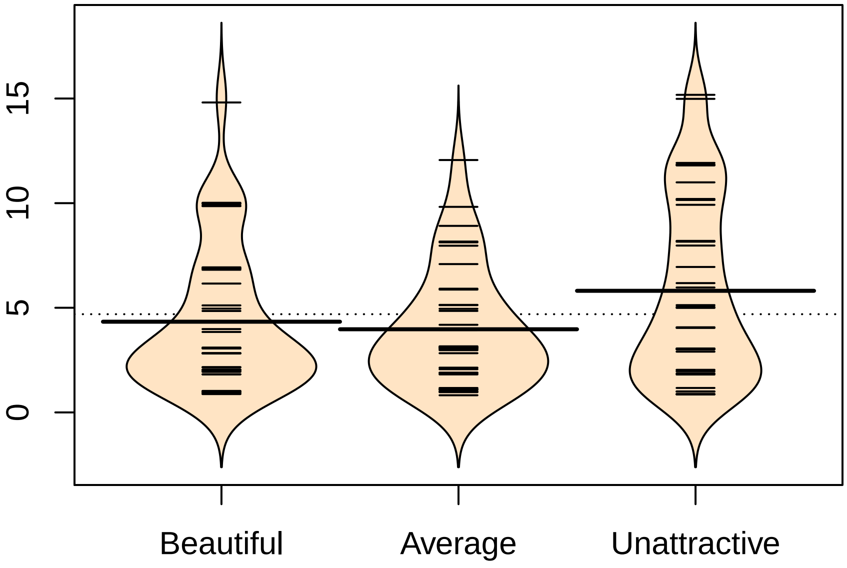

Figure 1.5: Beanplot of Years by picture group. Long, bold lines correspond to mean of each group.

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
```

```r
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:

```r
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):

```r
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:

```r
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.

```r
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`.

```r
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 1.6.

```r
par(mfrow=c(1,2))
boxplot(Years ~ Attr,data=MockJury2)
beanplot(Years ~ Attr,data=MockJury2,log="",col="bisque",method="jitter")
```

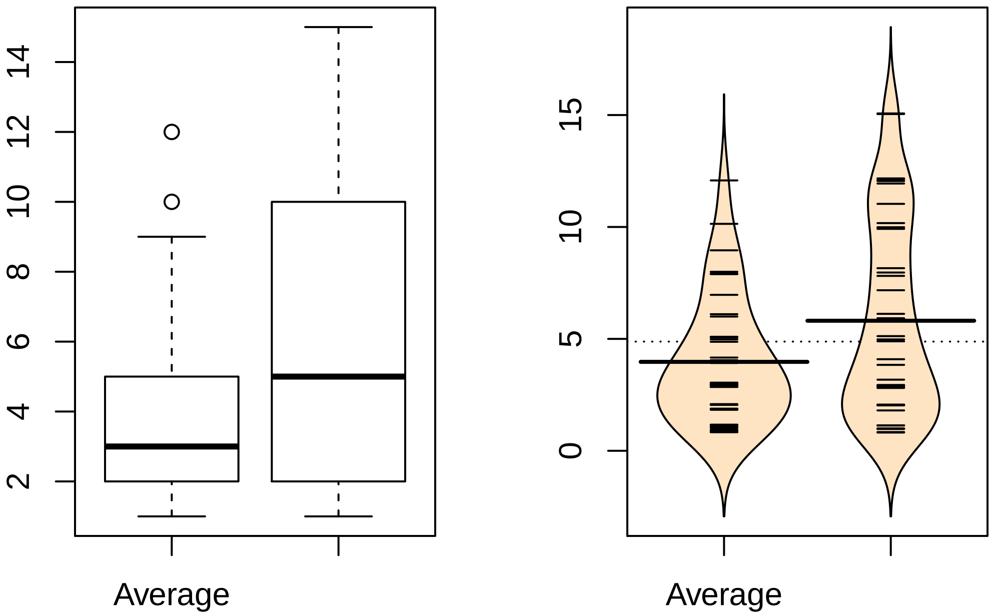

Figure 1.6: Boxplot and beanplot of the Years responses on the reduced data set.

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
```

1.3 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 $y_{ij} = \mu_j + \varepsilon_{ij}$, where we assume that $\varepsilon_{ij} \sim N(0, \sigma^2)$. For the moment, focus on the models that either assume the means are the same (null) or different (alternative), which imply:

- Null Model: $y_{ij} = \mu + \varepsilon_{ij}$ There is **no** difference in **true** means for the two groups.
- Alternative Model: $y_{ij} = \mu_j + \varepsilon_{ij}$ 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 $y_{42} = \mu_2 + \varepsilon_{42}$, 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 $y_{51} = \mu_1 + \varepsilon_{51}$. 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 1.7 – 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 1.7 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.

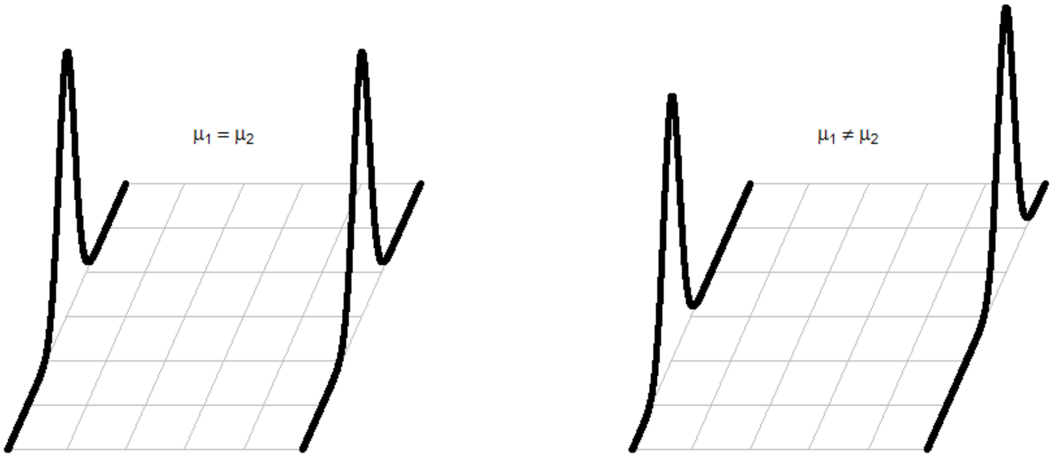

Figure 1.7: Illustration of the assumed situations under the null (left) and a single possibility that could occur if the alternative were true (right) and the true means were different.

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

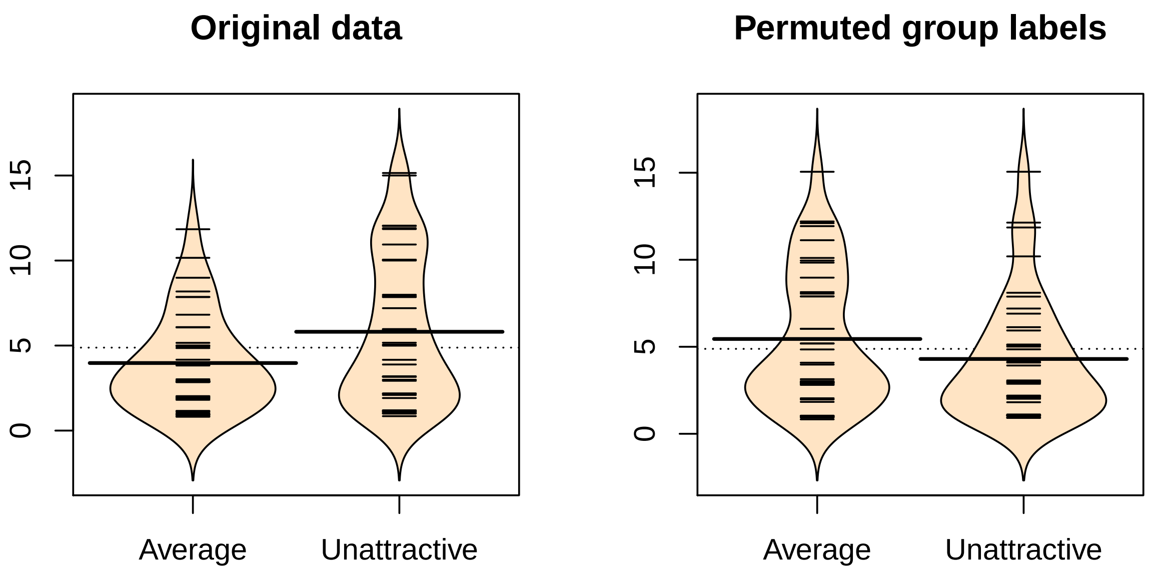

Figure 1.8: Boxplots of Years responses versus actual treatment groups and permuted groups.

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 1.8 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 1.8). 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.

1.4 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:

[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.

```
> 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
[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)
```

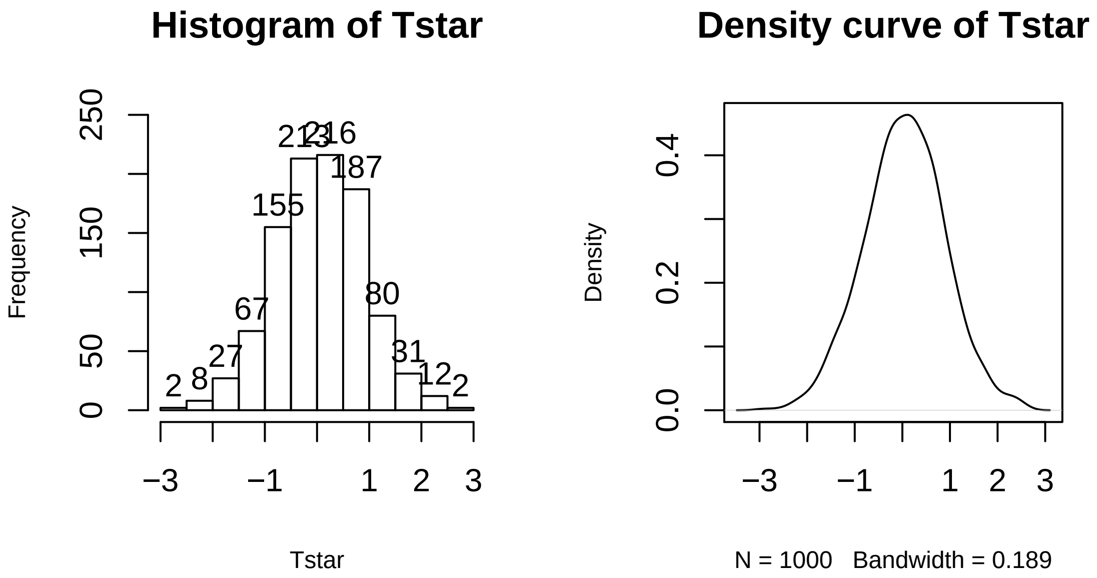

Figure 1.9: Histogram (left, with counts in bars) and density curve (right) of values of test statistic for 1,000 permutations.

```
}
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:

```
par(mfrow=c(1,2))
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")
```

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

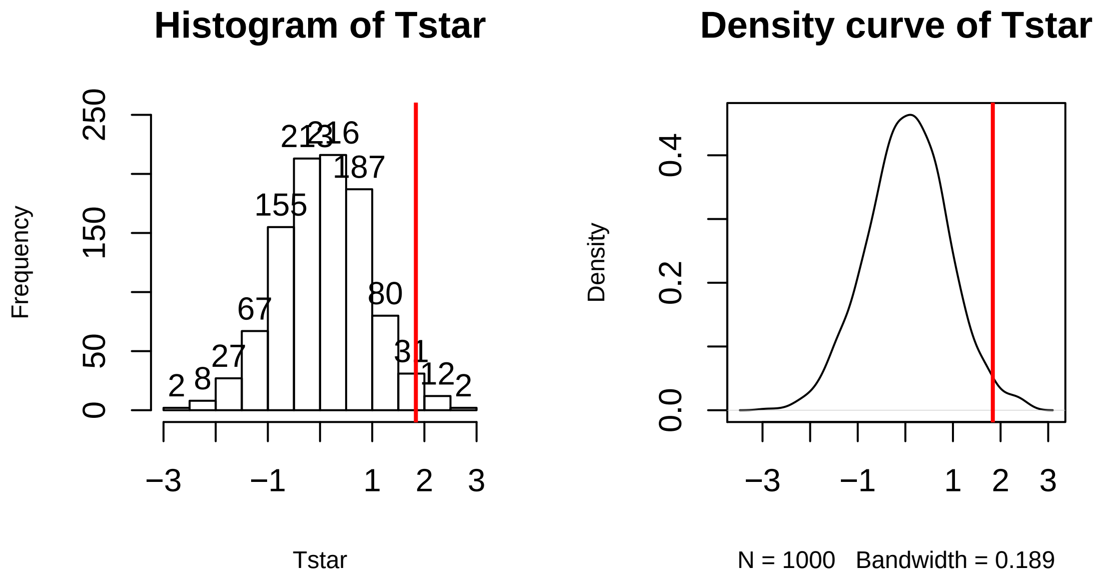

Figure 1.10: Histogram (left) and density curve (right) of values of test statistic for 1,000 permutations with bold vertical line for value of observed test statistic.

```
favstats(Tstar)
```

```
##        min         Q1     median        Q3      max       mean        sd
##  -2.910384 -0.5099573 0.07681366 0.6102418 2.530583 0.04694168 0.8497364
##     n missing
##  1000       0
```

Figure 1.9 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 1.10, using the `abline` function.

```
par(mfrow=c(1,2))
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.

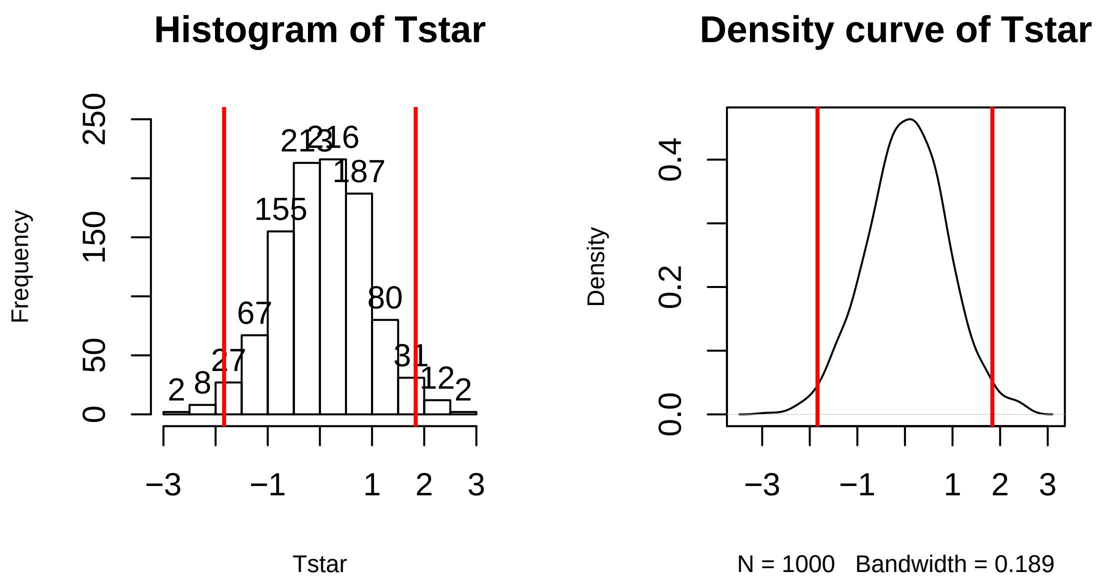

Figure 1.11: Histogram and density curve of values of test statistic for 1,000 permutations with bold lines for value of observed test statistic and its opposite value required for performing two-sided test.

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*** ($\mathbf{H_A}: \boldsymbol{\mu_1 \neq \mu_2}$). 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:

```r
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 1.11 shows both cut-offs on the histogram and density curve.

[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.

```
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")
```

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 1.11.

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 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,

[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.

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.

1.5 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.

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 1.2 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.

Table 1.2: Scope of inference summary.

Random Sampling/Random Assignment	**Random Assignment (RA) – Yes (controlled experiment)**	**Random Assignment (RA) – No (observational study)**
Random Sampling (RS) – Yes (or some method that results in a representative sample of population of interest)	Because we have RS, we can generalize inferences to the population the RS was taken from. Because we have RA we can assume the groups were equivalent on all aspects except for the treatment and can establish causal inference.	Can generalize inference to population the RS was taken from but cannot establish causal inference (no RA – cannot isolate treatment variable as only difference among groups, could be confounding variables).
Random Sampling (RS) – No (usually a convenience sample)	Cannot generalize inference to the population of interest because the sample was not random and could be biased – may not be "representative" of the population of interest. Can establish causal inference due to RA → the inference from this type of study applies only to the sample.	Cannot generalize inference to the population of interest because the sample was not random and could be biased – may not be "representative" of the population of interest. Cannot establish causal inference due to lack of RA of the treatment.

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 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 1.3 to help you remember all the possibilities.

Table 1.3: Table of decisions and truth scenarios in a hypothesis testing situation. But we never know the truth in a real situation.

	H_0 True	**H_0 False**
FTR H_0	Correct decision	Type II error
Reject H_0	Type I error	Correct decision

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.

1.6 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 1.4 with the new t-statistic replacing the difference in the sample means that we used before.

The permutation distribution in Figure 1.12 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.

```
par(mfrow=c(1,2))
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
}
pdata(abs(Tstar),abs(Tobs),lower.tail=F)
```

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

```
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")
```

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.

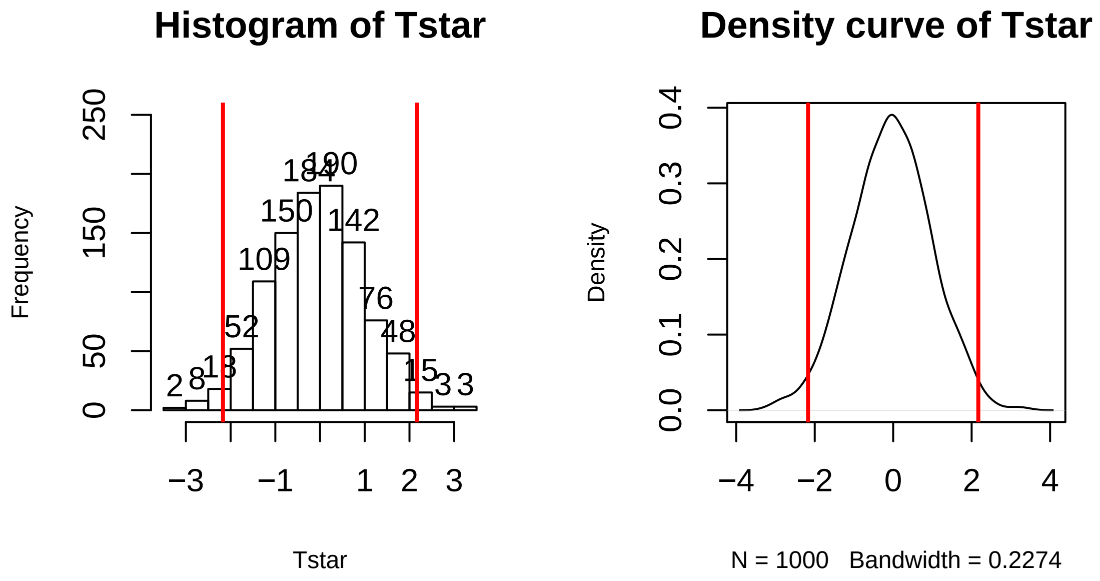

Figure 1.12: Permutation distribution of the t-statistic

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 1.13. 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.

```
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 1.14 shows how similar the

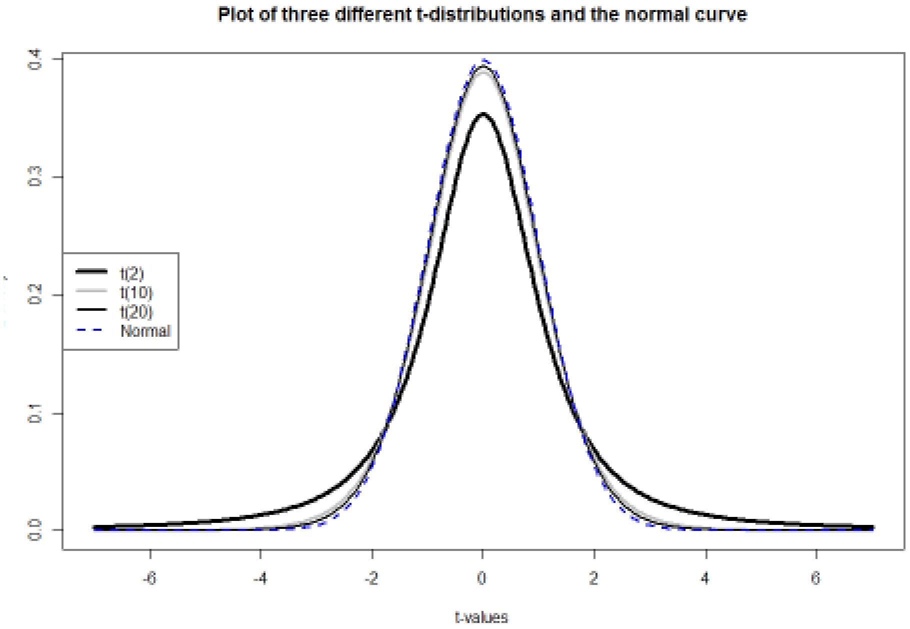

Figure 1.13: Plots of t and normal distributions

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:

[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.

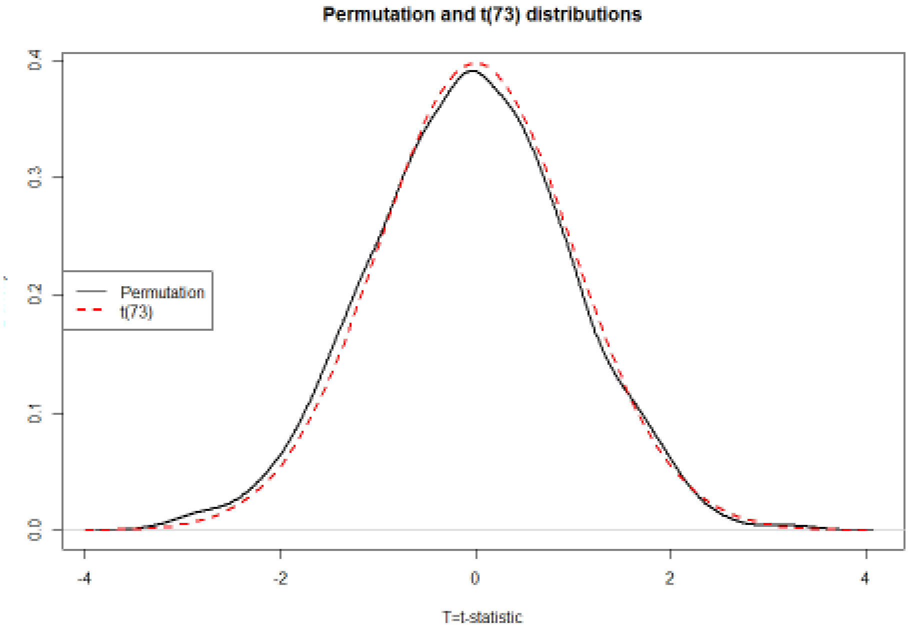

Figure 1.14: Plot of permutation and t distribution with $df = 73$.

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]

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 1.6 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

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

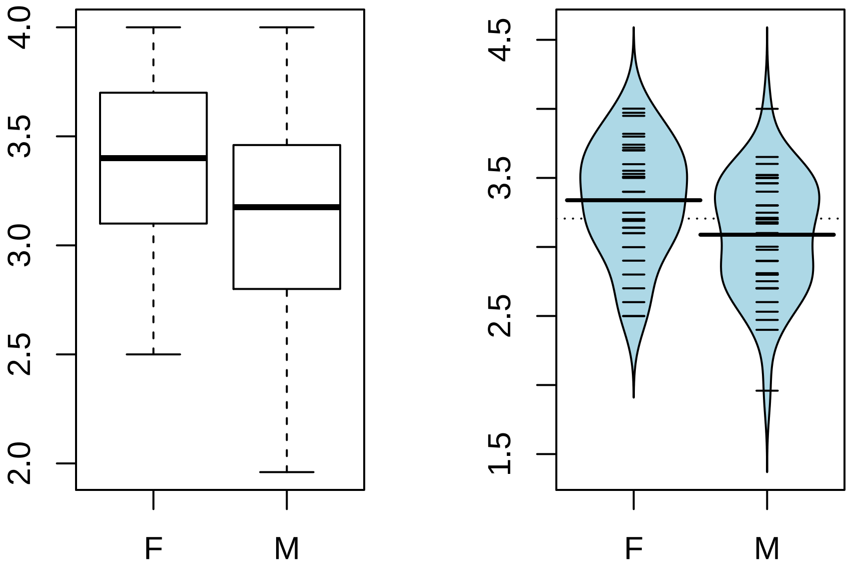

Figure 1.15: Side-by-side boxplot and beanplot of GPAs of STAT 217 students by sex.

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.

1.7 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 1.15.

```
s217 <- read.csv("http://www.math.montana.edu/courses/s217/documents/s217.csv")
require(mosaic)
require(beanplot)
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
## 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
```

```
par(mfrow=c(1,2))
boxplot(GPA~Sex, data=s217)
beanplot(GPA~Sex, data=s217, log="", col="lightblue", method="jitter")
```

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

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

    - $H_0 : \mu_{male} = \mu_{female}$

        – where μ_{male} is the true mean GPA for males and μ_{female} is true mean GPA for females.

    - $H_A : \mu_{male} \neq \mu_{female}$

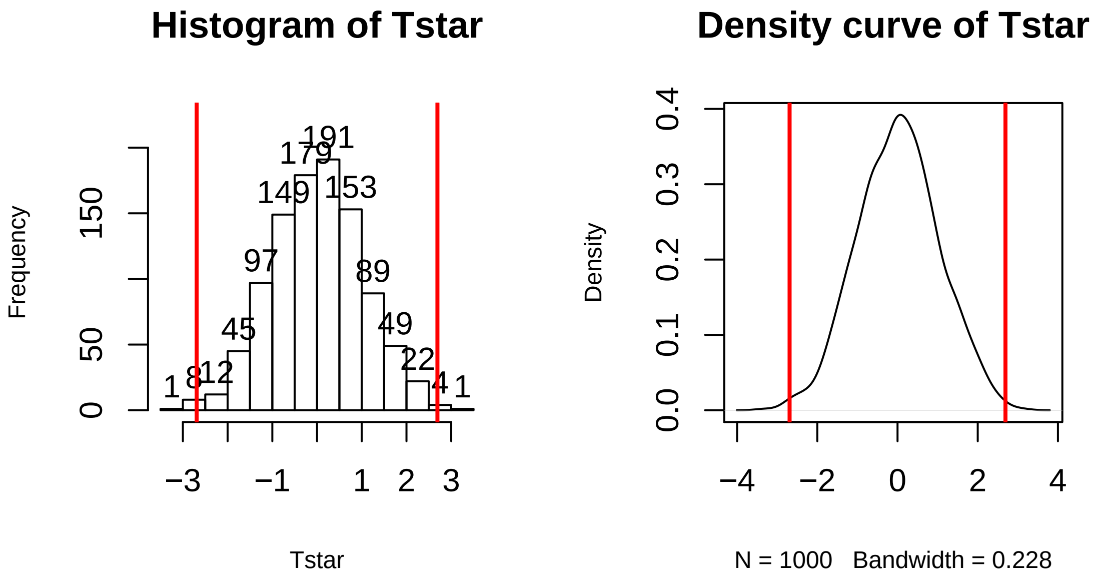

Figure 1.16: Histogram and density curve of permutation distribution of test statistic for STAT 217 GPAs.

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.

1.8 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.

[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.

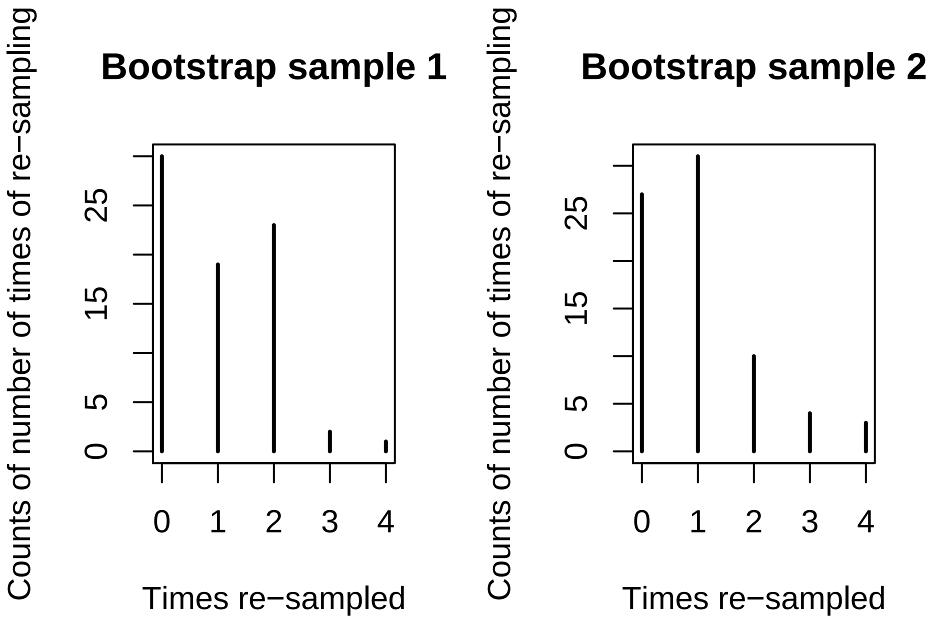

Figure 1.17: Counts of number of times of observation (or not observed for times re-sampled of 0) for two bootstrap samples.

```
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 1.17. 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.

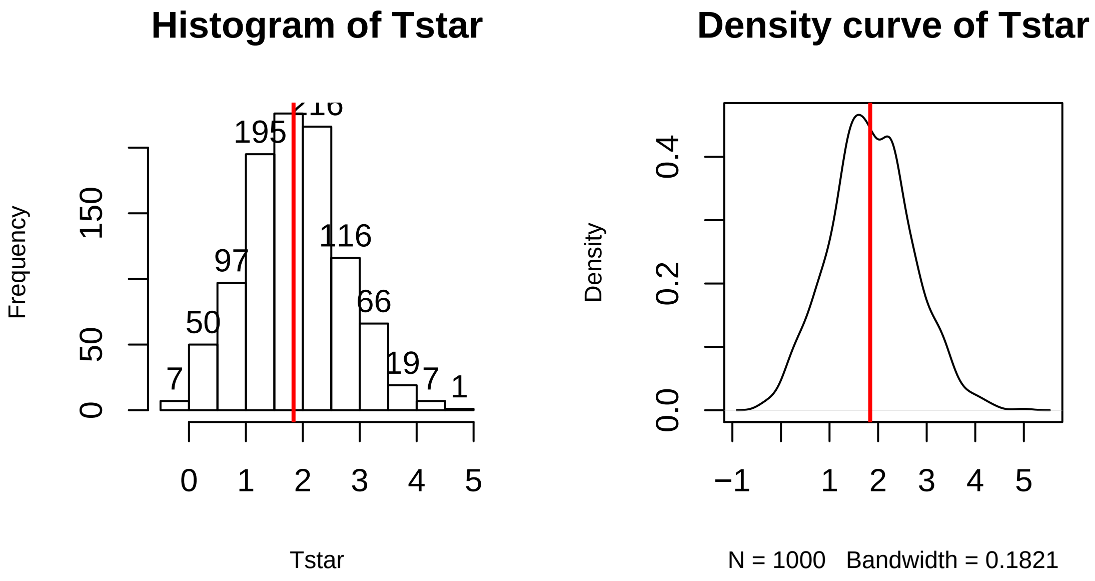

Figure 1.18: Histogram and density curve of bootstrap distributions of difference in sample mean `Years` with vertical line for the observed difference in the means of 1.84 years.

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
```

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

```
B <- 1000
Tstar <- matrix(NA,nrow=B)
for (b in (1:B)){
  Tstar[b] <- diffmean(Years ~ Attr, data=resample(MockJury2))
  }
favstats(Tstar)
```

```
##         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
```

[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.

```
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)
```

In this situation, the observed difference in the mean sentences is 1.84 years (Unattractive-Average), which is the vertical line in Figure 1.18. 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 1.12) 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
```

```
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 1.19 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

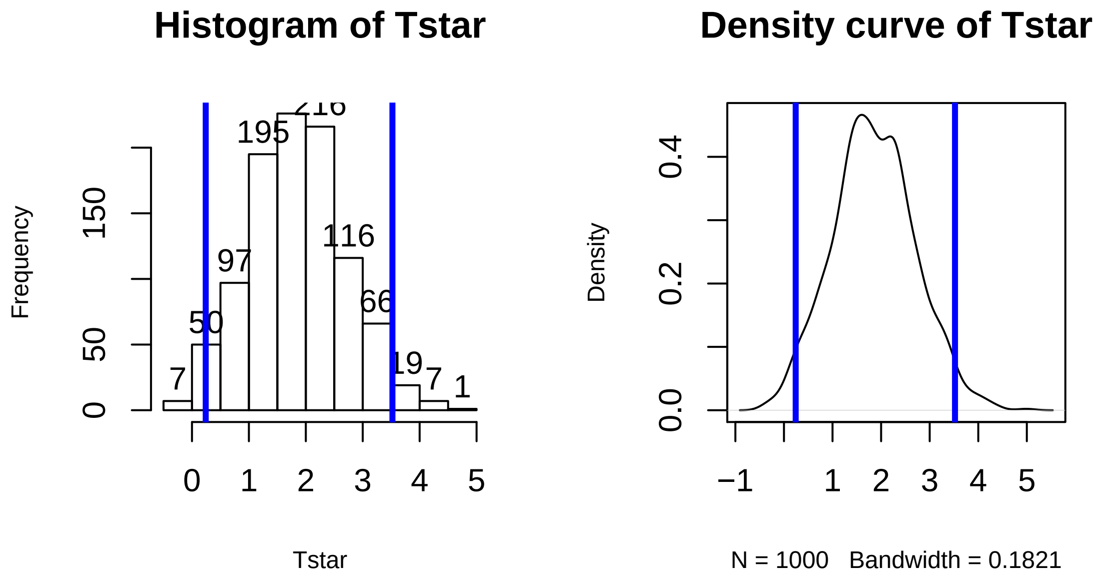

Figure 1.19: Histogram and density curve of bootstrap distribution with 95% bootstrap confidence intervals displayed (vertical lines).

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

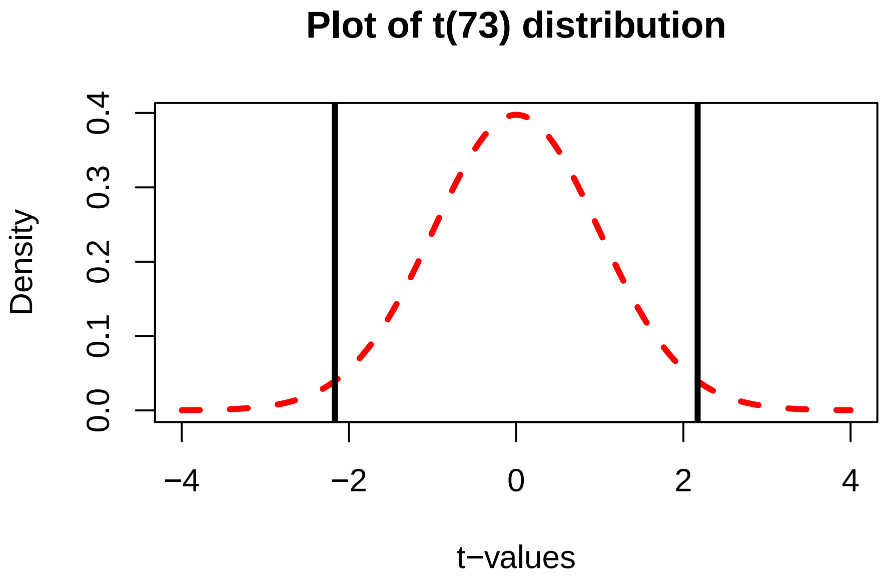

Figure 1.20: Plot of $t(73)$ with cut-offs for putting 95% of distributions in the middle.

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 1.20 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)
```

[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

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

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

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 repeated below 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
```

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
```

1.9 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
```

[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.

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

```
##
##  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 1.21.

```
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
```

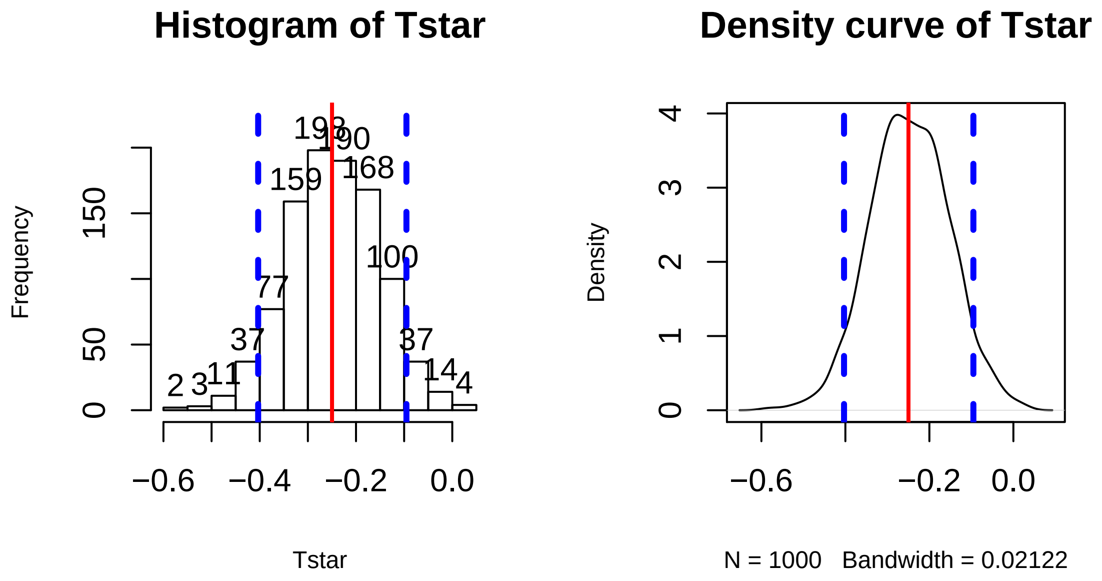

Figure 1.21: Histogram and density curve of bootstrap distribution of difference in sample mean GPAs (male minus female) with observed difference (solid vertical line) and quantiles that delineate the 90% confidence intervals (dashed vertical lines).

```
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 1.21 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.

1.10 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 **??**, we'll learn some new tools for doing diagnostics to help us assess how much those conditions are violated.

1.11 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)
    - Requires the `beanplot` package is loaded.
    - Makes a beanplot of a variable named Y for groups in X from the data set.

- mean(Y~X, data=DATASETNAME); sd(Y~X, data=DATASETNAME)
  - This usage of `mean` and `sd` requires the `mosaic` package.
  - Provides the mean and sd of responses of Y for each group described in X.
- 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`)

1.12 Practice problems

Load the `HELPrct` data set from the `mosaicData` package (Pruim et al., 2016a) (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 2

Bibliography

Fox, J. and Friendly, M. (2016). *heplots: Visualizing Hypothesis Tests in Multivariate Linear Models*. R package version 1.3-3.

Kampstra, P. (2008). Beanplot: A boxplot alternative for visual comparison of distributions. *Journal of Statistical Software, Code Snippets*, 28(1):1–9.

Kampstra, P. (2014). *beanplot: Visualization via Beanplots (like Boxplot/Stripchart/Violin Plot)*. R package version 1.2.

Plaster, M. E. (1989). Inmates as mock jurors: The effects of physical attractiveness upon juridic decisions. Master's thesis, East Carolina University, Greenville, NC. MockJury data taken from the thesis.

Pruim, R., Kaplan, D., and Horton, N. (2016a). *mosaicData: Project MOSAIC Data Sets*. R package version 0.14.0.

Pruim, R., Kaplan, D. T., and Horton, N. J. (2016b). *mosaic: Project MOSAIC Statistics and Mathematics Teaching Utilities*. R package version 0.14.4.