LAB 1: INTRODUCTION TO R

Statistical analysis is the core of nearly all research projects, and researchers have a wide variety of statistical tools that they can use, like SPSS and SAS. Unfortunately, these analysis tools are expensive or difficult to master, so this lab manual introduces R, a powerful, free statistical analysis program. However, before diving into a statistics package, one necessary background fundamental must be covered: data.

1 Types of Data

There are two main data types, and it is essential to understand the difference between them since that determines appropriate analytical tests.

- **Continuous Data** is numeric and is typically used for counts or measures – like a person's weight, a tree's height, or a car's speed. Continuous data is measured with a scale that uses equal divisions to calculate the difference between any two values. Continuous data is analyzed using values like means or tests like ANOVA, covered in a later lab.
- **Categorical Data** is a group of observations, like a type of pet (cat, dog, bird) or state of residence (Arizona, California). One common categorical data type is generated with a Likert scale: "I enjoy reading: Strongly Agree -- Agree -- Neutral -- Disagree -- Strongly Disagree." Categorical data is analyzed using values like mode or tests the Mann-Whitney U, covered in a later lab.

2 The R Command Line

All R commands are entered from a "Command Line" environment. Many students find this challenging, but the command line becomes easy and fast once they learn some foundational concepts.

2.1 The R Command Line

All exercises in this course are completed at a free online site: https://rdrr.io/snippets/. When the site is first accessed, the Snippets box contains a sample code, which should be deleted so the code from the lab manual can be inserted. Following is a screen grab of the initial Snippets input box.

To enter the R code from the lab manual, select the sample text in the Snippets box and tap the delete key. Then, copy and paste the R code in the right-hand box below into the Snippets text box and tap the "Run" button.

Here is an example of several R commands. Each line is explained below the command listing box.

```
1  # Some basic calculations
2  3 + 5
3  5 + 8 * 2
4  # Using variables
5  MaxScore <- 57
6  MinScore <- 22
7  Range <- MaxScore - MinScore
8  Range
```

Line 1: This is a comment used to record notes in a script. All comments start with a hash-mark (#) in R, and everything after that symbol is ignored. Comments are used frequently in scripts presented in this manual to explain what the script is doing. Good programmers comment liberally so team members can quickly figure out what they did.

Line 2: Calculate the value of 3 + 5.

Line 3: Calculate the value of 5 + 8 * 2.

Line 4: This is a comment line.

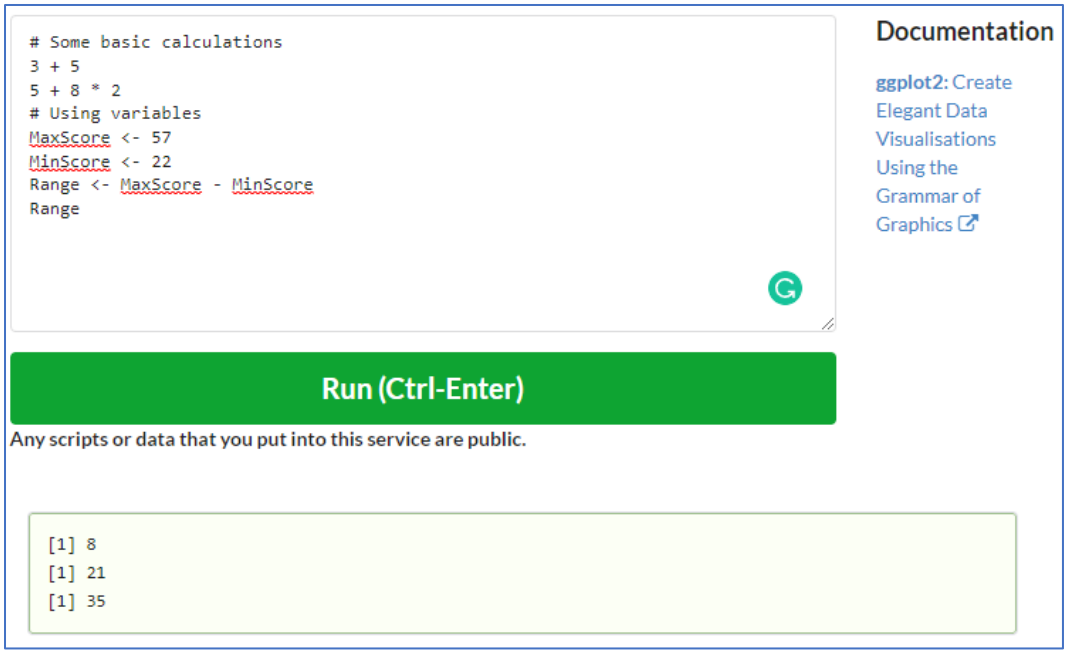
Lines 5-6: These lines create two variables, *MaxScore* and *MinScore*, and then assign values to the variables. You should note two important things about these lines. First, the "assignment" operator is a less-than sign followed by a hyphen, making a left-pointing arrow, like <-. That tells R to store the number on the right side of the arrow operator into the variable named on the left side of the line. Also, remember that capitalization matters with R. Thus, the variable named *MaxScore* would be different from a variable named *maxscore*. These lines only store values in variables, and nothing gets printed on the screen.

A variable is nothing more than a place in memory to store temporary data. Think of it as a "box" used to store something until it is needed later.

Line 7: The variable *Range* is filled with the result of subtracting *MaxScore* minus *MinScore*.

Line 8: Entering a variable name, like *Range*, on a line by itself causes the value stored in that variable to be displayed.

The following is when the above R code lines are inserted in the Snippets box and executed.



Notice that the result shows only the answers, not the initial problems. So, the first answer, 8, is for the problem 3 + 5. It will not take long to learn how to link the solution's lines to the questions asked.

2.2 Activity: The R Command Line

For this lab, calculate the following problems.

1	5 + 8 * 3
2	10 * 2 / 4
3	5 * 3 + 3 - 20

Include the answers to these three problems in the deliverable document for this lab.

3 Data Frames

A data frame is a collection of data generated during a research project. An example data frame that is easy to understand would be a spreadsheet that contains the times recorded for a race. R comes configured with 103 built-in data frames used for training, and the R script below is an introduction to one of the data frames used in several of the labs in this manual: *stackloss*. This data frame is Brownlee's Stack-Loss Plant Data, 21 observations on stack-loss (the loss of acid through the stack) in a chemical plant to convert ammonia to nitric acid. There are three explanatory variables: airflow, cooling water inlet temperature, and acid concentration.

3.1 Demonstration: Data Frames

The following R code is an example of data frame operations.

```
1  # Exploring the Stack Loss Data Frame
2  stackloss
3  str(stackloss)
4  MaxAirFlow <- max(stackloss$Air.Flow)
5  MaxAirFlow
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: Entering the data frame's name, *stackloss*, on a line by itself causes R to print the contents of the entire data frame to the screen. Since *stackloss* is relatively small, it is acceptable to print it to the screen, but some data frames have hundreds of lines, which may cause the screen to "scroll" for some time before the end of the data frame is reached.

Line 3: This prints the structure of the *stackloss* data frame. The result shows a data frame with 21 observations (that is, how many lines are in the data frame) of 4 variables (things like *Air.Flow*). Also, the structure command displays the type of data in the data frame. For example, all four variables are of the "number" type. The *str* function is frequently used to understand a data frame.

Line 4: This line puts the maximum *Air.Flow* value for the *stackloss* data frame into a variable named *MaxAirFlow*. Note that the specific variable in the data frame is indicated by both the data frame name and variable name separated by a dollar sign, like *stackloss$Air.Flow* on this line.

Line 5: Entering the name of a variable on a line by itself displays the value of that variable, so this line displays the value of *MaxAirFlow*.

The five R code lines in the code box above should be pasted into the Snippets input box and executed to generate the following result.

```
   Air.Flow Water.Temp Acid.Conc. stack.loss
1        80         27         89         42
2        80         27         88         37
3        75         25         90         37
4        62         24         87         28
5        62         22         87         18
6        62         23         87         18
7        62         24         93         19
8        62         24         93         20
9        58         23         87         15
10       58         18         80         14
11       58         18         89         14
12       58         17         88         13
13       58         18         82         11
14       58         19         93         12
15       50         18         89          8
16       50         18         86          7
17       50         19         72          8
18       50         19         79          8
19       50         20         80          9
20       56         20         82         15
21       70         20         91         15
'data.frame':    21 obs. of  4 variables:
 $ Air.Flow  : num  80 80 75 62 62 62 62 62 58 58 ...
```

```
 $ Water.Temp: num  27 27 25 24 22 23 24 24 23 18 ...
 $ Acid.Conc.: num  89 88 90 87 87 87 93 93 87 80 ...
 $ stack.loss: num  42 37 37 28 18 18 19 20 15 14 ...
[1] 80
```

3.2 Activity: Data Frames

Using the *stackloss* data, find the maximum values for *Water.Temp* and *Acid.Conc* and include those values in the deliverable document for this lab.

4 Deliverable

Complete the activities in this lab and consolidate the responses into a single document. Name the document with your name and "Lab 1," like "George Self Lab 1," and submit that document for grading.

Lab 2: Central Measures

1 Introduction

It is often desirable to describe an entire group of numbers as a single value, and the number "in the middle" of the group would seem most logical to use. Students in elementary school are taught how to find the average of a group of numbers and learn that the average is the best representation for that entire group. In statistics, several different numbers are often used to represent an entire group of numbers. These are collectively known as the *Central Measures* or numbers that are the "middle" of the group.

2 N

One of the most straightforward measures is nothing more than the number of items in a group. For example, the *N*, or number of items in the group 5, 7, 13, 22 is 4. *N* is not a central measure, but it is commonly used and is included here for completeness.

2.1 Demonstration: N

Use R's *length* procedure to find the number of items in a variable (N). Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Finding N
2  length(faithful$eruptions)
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored. (Fun fact, the official name for that symbol is "octothorpe.")

Line 2: The length of the eruptions variable in the *faithful* data frame[1] is printed, the same as the number of items in the variable, or N. Notice how a variable is identified using the data frame name, then a dollar sign, then the variable name.

R reports the following N.

```
[1] 272
```

2.2 Activity: N

Using the *attitude*[2] data frame, find the number of items in the *rating* variable and include that number in the deliverable document for this lab.

3 Mean

The mean is calculated by adding all the data items and dividing that sum by the number of items (N). This process is taught in elementary school as the *average*. For example, given the group: 6, 8, and 9, the total is 23, divided by 3 (N), is 7.66; so, the mean of 6, 8, and 9 is 7.66.

3.1 Demonstration: Mean

Use R's *mean* procedure to calculate the mean of a variable. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Find means for faithful
2  mean(faithful$eruptions)
3  mean(faithful$waiting)
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Lines 2-3: Calculate the means for eruptions and waiting in the *faithful* data frame.

R reports the following means.

```
[1] 3.487783
[1] 70.89706
```

3.2 Activity: Mean

Using the *attitude* data frame, find the mean of the *critical* variable and include that number in the deliverable document for this lab.

4 Trimmed Mean

If a group has a few unusually large or small values, then the mean is often skewed to no longer represent the "average" value. As an example, the weight, in grams, of chicks being fed different diets over 21 days ranges from 35 to 373. These weights are considerably spread with several unusually high weights (some chicks seemed to fare much better on their diets than others). The mean weight, 121.82g, is not representative of the entire group because those few chubby chicks tended to skew the mean upward. One way to compensate for unusually high or low values in a group is to use a trimmed mean (sometimes called a truncated mean).

A trimmed mean is calculated by removing a specific number of values from both the top and bottom of the group and then finding the mean of the remaining values. In the case of the chick weights, the following table shows the chick weight means with various amounts trimmed.

Trim	Mean
0.00	121.82g
0.05	116.48g
0.10	113.18g
0.15	110.83g

The untrimmed mean is 121.82g, but the unusually high weights are removed as values are trimmed. Thus, the trimmed mean better represents the entire group of chick weights. It is not possible or desirable to trim a different amount from the top and bottom; a trimmed mean always trims the same amount from both ends. A trimmed mean is not commonly used in actual practice since it is difficult to know how much to trim, and the resulting mean may be just as skewed as if no values were trimmed. Thus, the best "middle" term to report is the median when unusual values are suspected.

4.1 Demonstration: Trimmed Mean

Use R's mean procedure with a "trim" parameter to calculate the trimmed mean of a variable. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Find trimmed means for attenu
2  mean(attenu$accel)
3  mean(attenu$accel, trim=0.15)
4  mean(attenu$accel, trim=0.25)
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Lines 2-4: calculate the trimmed mean by including "trim=…" in the mean function. Line 2 is the mean for the accel group in the *attenu* data frame[3], Line 3 trims 15% from each end of the group, and Line 4 trims 25%.

R reports the following trimmed means.

```
[1] 0.1542198
[1] 0.1243906
[1] 0.1166196
```

4.2 Activity: Trimmed Mean

Using the *attenu* data frame, find the mean of the *accel* variable with 20% trimmed and include that number in the deliverable document for this lab.

5 Median

The median is found by listing all the data items in numeric order and then finding the middle item mechanically. For example, using 6, 8, and 9, the middle item (or median) is 8. The median for groups with an even number of items is the mean between the two middle items. For example, in 6, 8, 9, and 13, the median is 8.5, and the mean of the two middle terms, 8 and 9.

The median is useful in cases where the group has unusually small or large values. As an example of using a median rather than a mean, consider 5, 6, 7, 8, and 30. The mean is (5+6+7+8+30)/5 = 56/5 = 11.2. However, 11.2 is much higher than most of the values in that group since one unusually high value, 30, is significantly skewing the mean upward. A much better representation of the center of this group is the median, 7.

As another example where the median is the best central measure, suppose a newspaper reporter wanted to find the "average" wage for a group of factory workers. The ten workers in that factory all have an annual salary of $25,000; however, the supervisor has a salary of $125,000. In the newspaper article, the supervisor says that his workers have an average salary of $34,090. That is correct if the mean of all those salaries (including the supervisor) is reported. However, that number is higher than any reasonable "average" salary for workers in the factory due to the large salary. In this case, the median of $25,000 would be much more representative of the "average" salary. The median is typically reported for salaries, home values, and other groups where one or two unusually small or large values distort the reported "middle" value. The mean and median are the same if the group contains no unusually small or large values. However, if those extreme values exist, these two measures become separated, often by a significant amount.

5.1 Demonstration: Median

Use R's median procedure to calculate the median of a variable. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Find medians for faithful
2  median(faithful$eruptions)
3  median(faithful$waiting)
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Lines 2-3: Calculate the medians for *eruptions* and *waiting* in the *faithful* data frame.

R reports the following medians.

```
[1] 4
[1] 76
```

5.2 Activity: Median

Using the *attitude* data frame, find the median of the *critical* variable and include that number in the deliverable document for this lab.

6 Mode

The mode describes the center of categorical data (lists of items) and is only the item that appears most often in the group. For example, if a question asked respondents to select their zip code from a list of five local codes and "12345" was selected more often than any other, that would be the mode for zip code. Calculating the mode is no more difficult than counting the times the various values are found and choosing the most frequent one.

There is no procedure for finding mode in this lab, but the tutorial on Frequency Tables in Lab Five shows an easy way to determine the mode of a variable.

7 Deliverable

Complete the four activities in this lab and consolidate the responses into a single document. Name the document with your name and "Lab 2," like "George Self Lab 2," and submit that document for grading.

---

[1] The faithful data frame contains observations about the Old Faithful Geyser in Yellowstone National Park. The data frame includes these variables: faithful$waiting, the waiting time between eruptions, and faithful$eruptions, the eruption time in minutes.

[2] The attitude data frame contains information from a survey of the clerical employees of a large financial organization. The data frame includes these variables: attitude$rating, attitude$complaints, attitude$privileges, attitude$learning, attitude$raises, attitude$critical, and attitude$advance.

[3] The attenu data frame gives peak accelerations measured at various observation stations for 23 earthquakes I in California. The data is used to estimate the attenuating effect of distance on ground acceleration. The data frame includes these variables: attenu$event, attenu$mag, attenu$station, attenu$dist, and attenu$accel.

Lab 3: Dispersion

1 Introduction

One way to describe a variable is to report its dispersion or spread. Image, a poll of 100 potential Chevrolet customers were asked which model they preferred. A local dealer would schedule different sales events if the responses were evenly distributed among Impala, Camaro, and Traverse than if 90% of the customers preferred Impala. How tightly survey results are grouped (or scattered) is called "dispersion." This lab explores data dispersion and the methods used to calculate that value with R.

2 Range

The maximum and minimum values are those at the extreme ends of a variable, and the range is nothing more than the maximum minus the minimum values. For the 2016 version of the Scholastic Aptitude Test (SAT), the maximum score is 1600, and the minimum score is 400, so the range is 1600 - 400, or 1200. Even though the range is sometimes expressed as a single number, as in the case of the SAT scores, researchers typically want to know the actual endpoints rather than just the spread, and those endpoints are what R reports for range.

2.1 Demonstration: Range

To find the range of a variable, use R's range function. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Range for trees
2  range(trees$Girth)
3  range(trees$Height)
4  range(trees$Volume)
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Lines 2-4: Display the ranges for three variables in the *trees* data frame[1].

R reports the following ranges.

```
[1] 8.3 20.6
[1] 63 87
[1] 10.2 77.0
```

2.2 Activity: Range

Using the *faithful*[2] data frame, find the range for *eruptions* and include that number in the deliverable document for this lab.

3 Quartiles

A measure closely related to the median is the first and third quartile. The first quartile (Q1) is the score that splits the lowest 25% from the rest, and the third quartile (Q3) splits the highest 25% from the rest. The second quartile (Q2) is the same as the median; normally, the term *median* is used rather than Q2. For example, consider this group: 5, 7, 10, 13, 17, 19, 23. The median of this variable is 13 because it is in the middle. The first quartile is 7, the score that splits the lowest 25% from the rest of the data. The third quartile is 19, the score that splits the highest 25% from the rest of the data.

3.1 Demonstration: Quartiles

To find the quartiles of a variable, use R's *summary* function. R calculates the quartiles and several other descriptive measures and is a staple in a data scientist's toolbox. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Summary for trees
2  summary(trees$Girth)
3  summary(trees$Height)
4  summary(trees$Volume)
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Lines 2-4: Display summary information for three variables in the *trees* data frame.

R reports the following summaries.

```
   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
   8.30   11.05   12.90   13.25   15.25   20.60
   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
   63      72      76      76      80      87
   Min. 1st Qu. Median   Mean 3rd Qu.   Max.
   10.20  19.40   24.20   30.17   37.30   77.00
```

3.2 Activity: Quartiles

Using the *USJudgeRatings*[3] data frame, find the summary for *WRIT* and include that information in the deliverable document for this lab.

4 IQR (Inter-Quartile Range)

Another measure of dispersion that is occasionally used is the Inter-Quartile Range (IQR), the difference between Q1 and Q3. To find the IQR for a variable, use the IQR function. For example, R reports that the difference between Q1 and Q3 for the girth variable in the *trees* data frame is 4.2. Since the IQR removes outliers, it occasionally provides a better indication of the data dispersion.

4.1 Demonstration: IQR (Inter-Quartile Range)

To find the Inter-Quartile Range of a variable, use R's IQR function. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button. Note: IQR is one of the few functions in R that uses upper-case letters.

```
1  # IQR for trees
2  IQR(trees$Girth)
3  IQR(trees$Height)
4  IQR(trees$Volume)
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Lines 2-4: Display the Inter-Quartile Range for three variables in the *trees* data frame.

R reports the following IQRs.

```
[1] 4.2
[1] 8
[1] 17.9
```

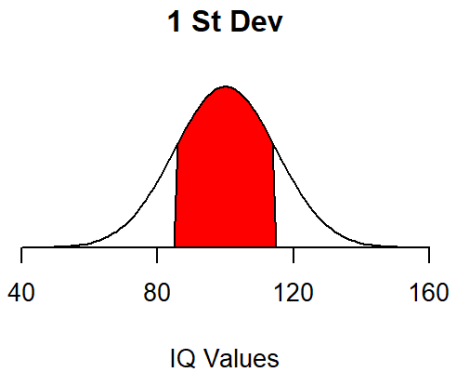
4.2 Activity: Inter-Quartile Range (IQR)

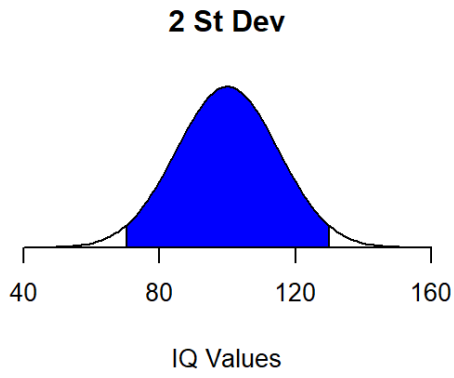
Using the *swiss*[4] data frame, find the IQR for *Agriculture* and include that number in the deliverable document for this lab.

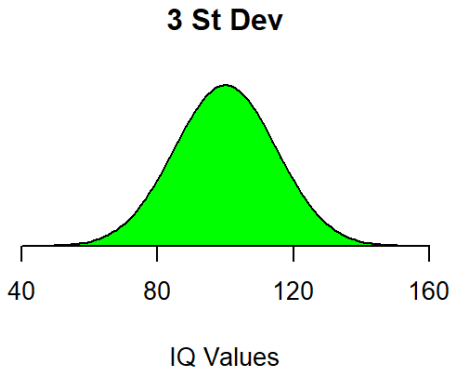
5 Standard Deviation

The standard deviation of a variable is a number that indicates how "scattered" the data is from the mean. Data with significant variance from the mean leads to a large standard deviation.

About 68.2% of the samples will lie closer to the mean than the standard deviation. So, one standard deviation explains about 68.2% of the variance from the mean. To show this concept graphically, consider the three following representations of IQ scores.







The mean of an IQ distribution is 100, and one standard deviation is 15. The shaded area under the first curve, in red, includes about 68.2% of all IQ scores. In the same way, two standard deviations from the mean would include about 95.4% of the data points and are illustrated in blue in the second image. Three standard deviations would include more than 99.7% of the data points, illustrated in green in the third image.

As one last example, imagine several classes with 500 students where the professors administered an examination worth 100 points. If the mean score for that examination was 80 and the standard deviation was 5, then the professors would know that the scores were tightly grouped (341 scores of the 500 (68.2%) were between 75-85, within 5 points of the mean), and this would probably be good news. On the other hand, if the mean score was 60 and the standard deviation was 15, then the scores were "all over the place" (more precisely, 341 scores of the 500 were between 45-75), and that may mean that the professors would have to re-think how the lesson was taught or maybe that the examination itself was flawed.

It is difficult to categorically state whether a specific standard deviation is good or bad; it is simply a measure of how concentrated the data is around the mean. For something like a manufacturing process where the required tolerance for the parts being produced is tight, the standard deviation for the weights of random samples pulled off the line must be very small; the parts must be nearly identical. However, in another context, the standard deviation may be pretty large. Imagine measuring the time it takes a group of high school students to run 100 yards. Some would be very fast, but others would be much slower, and the standard deviation for that data would likely be large.

5.1 Demonstration: Standard Deviation

To find the standard deviation of a variable, use R's *sd* function. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Standard deviation for trees
2  sd(trees$Girth)
3  sd(trees$Height)
4  sd(trees$Volume)
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Lines 2-4: Display the standard deviation for three different variables in the *trees* data frame.

R reports the following standard deviations.

```
[1] 3.138139
[1] 6.371813
[1] 16.43785
```

The results show that the tree girth has a standard deviation of about 3.14 inches while the range was 8.3-20.6 inches (the range function was found in an earlier activity). The range is extensive, but the standard deviation indicates that the girths of most trees were within about 6 inches of each other, so the trees were about the same size. The standard deviation can provide essential information to help interpret the range.

5.2 Activity: Standard Deviation

Using the *stackloss*[5] data frame, find the standard deviation for *Air.Flow* and include that number in the deliverable document for this lab.

6 Deliverable

Complete the activities in this lab and consolidate the responses into a single document. Name the document with your name and "Lab 3," like "George Self Lab 3," and submit that document for grading.

---

[1] The trees data frame provides measurements of the girth, height, and volume of timber in 31 felled black cherry trees. The data frame includes these variables: trees$Girth, trees$Height, and trees$Volume. Note that girth is the diameter of the tree (in inches) measured at 4 ft 6 in above the ground.

[2] The faithful data frame contains observations about the Old Faithful Geyser in Yellowstone National Park. The data frame includes these variables: faithful$waiting, the waiting time between eruptions, and faithful$eruptions, the eruption time in minutes.

[3] The USJudgeRatings data frame contains lawyers' ratings of state judges in the US Superior Court. The data frame includes these variables: USJudgeRatings$CONT, USJudgeRatings$INTG, USJudgeRatings$DMNR, USJudgeRatings$DILG, USJudgeRatings$CFMG, USJudgeRatings$DECI, USJudgeRatings$PREP, USJudgeRatings$FAMI, USJudgeRatings$ORAL, USJudgeRatings$WRIT, USJudgeRatings$PHYS, and USJudgeRatings$RTEN.

[4] The swiss data frame contains data about Swiss fertility and socioeconomic indicators from 1888. The data frame includes these variables: swiss$Fertility, swiss$Agriculture, swiss$Examination, swiss$Education, swiss$Catholic, and swiss$Infant.Mortality.

[5] The stackloss data frame contains observations on stack-loss (the loss of acid through the stack) in a chemical plant for the conversion of ammonia to nitric acid. The data frame includes four variables: stackloss$Air.Flow, stackloss$Water.Temp, stackloss$Acid.Conc, and stackloss$stack.loss.

Lab 4: Visualizing Descriptives

1 Introduction

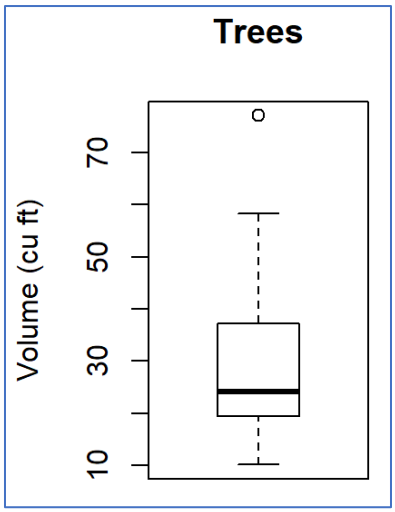
R makes it easy to calculate various data descriptives, as in Lab 3; however, most people find it easier to understand them when presented graphically. Fortunately, R has a great graphic tool for visualizing data descriptives: Boxplot (sometimes called a "Box and Whisker" plot). A Boxplot graphically illustrates Q1, the mean, the median, Q3, outlier boundaries, and outliers (if any are present).

2 About Visualizations

Several labs in this class focus on data visualization because it is a critically important tool for analysis. Visualizations are helpful in two different phases of the analysis process: exploration and explanation. Researchers are looking for interesting relationships in the data in the exploration phase. Those relationships are often difficult to detect in a table full of numbers, but a visualization makes them instantly clear. For example, here are two ways to look at the *volume* variable in the *trees*[1] data frame.

```
 [1] 10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2 22.6 19.9 24.2 21.0
[13]21.4 21.3 19.1 22.2 33.8 27.4 25.7 24.9 34.5 31.7 36.3 38.3
[25]42.6 55.4 55.7 58.3 51.5 51.0 77.0
```

The above table shows the measured volume for 31 Black Cherry Trees. Researchers looking at these numbers would not be able to detect very much. However, a simple box plot reveals a few interesting details, such as the presence of one upper outlier (the circle at the top of the plot) and that the data is positively skewed (the dark "median" line is low in the box).



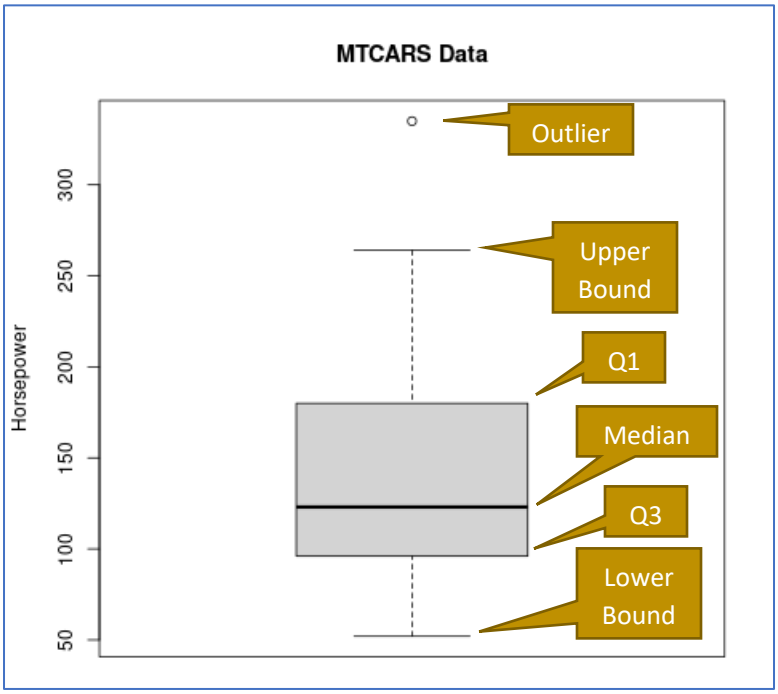
Visualizations like this make it easy to detect patterns that are not obvious from the data table. Researchers commonly use these types of visualizations in the exploratory phase of analysis. In the explanatory phase, where research findings are revealed to the public, different visualizations that are easier to understand are

more appropriate. Researchers must carefully consider the many available visualizations for exploration or explanation to ensure that they help rather than hinder understanding.

3 Boxplots

Following is the summary data for *hp* from the mtcars[2] data frame and the boxplot for that same data.

```
   Min. 1st Qu. Median Mean  3rd Qu. Max.
   52.0 96.5    123.0  146.7 180.0   335.0
```



A dark line indicates the median at 123, Q1 is 96.5 (the lower edge of the box), and Q3 is 180 (the upper edge of the box). The following equations show how the "whiskers" are calculated. They indicate the limits for outliers, so any data that lies outside those whiskers are outliers indicated by a small circle on the boxplot.

$$LowerBoundary = Q1 - (1.5 * IRQ)$$
$$LowerBoundary = 96.5 - (15 * 83.5)$$
$$LowerBoundary = 96.5 - 125.25$$
$$LowerBoundary = 0$$

Since the smallest value in the variable, 52, is larger than the calculated lower boundary, 0, the lower whisker is 52.

$$UpperBoundary = Q3 + (1.5 * IRQ)$$
$$UpperBoundary = 180.0 + (1.5 * 83.5)$$
$$UpperBoundary = 180.0 + 125.25$$
$$UpperBoundary = 305.25$$

Since the calculated upper boundary, 305.25, is smaller than the largest value in the variable, 335.0, the upper whisker is placed at the largest data value smaller than or equal to 305.25, or 264.

The circle above the boxplot represents an outlier, which is 335. If the data is in a normal distribution, the whiskers will usually enclose all values in the variable, and outliers will be rare.

3.1 Demonstration: Boxplots

To generate a boxplot for a variable, use R's boxplot function. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Summary and boxplot for rock$area
2  summary(rock$area)
3  boxplot(rock$area, main="Rock Area")
```

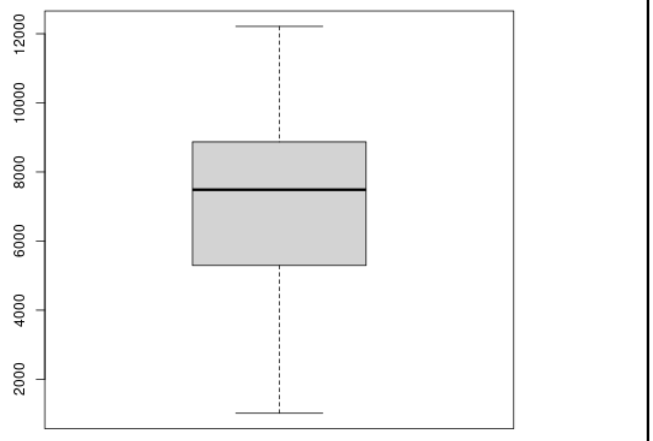
Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: Display the summary information for *rock$area*[3].

Line 3: Create the boxplot for *rock$area*. Note that the *main* attribute adds a title above the boxplot.

Following is the result of the script.

```
Min. 1st Qu. Median Mean 3rd Qu. Max.
1016 5305    7487   7188 8870    12212
```



To copy the boxplot, right-click on the image, select "Copy Image," and paste it into the Word document.

3.2 Activity: Boxplot

Using the *rock* data frame, generate a summary and boxplot for the *shape* variable. Include the summary information and the boxplot in the deliverable document for this lab.

4 Outliers

Considering data observations far outside the "normal" in any given variable is helpful. For example, imagine a neighborhood where the houses cost about $150,000. Suppose someone wins a lottery and decides to build a $500,000 house in that same neighborhood. The house's value would be an outlier in a data frame that contains the house values for the neighborhood; that is, it would be outside the "average" house value. Outliers are essential when discussing data since they tend to skew certain types of measures.

Statistically, outliers are defined as values outside boundaries that are 1.5 times the Inter-Quartile Range (IQR) below the first quartile or above the third quartile. R includes a function that displays the values used to create a boxplot, including outliers, so the values of any outliers can be easily determined.

4.1 Demonstration: Outliers

To determine the values R used to generate a boxplot, use the command *boxplot.stats(rock$shape)*. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Boxplot stats for shape
2  boxplot.stats(rock$shape)
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: Display the information used to generate the boxplot

The output has four different items.

```
$stats
[1] 0.0903296 0.1621295 0.1988620 0.2626890 0.3412730
$n
[1] 48
$conf
[1] 0.1759291 0.2217949
$out
[1] 0.438712 0.464125 0.420477
```

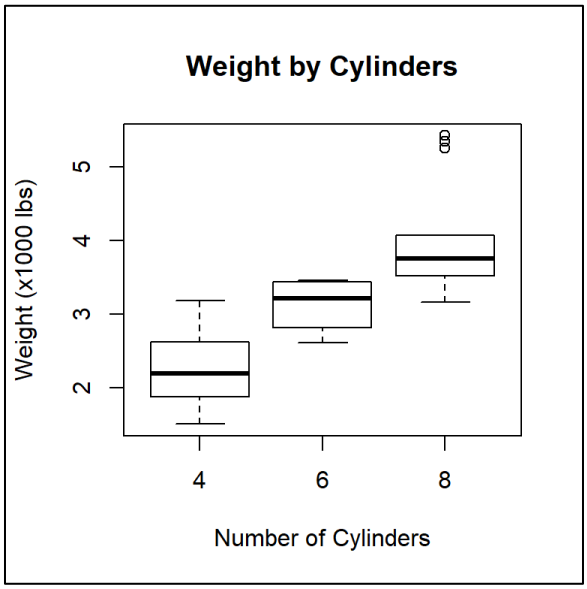
1. **$stats** are five horizontal line locations. The lower whisker is at 0.0903, Q1 (the lower edge of the box) is at 0.1621, the median (the heavy line in the middle of the box) is at 0.1988, Q3 (the upper edge of the box) is at 0.2626, and the upper whisker is at 0.3412.
2. **$n** is the number of observations in the variable, or 48 in this case, since 48 rocks were measured.
3. **$conf** is the value on the y-axis that would be used to mark a 95% confidence level, but that statistic is not used in this lab.
4. **$out** are the values of the outliers, and *rock$shape* has three: 0.438712, 0,464125, and 0.420477. If there are no outliers, that line reports *numeric(0)* to indicate zero outliers.

4.2 Activity: Outliers

Using the *trees* data frame, find the outlier for *Volume* and include that number in the deliverable document for this lab. Notice that the variable name, *Volume*, starts with a capital letter.

5 Grouped Boxplots

Boxplots become much more helpful when more than one data item is plotted side-by-side for comparison. For example, the following boxplots from the *MT Cars* data frame help determine if there is a difference in automobile weight by the number of cylinders in the engine.



By comparing the three boxplots, it is easy to see that the more cylinders an engine has, the more the automobile weighs since the plots tend to be "higher" as the number of cylinders increases. Also, notice that the plot for 8-cylinder cars does not have an upper whisker since it is precisely the same as Q3 but includes outliers. It is also interesting to note that the whiskers for the three plots overlap, indicating, for example, that some 4-cylinder cars are heavier than some 6-cylinder cars.

Color can be added to a boxplot to make it more pleasing and easier to understand.

5.1 Demonstration: Grouped Boxplots with Color

R's boxplot function can be used with a few modifications to generate grouped boxplots with color. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Boxplot of airquality data
2  boxplot(Temp ~ Month,
3  data = airquality,
4  main = "Temp By Month",
5  col = rainbow(8))
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: *Temp ~ Month* This tells R to calculate the boxplot for the *Temperature* variable but group those temperatures by *Month*. It is important to remember the order of these two variables. The first is the continuous data that should be analyzed, and the second is the grouping variable. Also, notice that

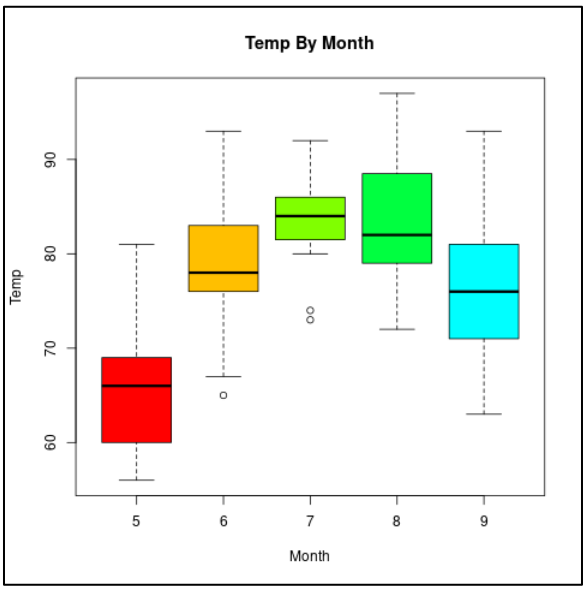
an open parenthesis on this line is not closed until line five. Commands in R are frequently spread over several lines to make them easier to read.

Line 3: *data = airquality*. The $ operator prepended the variable name to the data frame. However, for simplicity, many R functions are designed to enter only the variable names and then later specify the data frame. In this case, the *airquality*[4] data frame is identified as the source for plotting the two variables.

Line 4: *main = "Temp By Month"* This is the title for the boxplot and is automatically printed in large font above the boxplot.

Line 5: *col = rainbow(8)* This sets the color palette to rainbow and instructs R to use eight colors from that palette. It is often helpful to experiment with the number of colors requested from the palette. The colors selected will change depending on the number requested; some combinations may be easier to read than others.

R generates the following boxplot.



5.2 Activity: Grouped Boxplot with Color

Using the *airquality* data frame, generate the following boxplot.

- Variable: Ozone grouped by Month
- Main: Ozone by Month
- Color: eight colors from the rainbow pallet

Include the generated boxplot in the deliverable document for this lab.

6 Deliverable

Complete the activities in this lab and consolidate the responses into a single document. Name the document with your name and "Lab 4," like "George Self Lab 4," and submit that document for grading.

[1] The trees data frame provides measurements of the girth, height, and volume of timber in 31 felled black cherry trees. The data frame includes these variables: trees$Girth, trees$Height, and trees$Volume. Note that girth is the diameter of the tree (in inches) measured at 4 ft 6 in above the ground.

[2] The mtcars data frame was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models). The data frame includes these variables: mtcars$mpg, mtcars$cyl, mtcars$disp, mtcars$hp, mtcars$drat, mtcars$wt, mtcars$qsec, mtcars$vs, mtcars$am, mtcars$gear, and mtcars$carb.

[3] The rock data frame provides measurements on 48 petroleum Rock samples. The data frame includes these variables: rock$area, rock$peri, rock$shape, and rock$perm.

[4] The airquality data frame contains information from New York air quality measurements. The data frame includes these variables: airquality$Ozone, airquality$Solar.R, airquality$Wind, airqualtiy$Temp, airquality$Month, and airquality$Day.

Lab 5: Frequency Tables

1 Introduction

Categorical data items are typically reported in frequency tables and crosstabs, where the counts for a particular item are displayed. The only difference between these two tables is the number of dimensions; frequency tables display only a single variable while a crosstab displays two or more variables. Both tables are commonly used to display polling data during an election. They would list things like the number of voters who support some proposition (frequency table) or that same data broken out by party affiliation, sex, age, or other categories (crosstab). This lab explores both types of tables.

2 Frequency Table

A frequency table is a one-dimensional table that lists a count of the number of times that some categorical data item appears in a variable. For example, consider the following table, which lists the number of cars for each number of cylinders in the *mtcars*[i] data frame.

```
4  6 8
11 7 14
```

This table shows that 11 cars in the data frame had four cylinders, seven had six cylinders, and 14 had eight cylinders.

Frequency tables are only helpful for categorical data items. To illustrate why this is true, imagine creating a survey for all the students at the University of Arizona and including "age" (continuous data) as one of the survey questions. Attempting to create a frequency table for the respondents' ages would have more than 65 columns since student ages would range from about 15 to more than 80, and each column would report the number of students for that age. While R could create a large frequency table, it would have so many columns that it would be virtually unusable. Typically, if continuous data needs to be displayed in a table, the data is grouped in some way, like ages 15-19 and 20-24, so there would be a manageable number of groups to display.

2.1 Demonstration: Frequency Table

To create a frequency table, use the *table* command. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Count of cars by gears
2  table(mtcars$gear)
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: Create a simple frequency table listing the number of cars in the *mtcars* data frame by the number of forward gears.

Following is the result of the script.

```
3  4  5
15 12 5
```

2.2 Activity: Frequency Table

Using the *mtcars* data frame, generate a frequency table for the number of cars grouped by the number of carburetors (*mtcars$carb*). Include the table in the deliverable document for this lab.

3 Margins

It is often helpful to include the total number of items counted in a frequency table, and the *addmargins* command provides that total. For example, consider the following table, the *mtcars cylinder* table demonstrated above but with the total number of cars.

```
4  6 8  Sum
11 7 14 32
```

3.1 Demonstration: Margins

To create a frequency table with margins, use the *addmargins* command. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Count of cars by gears
2  addmargins(table(mtcars$gear))
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: Create a frequency table with margins listing the number of cars in the *mtcars* data frame grouped by the number of forward gears.

3.2 Activity: Margins

Using the *mtcars* data frame, generate a frequency table for cars grouped by the number of carburetors (*mtcars$carb*), and include the margins on this table. Include the frequency table in the deliverable document for this lab.

4 Proportion Table (Proptable)

Occasionally, researchers prefer to present percentages rather than raw numbers since those may be easier to interpret. Here is a proportion table of the number of cylinders for cars in the *mtcars* data frame.

```
4      6      8
34.375 21.875 43.750
```

Thus, about 44% of the cars have eight cylinders.

4.1 Demonstration: Proportion Table (Proptable)

To create a proportion table, use the *prop.table* command. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Proportion table
2  prop.table(table(mtcars$gear)) * 100
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: Create a proportion table listing the number of cars in the *mtcars* data frame by the forward gears. The proportions are multiplied by 100 to make them percentages rather than decimals.

Following is the result of the script.

```
      3       4       5
46.875 37.500 15.625
```

4.2 Activity: Proportion Table (Proptable)

Using the *mtcars* data frame, generate a proportion table for *carburetors*. The proportions should be multiplied by 100 to make them percentages. Include the table in the deliverable document for this lab.

5 Crosstab

A crosstab (sometimes called a contingency table or pivot table) is a table of frequencies used to display the relationship between two or more categorical variables. As an example of a crosstab, consider a table from the *mtcars* data frame that lists the number of cars by forward gears and the number of cylinders.

```
     cyl
gear  4  6  8
   3  1  2 12
   4  8  4  0
   5  2  1  2
```

In this case, one car had a four-cylinder engine and three forward gears, while 12 cars had eight cylinders and three forward gears. Using a crosstab, a researcher can determine the frequency of observations (number of cars) by two criteria (gears and cylinders).

5.1 Demonstration: Crosstab

To create a crosstab, use the *xtabs* command. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Count of gears and cylinders
2  xtabs(~mtcars$carb+mtcars$cyl)
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: The *xtabs* command (for "Crosstabs") creates a crosstab for the variables entered. It is essential to notice the tilde character in this function. In many R commands, the tilde separates two parts of a formula. The part before the tilde is the data values to be acted upon, and the second part is the grouping variables. In Line 2, no data values are specified before the tilde, so R will count the number of times the various groups appear. The "row" group is listed first and then the "column" group. For example, "carb 1 - cylinder 4" appears in the data frame five times.

Following is the result of the script.

```
            mtcars$cyl
mtcars$carb  4  6  8
           1 5  2  0
           2 6  0  4
           3 0  0  3
           4 0  4  6
           6 0  1  0
           8 0  0  1
```

5.2 Activity: Crosstab

Using the *mtcars* data frame, create a crosstab of *carb* and *gear*. Include the crosstab in the deliverable document for this lab.

6 Multi-Dimensional Crosstab

A crosstab can contain more than two dimensions. As an example, consider the *mtcars* data frame. A researcher wanted to know if there is any relationship between the number of forward gears, the number of cylinders, and the transmission type. Here is the crosstab that was created.

```
am = 0
   gear
cyl  3  4  5
  4  1  2  0
  6  2  2  0
  8 12  0  0

am = 1
   gear
cyl  3  4  5
  4  0  6  2
  6  0  2  1
  8  0  0  2
```

When *am* is 0 (the code for automatic transmission), there were no cars with five forward gears, and when *am* is 1 (manual transmission), there were no cars with three forward gears.

6.1 Demonstration: Multi-dimensional Crosstab

To create a multi-dimensional crosstab, use the *xtabs* command with three variables. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and then the "Run" button.

```
1  # Count of carbs and cylinders by engine type
2  xtabs(~carb+cyl+vs, data = mtcars)
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: The *xtabs* command creates a crosstab for the variables entered. With nothing on the left side of the tilde, R will return a count of the variables. Since there are three grouping variables, *gear*, *cyl*, and *vs* (V-8 or straight engine type), R will count instances for all three variables. For example, "gear 4" and "cylinder 6" appear twice when the type is zero and twice when the type is one. Note that *the data =*

mtcars parameter is specified, so the *$* operator does not need to be used for each variable, making the formula easier to read.

Following is the result of the script.

```
vs = 0
   gear
cyl  3  4  5
  4  0  0  1
  6  0  2  1
  8 12  0  2

vs = 1
   gear
cyl  3  4  5
  4  1  8  1
  6  2  2  0
  8  0  0  0
```

6.2 Activity: Multi-dimensional Crosstab

Using the *mtcars* data frame, create a multi-dimensional crosstab of *gear*, *cylinder*, and *carburetor*. Include the crosstab in the deliverable document for this lab.

7 Calculated Crosstab

Each of the presented crosstabs has only data counts; however, a crosstab can also display calculated values using the aggregate function. For example, the following table lists the mean horsepower for automobiles calculated by the number of forward gears and engine cylinders from the *mtcars* data frame.

```
gear cyl      hp
   3   4  97.0000
   4   4  76.0000
   5   4 102.0000
   3   6 107.5000
   4   6 116.5000
   5   6 175.0000
   3   8 194.1667
   5   8 299.5000
```

7.1 Demonstration: Calculated Crosstab

To create a calculated crosstab, use the *aggregate* command. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Mean of displacement by cylinders
2  aggregate(disp~cyl, data = mtcars, FUN = mean)
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: This line uses the aggregate command to calculate the mean displacement for each category of cylinders. This command has a tilde formula where displacement (*disp*) will be aggregated for each cylinder category (*cyl*). Calculating values come first (*disp*), and the grouping variable (*cyl*) comes

second. Next is the name of the data frame used, *mtcars*. Finally, the statistical function is listed as *FUN = mean*. Note that the keyword *FUN* is in capital letters. This aggregate function determines that four-cylinder cars have a mean displacement of slightly more than 105 cubic inches.

```
  cyl     disp
1   4 105.1364
2   6 183.3143
3   8 353.1000
```

7.2 Activity: Calculated Crosstab

Using the *mtcars* data frame, create a calculated crosstab of the mean miles-per-gallon (*mpg*) when grouped by carburetors (*carb*). Include the crosstab in the deliverable document for this lab.

8 Rounding

To facilitate understanding, R makes it easy for researchers to round the results of calculations to whatever level is desired. The R command to round a number is *round()*. The number to be rounded is listed first, and the number of decimal places is listed second. So, round(1.3498, 2) would be rounded to 1.35, and round(1.3498, 1) would be rounded to 1.3. It is important to note that R still uses the complete decimal number for calculations even if the displayed value is rounded. Also, when rounding a 5, "banker's" rounding is used, "go to the even digit." So, 2.5 is rounded to 2.0, but 3.5 is rounded to 4.0.

8.1 Demonstration: Rounding

Here is the same *aggregate* command used in the previous example but with rounded results. The *aggregate* command is the same, but it is contained in a *round* function to round off the calculations. The result is rounded to two decimal places. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1 # Mean of displacement by cylinders
2 round(aggregate(disp~cyl, data = mtcars, FUN = mean), digits = 2)
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: This line surrounds the *aggregate* command to calculate the mean displacement for each category of cylinders, rounded to two decimal places. The aggregate command is the same as used in the previous section.

```
  cyl   disp
1   4 105.14
2   6 183.31
3   8 353.10
```

8.2 Activity: Rounding

Using the *mtcars* data frame, create a calculated crosstab of the mean miles-per-gallon (*mpg*) when grouped by carburetors (*carb*). Round the results to two decimal places. Include the crosstab in the deliverable document for this lab.

9 Deliverable

Complete the activities in this lab and consolidate the responses into a single document. Name the document with your name and "Lab 5," like "George Self Lab 5," and submit that document for grading.

[i] The mtcars data frame was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models). The data frame includes these variables: mtcars$mpg, mtcars$cyl, mtcars$disp, mtcars$hp, mtcars$drat, mtcars$wt, mtcars$qsec, mtcars$vs, mtcars$am, mtcars$gear, and mtcars$carb.

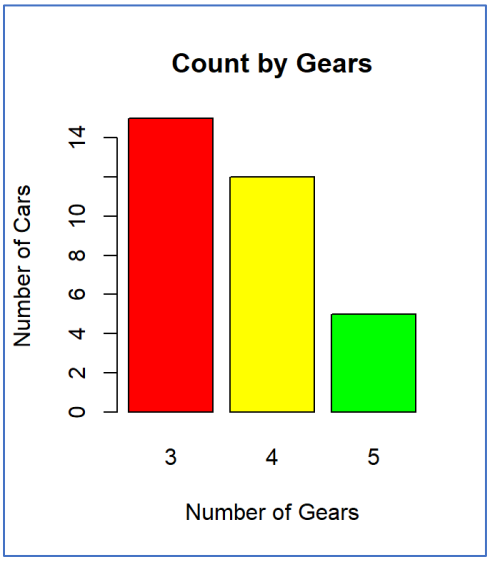
Lab 6: Visualizing Frequency

1 Introduction

Categorical data items are often reported using frequency tables where the number of times a particular item was observed is displayed. However, there are many ways to visualize categorical data. People using graphics rather than tables often find it easier to understand the underlying data.

2 Bar Plot

A bar plot is used to display the frequency count for categorical data. The following figure is a bar plot showing the number of automobiles with three, four, and five gears found in the *mtcars*[1] data frame.



These visuals are more effective than a table full of numbers and are easy to generate with R.

2.1 Demonstration: Bar Plot

To generate a bar plot, use the barplot function. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button. Notice that the barplot function has been broken over several lines to make it easier to read and understand.

```
1  # Simple Bar Plot
2  barplot(height = table(mtcars$gear),
3    main = "Number of Cars by Gears",
4    xlab = "Gears",
5    ylab = "Count",
6    col = cm.colors(3)
7  )
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: This creates a bar plot using the barplot function. The first argument sent to the function is the data source for the heights of each bar in the plot. In this case, R creates a table from the gears variable in *mtcars* and then uses that table as data input for the plot. All other lines in this script embellish the bar plot to make it more usable.

Line 3: The "main" attribute specifies the title for the bar plot.

Line 4: This creates the label for the x-axis.

Line 5: This creates the label for the y-axis.

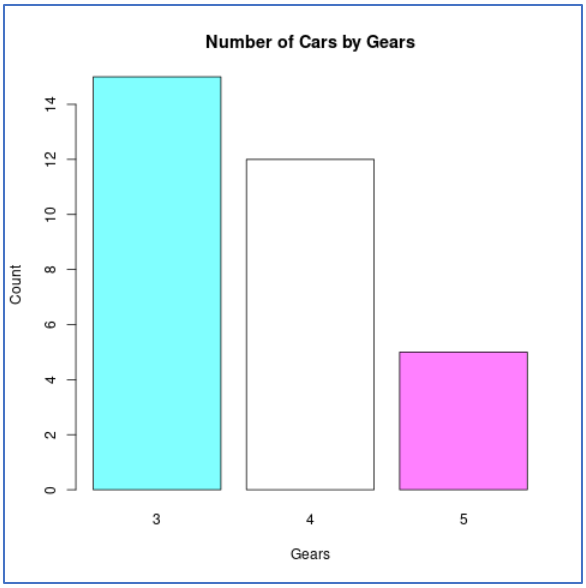
Line 6: This sets the color palette for the graph. In this case, the "cm.colors" palette is used. Three colors were requested from that palette but specifying any number larger than three would have worked and created a slightly different palette. Experimentation is needed to find the most suitable palette for any given graph.

Line 7: This parenthesis closes the barplot function started on line 2.

The Snippets interface generates the barplot in the results box. To transfer any plot from Snippets to a Word document:

1. Right-click on the plot and select Copy Image.
2. Click the location for the plot in the Word document to set the insertion point.
3. Click the "Paste" down-arrow in the Word Ribbon.
4. Select the "Picture" option.
5. Resize and apply other modifications to the image as needed.

Following is the result of the script.



2.2 Activity: Bar Plot

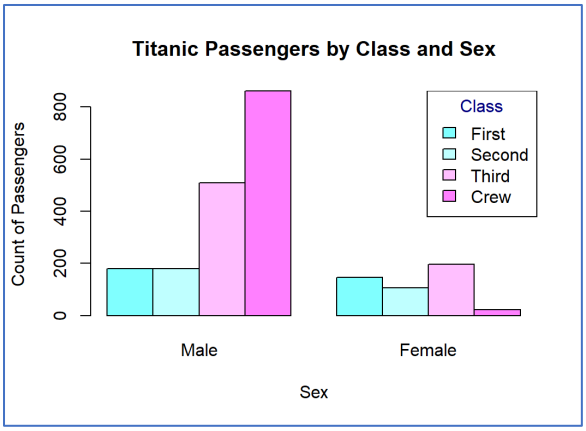
Using the *mtcars* data frame, create a bar plot for cylinders (*cyl*). The plot should meet these specifications:

- Title: Number of Cars by Cylinders
- X-axis label: Cylinders
- Y-axis label: Count
- Color: three colors from the topo.colors palette

Include the barplot in the deliverable document for this lab.

3 Clustered Bar Plot

A clustered bar plot (called a "Grouped Bar plot") displays two or more categorical variables. In general, clustered bar plots are best at showing relationships between variables but not so good for determining the size of each variable. The following plot showed the number of passengers on board the *Titanic*[2] when it sank. While it is easy to determine that there were many more males than females on board, it is impossible to read the exact bar height of third-class males, for example.



3.1 Demonstration: Clustered Bar Plot

To generate a clustered bar plot, use the barplot function. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button. Notice that the barplot function has been broken over several lines to make it easier to understand.

```
1  # Clustered Bar Plot with Gradient Colors
2  colpal <- colorRampPalette(c("blue", "white"))
3  barplot(height = table(mtcars$cyl, mtcars$gear),
4    main = "Cars by Gears and Cylinders",
5    xlab = "Gears",
6    ylab = "Count",
7    legend = TRUE,
8    beside = TRUE,
9    args.legend = list(title = "Cylinders"),
10   col = colpal(3)
11 )
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: The first line of the script creates a custom color palette named *colpal* (for "color palette"), containing color codes. In this case, the *colorRampPalette* function creates the codes for color gradients between blue and white.

Line 3: This begins the barplot function. It creates a table that contains the counts for cyl and gear in the *mtcars* data frame and then uses that table to produce the bar plot. Note the order of the variables in the table command. The grouping variable is listed second. In this example, the cars are grouped by gears, and the number of cylinders is displayed within each group. The count of cars determines the height of each bar in each group.

Lines 4-6: These lines are essentially the same as for a simple bar plot described in the previous demonstration.

Line 7: Setting legend to *TRUE* displays a legend in the corner of the plot. Whenever more than one variable is plotted, it is essential to display a legend for the reader. In this case, the legend displays the colors used for the *cyl* variable.

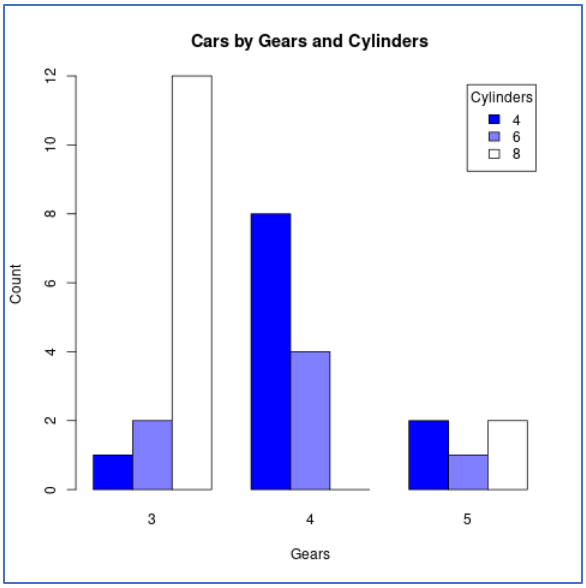
Line 8: A stacked bar is the default type of plot, but Line 8 instructs R to create a plot with the variables beside each other. "Stacked" plots are described in the next section of this lab.

Line 9: This odd-looking line adds a title to the legend; otherwise, users would be confused about the meaning of the various colors used in the plot.

Line 10: This selects three colors from the *colpal* variable created in Line 2.

Line 11: This parenthesis closes the barplot function started on line 3.

Following is the result of the script.



3.2 Activity: Clustered Bar Plot

Using the *mtcars* data frame, create a clustered bar plot that shows the number of cars by *vs* (engine type: V-8 or Straight) and *gear* (number of forward gears). The plot should have three clusters (gears 3, 4, and 5), and each should have two engine types (0 and 1). The plot should meet these specifications:

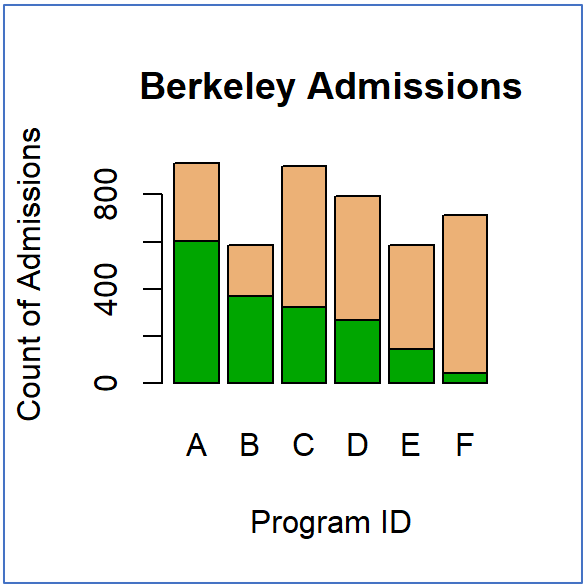
- Title: Cars by Gears and Engine Types
- X-axis label: Gears
- Y-axis label: Count
- Color: two values from a custom palette using tan to brown
- Legend title: Engine

Include the barplot in the deliverable document for this lab.

4 Stacked Bar Plot

A stacked bar plot has one variable stacked on top of another. These are very difficult to read and should only be used to make broad generalizations. Consider, for example, the following figure. This plot shows the

admission to the University of California at Berkeleyfor six programs. The top part of each bar (in brown) is the number admitted, while the bottom part of each bar (in green) is the number rejected. Examine programs D and E. Were more students accepted in D or E? Because these two values do not have the same baseline, it is impossible to tell which is more significant.



4.1 Demonstration: Stacked Bar Plot

To generate a clustered bar plot, use the barplot function. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button. Notice that the following script is the same barplot function used in the clustered bar plots above, except *beside = TRUE* is missing. By default, bar plots are stacked in R, so if the "*beside*" argument is missing (or set to "*FALSE*"), then the result is a stacked bar plot.

```
1  # Stacked Bar Plot with Gradient Colors
2  colpal <- colorRampPalette(c("brown", "white"))
3  barplot(height = table(mtcars$cyl, mtcars$gear),
4    main = "Cars by Gears and Cylinders",
5    xlab = "Gears",
6    ylab = "Count",
7    legend = TRUE,
8    args.legend = list(title = "Cylinders"),
9    col = colpal(3)
10 )
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: The first line of the script creates a custom color palette named *colpal* (for "color palette"), containing color codes. In this case, the *colorRampPalette* function creates the codes for color gradients between brown and white.

Line 3: This begins the barplot function. It creates a table that contains the counts for *cyl* and *gear* in the *mtcars* data frame and then uses that table to produce the bar plot. Note the order of the variables in the table command. The grouping variable is listed second. In this example, the cars are grouped by gears, and the number of cylinders is displayed within each group. The count of cars determines the height of each bar in each group.

Lines 4-6: These lines are essentially the same as for a simple bar plot described above.

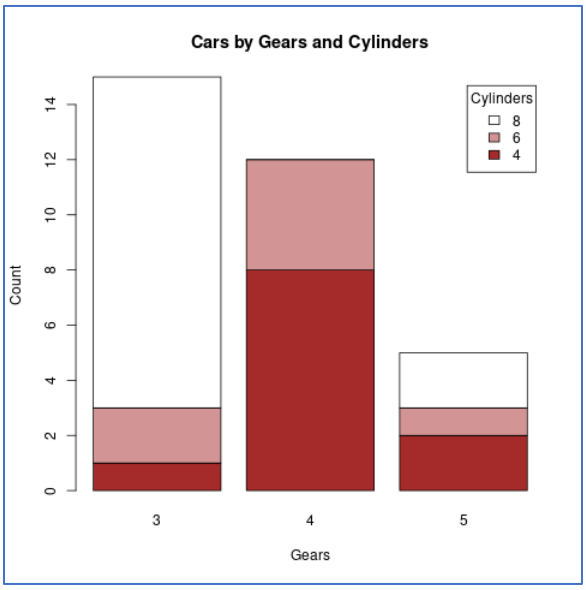
Line 7: Setting *legend* to *TRUE* displays a legend in the corner of the plot. Whenever more than one variable is plotted, it is essential to display a legend for the reader. In this case, the legend displays the colors used for the *cyl* variable.

Line 8: This odd-looking line adds a title to the legend; otherwise, users would be confused about the meaning of the various colors used in the plot.

Line 9: This selects three colors from the *colpal* variable created in Line 2.

Line 10: This parenthesis closes the barplot function started on line 3.

Following is the result of this script.



It should be evident that the bar plot created in the above script is not very useful. While it is easy to see that the number of 8-cylinder cars with three gears is much larger than the other categories, it is difficult to determine, for example, how many cars have five gears and eight cylinders. This difficulty is even worse when more than three levels are plotted for either of the two variables.

4.2 Activity: Stacked Bar Plot

Using the *mtcars* data frame, create a stacked bar plot that shows the number of cars by *vs* (engine type: V-8 or Straight) and *gear* (number of forward gears). The plot should have three clusters (gears 3, 4, and 5), and each should have two engine types (0 and 1). The plot should meet these specifications:

- Title: Cars by Gears and Engine Types
- X-axis label: Gears

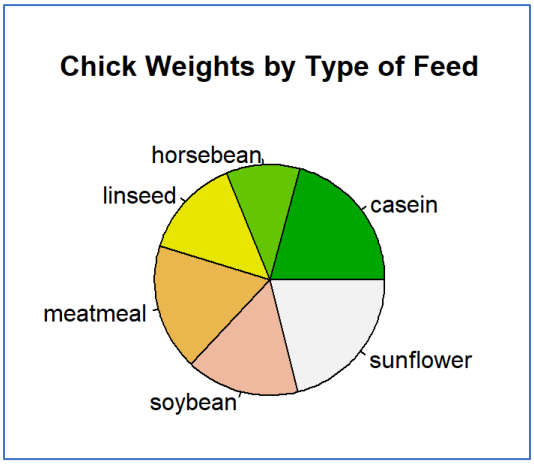
- Y-axis label: Count
- Color: two values from a custom palette using brown to burlywood
- Legend title: Engine

Include the barplot in the deliverable document for this lab.

5 Pie Chart

A pie chart is commonly used to display categorical data; however, pie charts are tricky to interpret, and effects like 3-D or "exploded" slices exacerbate the difficulty. The human brain seems to quickly compare the heights of two or more bars, as in bar plots, but the areas of two or more slices of a pie chart are challenging to compare. For this reason, pie charts should be avoided in research reports. If they are used, they should only illustrate one slice's relationship to the whole, not comparing one slice to another; moreover, no more than four or five slices should ever be presented on one chart.

The following figure shows the results of an experiment reported in the *chickwts*[3] data frame to compare the effectiveness of various feed supplements on the growth rate of chickens. This figure illustrates the problem with pie charts. Notice that "casein" seems to promote growth better than "horsebean," but it is impossible to determine if "casein" is better than "sunflower" from this chart.



5.1 Demonstration: Pie Chart

To generate a pie chart, use the pie function. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Count of Esophageal Cancer by Age
2  pie(x = table(esoph$agegp),
3    main = "Count of Esophageal Cancer by Age",
4    col = rainbow(6),
5    labels = c(levels(esoph$agegp))
6    )
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

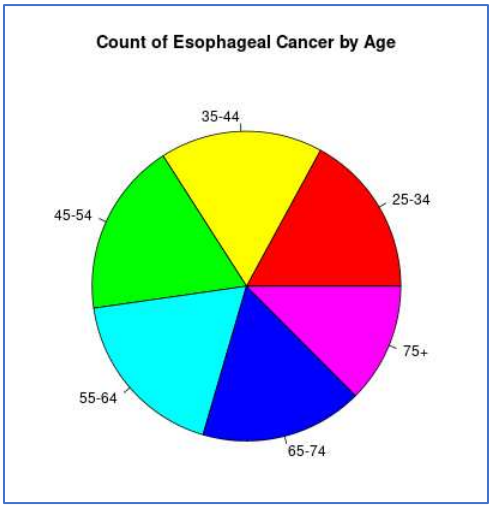
Line 2: This starts the pie chart function. The "x" parameter is the data that needs to be charted. In this line, the *agegp* (age group) variable in the *esophageal cancer*[4] data frame is extracted to a table since the pie chart function expects input as a table.

Lines 3-4: These lines define the main title and colors of the pie chart. These parameters are the same as seen in other lab graphs.

Line 5: This tells R to use the labels found in the *agegp* variable as the labels on the pie chart.

Line 6: This parenthesis closes the pie chart function started on line 2.

Following is the result of this script.



5.2 Activity: Pie Chart

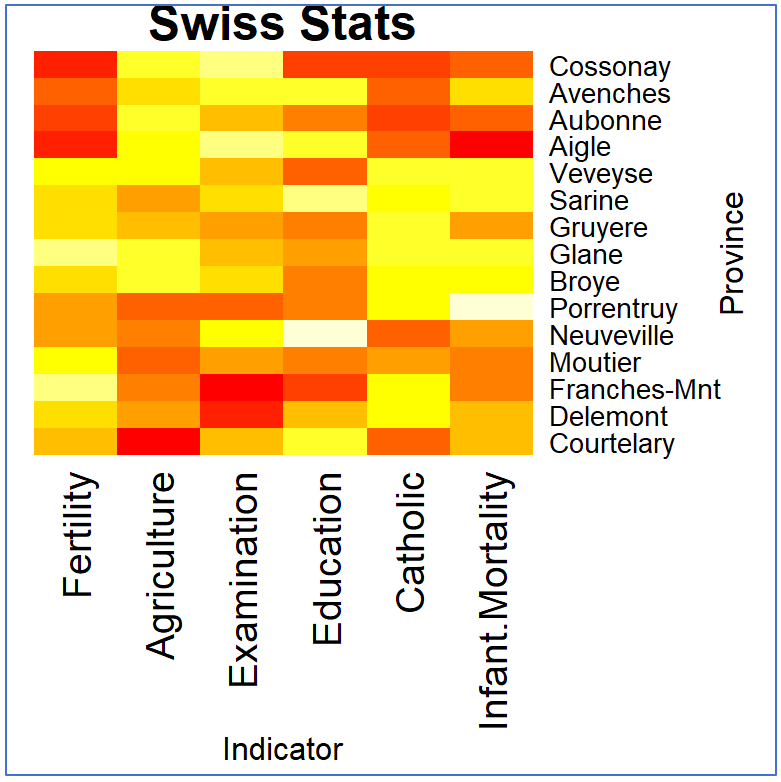
Using the *esoph* data frame, create a pie chart showing the number of people who consume tobacco (*tobgp*). The chart should meet the following specifications.

- Title: Count of Esophageal Cancer by Tobacco Use
- Colors: four colors from the rainbow pallet
- Labels: from *esoph$tobgp*

Include the pie chart in the deliverable document for this lab.

6 Heat Map

Heat maps use colors to depict the counts of variables and are commonly found around election time to depict how precincts are voting, red for Republicans and blue for Democrats. They are also routinely used on weather maps to depict areas with the most significant probability of rain or snow. While heat maps can be displayed in a geographical format where, for example, the various states are shaded to represent some factor, they are also commonly seen as a grid. The following figure shows a heat map generated from the *swiss*[5] data frame and various socio-economic indicators by province from 1888.



In a heat map produced by R, lighter colors represent more significant numbers. Thus, the province with the highest *fertility* rate is Franches-Mnt since it has the lightest color. The province with the minimum *agriculture* is Courtelary since it has the darkest color for those variables. Interpreting a heat map can be challenging since sometimes a light color is desirable but not in others. For example, the highest *education* level would be in Neuveville (desirable), but the highest *infant mortality* would be in Porrentruy (not desirable). Also, the colors are often very similar and difficult to distinguish. For example, Cossonay has a numeric value of 22 while Aigle has 21 for the *examination* variable. These two colors are slightly different, but it would not be easy to detect that from the image. The best that can be done with a heat map is to identify broad generalizations.

6.1 Demonstration: Heat Map

To generate a heat map, use the heatmap function. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Heatmap of Crime Stats by State
2  hmap <- as.matrix(USArrests[15:1,])
3  heatmap(hmap,
4    main = "Crime Stats by State",
5    xlab = "Crime",
6    ylab = "State",
7    Rowv=NA,
8    Colv=NA,
9    scale="column",
10   margins=c(10,8)
11 )
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: This line creates a matrix from rows 15 to 1 of the *USArrests*[6] data frame and stores that matrix in a variable named *hmap*. Those rows contain the data for the first 15 states alphabetically.

Line 3: This is the start of the heat map function. This line instructs R to create a heat map from the hmap matrix created in Line 2.

Line 4: The main title of the heatmap is *Crime Stats by State*.

Line 5: The x-axis is labeled Crime.

Line 6: The y-axis is labeled State.

Line 7: This suppresses the row dendrogram used to order the rows. The best way to see what this line does is to comment it out and re-run the script.

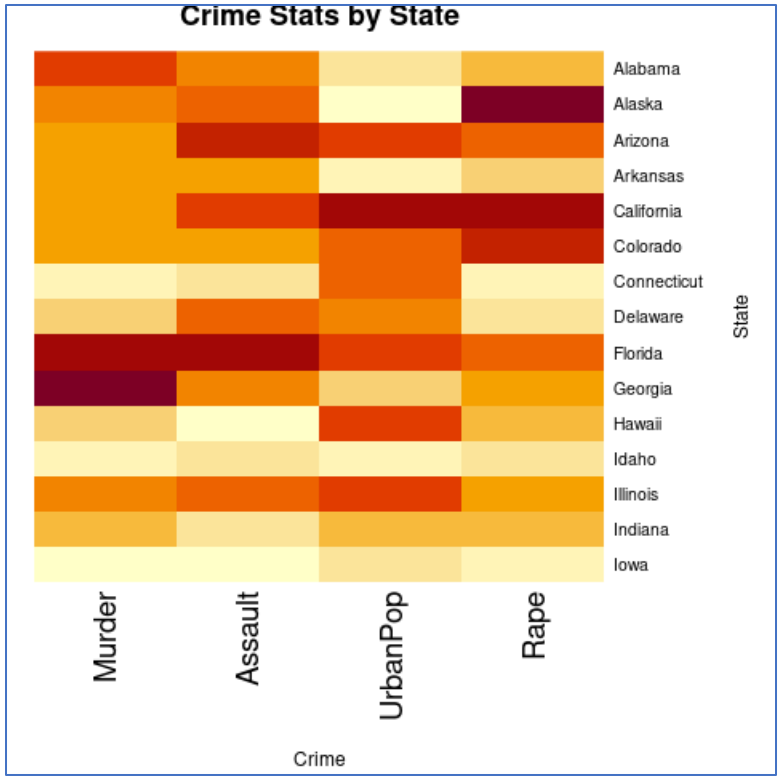
Line 8: This suppresses the column dendrogram.

Line 9: Sets the heat map to scale the columns. In this way, the color for each column cell is calculated such that the entire column's mean is zero and the standard deviation is one. The other option is to scale "row." Researchers should try both to see which better represents the data.

Line 10: This sets the bottom and right margins. The values were found by simple trial-and-error to produce the most legible heat map.

Line 11: This parenthesis closes the heat map function started on line 3.

Following is the result of this script.



6.2 Activity: Heat Map

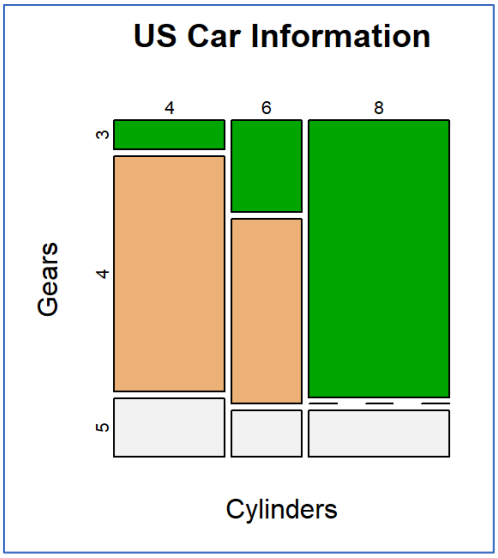
Using the *USJudgeRatings*[7] data frame, create a heat map that compares the ratings for the first 20 judges. The heat map should meet these specifications:

- Scope: specify the matrix contains rows 20:1
- Title: US Judge Ratings
- X-axis label: Characteristic
- Y-axis label: Name
- Rowv/Colv symbols: NA
- Scale: row
- Margins: c(8,10)

Include the heat map in the deliverable document for this lab.

7 Mosaic Plot

A mosaic plot indicates the relative counts of items in a data frame by sizing areas on a grid. The following figure is a mosaic plot that indicates the relationship between the number of gears and cylinders in several cars in the *mtcars* data frame. Notice that 8-cylinder cars overwhelmingly have three gears while 4-cylinder cars tend to have four gears. This plot gives a quick visual representation of the relationships between categorical variables, just as a pie chart shows the relationship between continuous variables. Mosaic plots suffer the same weaknesses as a pie chart and are, generally, rather tricky to interpret.



7.1 Demonstration: Mosaic Plot

To generate a mosaic plot, use the plot function. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Mosaic Plot of Gears vs. Cylinders
2  plot(x = table(mtcars$gear, mtcars$cyl),
3    main = "Gears vs Cylinders",
4    xlab = "Gears",
5    ylab = "Cylinders",
6    col = topo.colors(3)
7    )
```

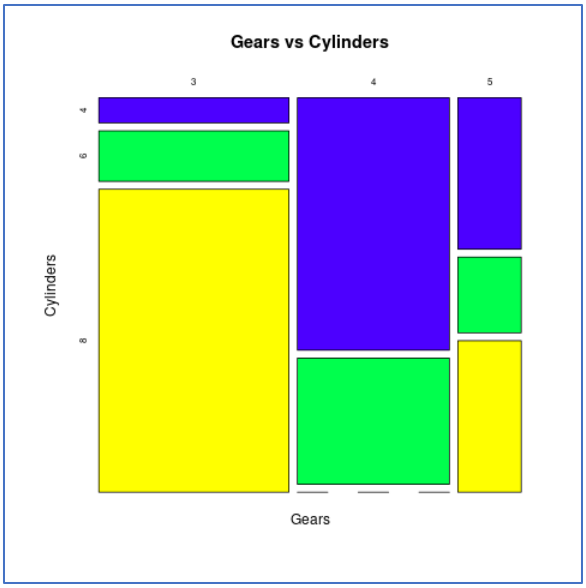
Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: A mosaic plot requires the input to be in a table format, so this line creates a table from the *gear* and *cyl* variables in the *mtcars* data frame. The table is entered as variable *x* in the mosaic plot.

Lines 3-6: These are like those used for other graphics functions and should be reasonably easy to understand.

Line 7: This parenthesis closes the mosaic plot function started on line 2.

Following is the result of this script.



7.2 Activity: Mosaic Plot

Using the *esoph* data frame, create a mosaic plot that compares the age groups (*agegp*) to tobacco use (*tobgp*). The plot should meet these specifications:

- Title: Esophageal Cancer Factors
- X-axis label: Age Groups
- Y-axis label: Tobacco use Groups
- Color: four colors from the cm.colors palette

The plot should have six columns, one for each age group, and four rows, one for each of the tobacco use groups. Include the mosaic plot in the deliverable document for this lab.

8 Deliverable

Complete the activities in this lab and consolidate the responses into a single document. Name the document with your name and "Lab 6," like "George Self Lab 6," and submit that document for grading.

---

[1] The mtcars data frame was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models). The data frame includes these variables: mtcars$mpg, mtcars$cyl, mtcars$disp, mtcars$hp, mtcars$drat, mtcars$wt, mtcars$qsec, mtcars$vs, mtcars$am, mtcars$gear, and mtcars$carb.

[2] The barchart was generated from the titanic data frame, which includes these variables: titanic$Class, titanic$Sex, titanic$Age, and titanic$Survived.

[3] The chickwts data frame contains the results of an experiment conducted to measure and compare the effectiveness of various feed supplements on the growth rate of chickens. The data frame includes these variables: chickwts$weight and chickwts$feed.

[4] The esoph data frame contains information from a case-control study of (o)esophageal cancer in Ille-et-Vilaine, France. The data frame includes these variables: esoph$agegp, esoph$alcgp, esoph$tobgp, esoph$ncases, and esoph$ncontrols.

[5] The Swiss data frame contains data about Swiss fertility and socioeconomic indicators from 1888. The data frame includes these variables: swiss$Fertility, swiss$Agriculture, swiss$Examination, swiss$Education, swiss$Catholic, and swiss$Infant.Mortality.

[6] The USArrests data frame contains statistics about the arrests per 100,000 residents for assault, murder, and rape in each of the 50 US states in 1973. Also given is the percent of the population living in urban areas. The data frame includes these variables: USArrests$Murder, USArrests$Assault, USArrests$UrbanPop, and USArrests$Rape.

[7] The USJudgeRatings data frame contains lawyers' ratings of state judges in the US Superior Court. The data frame includes these variables: USJudgeRatings$CONT, USJudgeRatings$INTG, USJudgeRatings$DMNR, USJudgeRatings$DILG, USJudgeRatings$CFMG, USJudgeRatings$DECI, USJudgeRatings$PREP, USJudgeRatings$FAMI, USJudgeRatings$ORAL, USJudgeRatings$WRIT, USJudgeRatings$PHYS, and USJudgeRatings$RTEN.

Lab 7: Correlation

1 Introduction

In some research projects, correlation describes a relationship between the independent (or x-axis) and dependent (or y-axis) variables. For example, imagine a corn production project where researchers applied a treatment to 50 acres of corn but not to another 50 acres in a nearby field. At the end of the growing season, they found that the untreated field yielded 150 bushels per acre while the treated field yielded 170 bushels per acre. This result would indicate a correlation, or relationship, between the treatment applied and the crop yield.

1.1 Causation

From the outset of this lab, it is essential to remember that correlation does not equal causation. If two factors are correlated, even if that correlation is relatively high, it does not follow that one is causing the other. For example, if a research project found that students who spend more hours studying tend to get higher grades, it would be an interesting correlation. However, that research would not prove that studying longer hours causes better grades. Other intervening factors, like the type of final examination used, are not accounted for in this simple correlation. As an egregious example of this point, consider that the mean age in the United States is rising (that is, people are living longer; thus, there are more older adults). Also, the incidence of human trafficking crime is increasing. While these two facts may be correlated, it would not follow those older adults are responsible for human trafficking! Instead, this simple correlation does not account for numerous social forces in play. It is essential to keep in mind that correlation does not equal causation.

1.2 Definition

A correlation is a number between -1.0 and +1.0, where 0.0 means no correlation between the two variables, and either +1.0 or -1.0 means a perfect correlation. A positive correlation means that as one variable increases, the other also increases. For example, as people age, they tend to weigh more, so a positive correlation between age and weight would be expected. On the other hand, a negative correlation means that as one variable increases, the other decreases. For example, as people age, they tend to run slower, so a negative correlation between age and running speed would be expected. Here are the verbal descriptions of a correlation's value.

Correlation	Description
+.70 or higher	Very strong positive
+.40 to +.69	Strong positive
+.30 to +.39	Moderate positive
+.20 to +.29	Weak positive
+.19 to -.19	None or negligible
-.20 to -.29	Weak negative
-.30 to -.39	Moderate negative
-.40 to -.69	Strong negative
-.70 or less	Very strong negative

2 Pearson's r

Pearson's Product-Moment Correlation Coefficient (typically called Pearson's r) measures the strength of the relationship between two variables having continuous data that is normally distributed (they have bell-shaped curves). (Note: Lab 1 contains information about various data types). The following examples of correlation using Pearson's r are from the *mtcars*[1] data frame.

Variables	Correlation	Description
disp-mpg	-0.8476	Very Strong Negative
wt-mpg	-0.8677	Very Strong Negative
wt-qsec	-0.1747	No Correlation
disp-qsec	-0.4337	Strong Negative
drat-qsec	+0.0912	No Correlation

2.1 Demonstration: Pearson's r

Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Pearson's r
2  cor.test(airquality$Wind, airquality$Ozone,
3    method = "pearson")
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: This is the start of the cor.test function, which calculates the correlation between two variables from the *airquality*[2] data frame. That function requires the x-axis variable to be listed first, then the y-axis.

Line 3: This continuation of the cor.test function specifies Pearson's r as the method. Since Pearson's r is the default method for the cor.test function, this line did not need to be included, but it is used in this example since the "method" specification is essential in later lab examples.

Following is the result of the script.

```
        Pearson's product-moment correlation

data:  airquality$Wind and airquality$Ozone
t = -8.0401, df = 114, p-value = 9.272e-13
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.7063918 -0.4708713
sample estimates:
       cor
-0.6015465
```

The cor.test function returns information that is not useful at this point but will be in later tutorials.

1. *Pearson's product-moment correlation*: This is the function's title.
2. *data:* This line lists the two variables being correlated.
3. *t=:* This line is the calculated result of the cor.test function.
    a. The "t" score is used to calculate the p-value at the end of the line.
    b. The "df" is the "degrees of freedom" and measures how many different levels the variables can take.

    c. The "p-value" is the probability value; typically, a p-value less than 0.05 is considered significant. (Significance and p-value are discussed later in this lab.)
4. *Alternative hypothesis:* A statement of the alternative hypothesis being tested. The default is that the correlation is not equal to zero. This line states the alternative hypothesis so the researcher can compare that hypothesis with the correlation and p-value to see if the null hypothesis can be rejected. ("Hypothesis" is discussed in Lab 9, so this line can be ignored for now.)
5. *95% confidence interval:* The actual correlation may differ from that calculated due to extraneous factors not considered. This line shows the 95% confidence level boundaries for the actual correlation. In this case, the actual correlation should be between -0.706 and -0.471.
6. *Sample estimates:* This begins the "estimates" section of the report.
7. *cor:* This verifies that the test executed was Pearson's r (Spearman's will report "rho" and Kendall's will report "tau").
8. *-0.6015465:* This is the calculated correlation between the two variables and is the value needed.

2.2 Activity: Pearson's r

Using the *CO2*[3] data frame, determine the correlation between *CO2$conc* and *CO2$uptake*. Include the correlation in the deliverable document for this lab. Only the correlation number should be reported, not the entire ten results lines.

3 Categorical Data

When one or both data elements are categorical, Spearman's rho or Kendall's tau is used for calculating the correlation. Besides the process used, the concept is the same as Pearson's r. Spearman's rho is used when at least one variable is ordered data and typically involves larger data samples. Kendall's tau can be used for categorical data but is more accurate for smaller data samples. (Note: "ordered data" has categories that imply some order, but the interval between the categories cannot be calculated. For example, if a survey included an item about respondents' college class (Freshman, Sophomore, Junior, Senior), an order is implied (Sophomore comes after Freshman). However, there is no clearly defined interval among the categories.)

For example, imagine a research project attempting to determine movie preference by a person's age. The researcher could ask a large group of people to indicate their age and how well they liked certain types of movies by giving each a "star" rating. Movie types rated five-star are somehow better than those rated four-star, but it is impossible to quantify that difference in any meaningful way. Spearman's rho would be used to calculate a correlation between age and movie rating. Suppose that correlation reached -0.632 (this is a made-up number) for "horror" movies. In that case, the researcher could conclude a negative relationship between age and preference for horror movies. As people age, they tend not to prefer horror movies.

On the other hand, imagine that a data frame included information about the age of people who purchased various makes of automobiles. If the "makes" are selected from a list (like Ford, Chevrolet, or Honda,) then the data is categorical with no implied order; that is, "Ford" is neither better nor worse than "Chevrolet," just different. Kendall's tau would calculate the correlation between the customer's age and preference for automobile make. Perhaps the correlation would reach +0.534 (this is a made-up number). This result would indicate a strong positive correlation between these two variables; that is, people tend to prefer a specific make based upon their age; or, to put it another way, as people age, their preference for automobiles changes predictably.

The following examples are from the *mtcars* data frame, and since they all involve ordered data, Spearman's rho was used to calculate the correlations.

Variables	Correlation	Description
cyl—gear	-0.5643	Strong Negative
gear—am	+0.8077	Very Strong Positive
cyl—carb	+0.5801	Strong Positive
carb—gear	+0.1149	No Correlation
vs—carb	-0.6337	Strong Negative

3.1 Demonstration: Spearman's rho

Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Spearman's rho
2  cor.test(as.numeric(esoph$agegp), esoph$ncases,
3    method = "spearman")
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: This is the start of the cor.test function, which calculates the correlation between two variables from the *esoph*[4] data frame. That function requires the x-axis variable to be listed first, then the y-axis. Also, note that *esoph$agegp* is inside an *as.numeric* function. Since *agegp* uses text like "25-34" instead of a number, this converts that text to a number for the calculation.

Line 3: This specifies Spearman's rho as the correlation method.

Following is the result of the script.

```
Spearman's rank correlation rho

data:  as.numeric(esoph$agegp) and esoph$ncases
S = 57515, p-value = 1.029e-06
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.4935437

Warning message:
In cor.test.default(as.numeric(esoph$agegp), esoph$ncases, method =
      "spearman") :
  Cannot compute exact p-value with ties
```

The interpretation of Spearman's rho results is like that for Pearson's r and will not be further explained here. The script also generated a warning about p-values, which can be ignored for this lab.

3.2 Activity: Spearman's rho

Using the *CO2* data frame, determine Spearman's rho correlation between *CO2$Plant* and *CO2$uptake*. Include the correlation in the deliverable document for this lab. Only the correlation number should be reported, not the entire result.

3.3 Demonstration: Kendall's tau

Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Kendall's tau
2  cor.test(as.numeric(npk$N), npk$yield,
3    method = "kendall")
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: This is the start of the cor.test function, which calculates the correlation between two variables from the *npk*[5] data frame. That function requires the x-axis variable to be listed first, then the y-axis. Also, the *npk$N* is inside an *as.numeric* function since that variable must be converted from text to a number.

Line 3: This specifies Spearman's rho as the correlation method.

Following is the result of the script.

```
Kendall's rank correlation tau

data:  as.numeric(npk$N) and npk$yield
z = 2.3687, p-value = 0.01785
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau
0.4135721

Warning message:
In cor.test.default(as.numeric(npk$N), npk$yield, method =
      "kendall") :
  Cannot compute exact p-value with ties
```

Interpreting Kendall's tau is like Pearson's r and will not be discussed further.

3.4 Activity: Kendall's tau

Using the *CO2* data frame, determine Kendall's tau correlation between *CO2$Type* and *CO2$uptake*. Include the correlation in the deliverable document for this lab. Only the correlation number should be reported, not the entire result.

4 Selecting a Correlation Test

Pearson's r, Spearman's rho, and Kendall's tau all calculate correlation, and it is reasonable to wonder which method should be used in any given situation. Here is a quick chart to help.

Correlation	Data
Pearson's r	both data items being correlated are continuous.
Spearman's rho	at least one variable is ordered, and the sample size is large.
Kendall's tau	at least one variable is categorical (but not necessarily ordered), and the sample size is small.

Imagine a survey where college students were asked to check a box for their class (freshman-sophomore-junior-senior) and enter their age. Spearman's rho would be used to correlate these two data items since the class is ordered ("senior" comes after "junior") and age is continuous.

5 Significance

Most people use the word significant to mean important, but researchers and statisticians have a much different meaning for the word significant, and it is vital to keep that difference in mind.

In statistics and research, significance means that the experimental results would not likely have been produced by chance. For example, if a coin is flipped 100 times, heads should come up 50 times. Of course, it would be possible for heads to come up 55 or even 60 times by chance. However, if heads came up 100 times, researchers would suspect something unusual was happening (and they would be right!). To a researcher, the central question of significance is, "How many times can heads come up and still be considered chance?"

Researchers use one of three significance levels: 1%, 5%, or 10%. A researcher conducting The Great Coin-Tossing Experiment may start by stating, "This result will be significant at the 5% level." That would mean that if the coin were tossed 100 times and the number of "heads" tosses was between 47.5-52.5, a 5% spread, it would be considered chance. However, 47 or 53 "heads" would be significant and be outside that 5% spread.

It must seem somewhat subjective for a researcher to select the desired significance level. However, many researchers in business and the social and behavioral sciences (like education, sociology, and psychology) choose a significance level of 5%. There is no real reason for choosing a 5% level other than how things have traditionally been done for many years. Therefore, if a researcher selected something other than 5%, peer researchers would want some explanation concerning the "weird" significance level.

The calculated significance is typically reported as a p-value (for "probability value"). The following table contains the correlation and p-value for several pairs of variables from the *mtcars* data frame.

Variables	Correlation	P-value
wt—qsec	-0.1747	0.3389
am—hp	-0.2432	0.1798
hp—drat	-0.4488	0.009989
cyl—vs	-0.8108	1.843 x 10-08
wt—disp	+0.8880	1.222 x 10-11

If a 5% significance level were specified for this data, then any p-value smaller than 0.05 is considered significant; the observed relationship is not likely due to chance. There is no significance in the correlation between *wt—qsec* and *am—hp* since the p-values for those correlations are greater than 0.05. However, the correlations between the other variables are significant since the p-values for those are smaller than 0.05.

5.1 Demonstration: Significance

Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Significance
2  cor.test(airquality$Wind, airquality$Temp,
3    method = "pearson")
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: This is the start of the cor.test function, which calculates the correlation between two variables from the *airquality* data frame. That function requires the x-axis variable to be listed first, then the y-axis.

Line 3: This specifies Pearson's r as the correlation method.

Following is the result of the script.

```
Pearson's product-moment correlation

data:  airquality$Wind and airquality$Temp
t = -6.3308, df = 151, p-value = 2.642e-09
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.5748874 -0.3227660
sample estimates:
       cor
-0.4579879
```

At the end of line two of the results (not counting the title line), the p-value is reported at 2.642e-09, which is how R reports 2.642 × 10-09, which is far smaller than 0.05. This correlation, then, would be considered statistically significant.

5.2 Activity: Significance

Using the *CO2* data frame, determine the p-value between *CO2$conc* and *CO2$uptake*. Since these are continuous data, Pearson's r should be the selected method. Include the p-value in the deliverable document for this lab. Only the p-value should be reported, not the entire result.

6 Deliverable

Complete the activities in this lab and consolidate the responses into a single document. Name the document with your name and "Lab 7," like "George Self Lab 7," and submit that document for grading.

---

[1] The mtcars data frame was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models). The data frame includes these variables: mtcars$mpg, mtcars$cyl, mtcars$disp, mtcars$hp, mtcars$drat, mtcars$wt, mtcars$qsec, mtcars$vs, mtcars$am, mtcars$gear, and mtcars$carb.

[2] The airquality data frame contains information from New York air quality measurements. The data frame includes these variables: airquality$Ozone, airquality$Solar.R, airquality$Wind, airqualtiy$Temp, airquality$Month, and airquality$Day.

[3] The CO2 data frame contains information about carbon dioxide uptake in grass plants from an experiment on the cold tolerance of the grass species Echinochloa crusgalli. The data frame includes these variables: CO2$Plant, CO2$Type, CO2$Treatment, CO2$conc, and CO2$uptake.

[4] The esoph data frame contains information from a case-control study of (o)esophageal cancer in Ille-et-Vilaine, France. The data frame includes these variables: esoph$agegp, esoph$alcgp, esoph$tobgp, esoph$ncases, and esoph$ncontrols.

[5] The npk data frame contains the result of a classical N, P, K (nitrogen, phosphate, potassium) experiment on the growth of peas conducted on 6 blocks. The data frame includes these variables: npk$block, npk$N, npkP, npkK, and npk$yield.

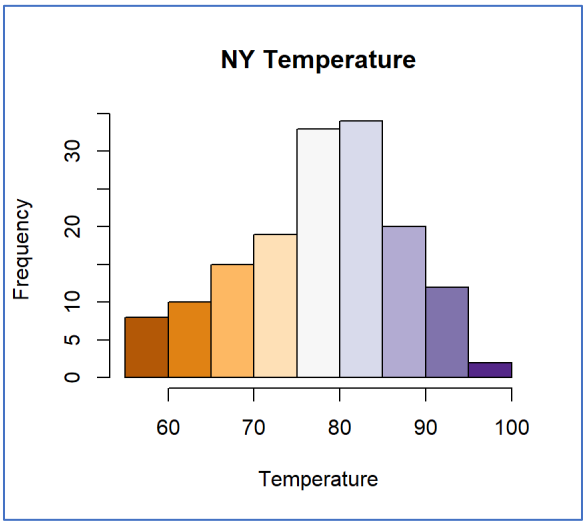
Lab 8: Visualizing Data

1 Introduction

Visualizing data frames rather than listing values in a table is often helpful. Visualizations can make subtle relationships evident that could be missed in a data table. Earlier labs described visualization techniques for descriptive and frequency data, and this tutorial applies the same techniques to continuous data.

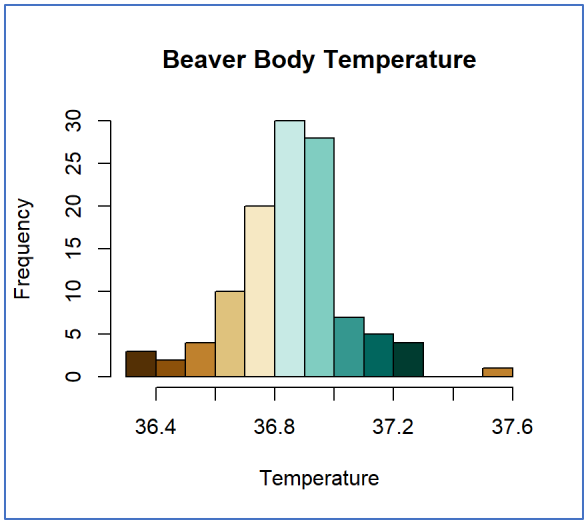
2 Histogram

A histogram is a graph that shows the distribution of continuous data, for example, age or height. A histogram resembles a bar chart, but there is a significant difference: a histogram is used for continuous data while a bar chart is used for categorical data. Thus, histograms are drawn with no space between the bars (the data is continuous along the x-axis). In contrast, bar charts are customarily drawn with a small space between bars (the data is categorical along the x-axis). As an example of a histogram, the following figure shows New York City's high temperature from May to September 1973 from the *airquality*[1] data frame.

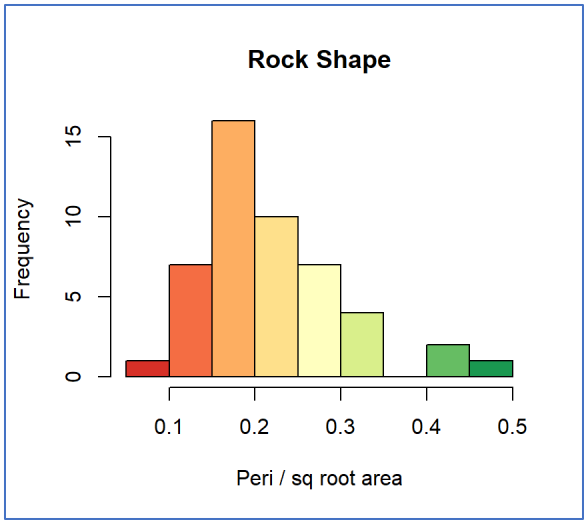


Notice there is no separate bar for each temperature; instead, R has clustered five temperatures into a single bar. Thus, there is a bar that combines the temperatures 70-74 and no separate bars for each of those temperatures. This histogram helps researchers determine if the temperature is normally distributed and whether it displays a typical "bell" curve. It should be evident from this graph that there are more temperatures around the 80° level than at the extremes, so this histogram does indicate that the data is normally distributed.

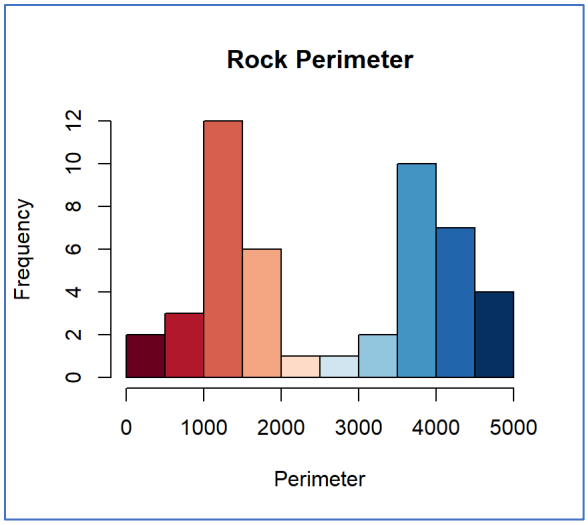
As another example, the following figure shows a histogram for the body temperature of a beaver recorded every 10 minutes over several hours, as found in the *beaver1*[2] data frame. This histogram indicates normally distributed data though there is an outlier at a temperature of 37.6°.



Histograms can also indicate skewed data, an important observation for researchers during a project's exploratory phase. Consider, for example, the following figure: the shape of petroleum rock samples in the *rock*[3] data frame. While this histogram indicates a normal distribution with levels falling off from a peak, there is a longer "tail" to the right, so the data has a positive skew.



Finally, consider the histogram in the following figure, taken from the *peri* variable in the *rock* data frame. This histogram shows a bimodal distribution with two clear peaks in the data. Researchers must know that the data is bimodal before analysis since having two modes can cause specific statistical tests to fail.



2.1 Demonstration: Histogram

To generate a histogram for a variable, use R's hist function. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Histogram
2  hist(morley$Speed,
3    main = "Morley's Experiment",
4    xlab = "Speed",
5    ylab = "Frequency",
6    breaks = 8,
7    col = cm.colors(10)
8  )
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: This is the beginning of the histogram function (it ends on Line 8). For this histogram, the *Speed* variable from the *morley*[4] data frame is specified as the data source for the histogram.

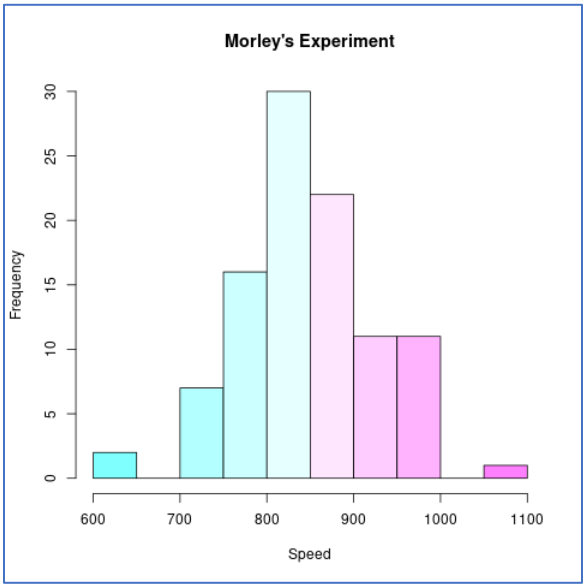
Lines 3-5: Specify the titles of the main, x-axis, and y-axis.

Line 6: To create a histogram, R analyzes the values in a variable and creates "bins" for those values, so many will be grouped for analysis. The "breaks" parameter tells R how many breaks to allow in the variable. In this case, eight breaks are specified, which would create nine bins. R will analyze the data, use the "breaks" parameter as a "suggestion," and only use that number of breaks if it makes sense for the graphed data. Changing the number of breaks by just one or two will not change the histogram produced, so researchers should play around with the "breaks" number to get the best possible representation of the data.

Line 7: This specifies that ten colors will be used from the "cm.color" palette to shade the various bars in the histogram. Researchers need to experiment with the color palette and number of colors to get the best result; however, "cm.color" and the number of bars in the histogram seem to work well.

Line 8: This parenthesis closes the hist function started on line 2.

Following is the result of the script.



2.2 Activity: Histogram

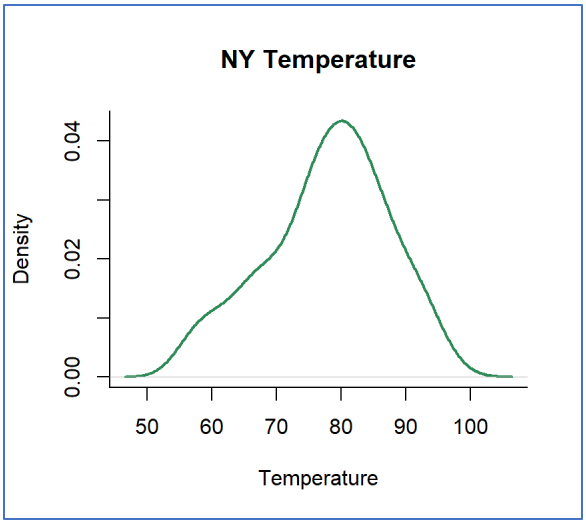
Using the *USJudgeRatings*[5] data frame, create a histogram of the *INTG* variable. The histogram should meet these specifications:

- Title: US Judge Ratings
- X-axis label: Integrity Rating
- Y-axis label: Count
- Breaks: 11
- Color: eight colors from the cm.colors palette

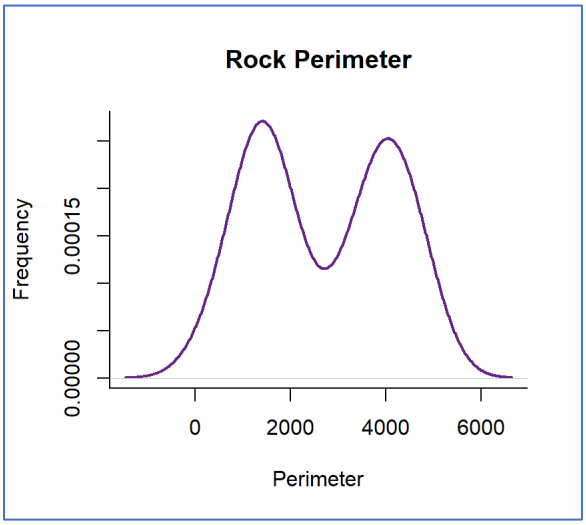
Include the histogram in the deliverable document for this lab.

3 Density Plot

A density plot provides the same information as a histogram but is smoothed out, making it easier to read. The same NY City temperature data from an earlier histogram is drawn as the following density plot.



Given the previous plot, it is natural to wonder, "What is density?" It is a calculated value such that the total area under the curve is assumed to be one. Then each point along the x-axis is calculated to contribute the correct proportional amount to that total density. It is adequate to consider a density plot a smoothed histogram for many purposes. As just one other example, the following is a density plot of the bimodal histogram presented earlier. The density plot makes the bimodal nature of the data very evident.



3.1 Demonstration: Density Plot

To generate a density plot for a variable, use R's plot and density functions. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Density plot of US Arrests for rape.
2  plot(density(USArrests$Rape),
3    main = "US Rape Arrests",
4    xlab = "Arrests",
5    ylab = "Density",
6    lwd = 2,
7    col = "magenta4"
8  )
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: Create a density plot for the Rape variable in the *USArrests*[6] data frame.

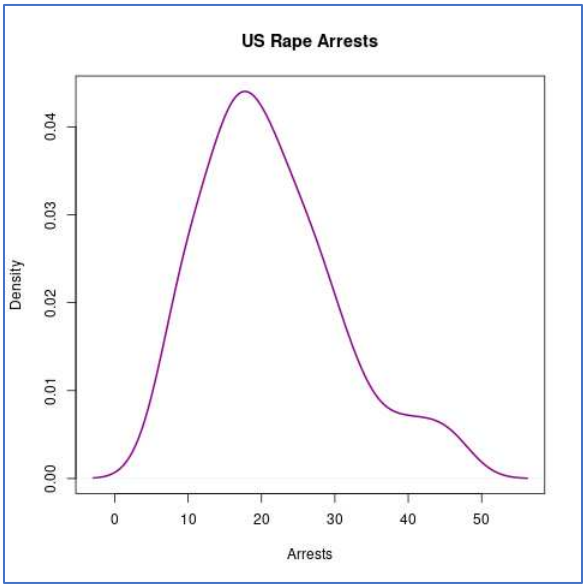
Lines 3-5: Specify the titles of the main, x-axis, and y-axis.

Line 6: The line type. "lwd 2" is a relatively thick line that is easy to see on a graph.

Line 7: The color for this graph is "magenta4"

Line 8: This parenthesis closes the plot function started on line 2.

Following is the result of the script.



3.2 Activity: Density Plot

Using the *USJudgeRatings* data frame, create a density plot of the *INTG* variable. The plot should meet these specifications:

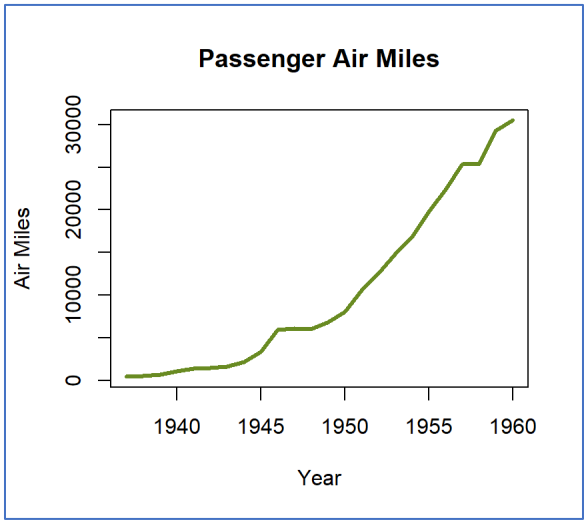
- Title: US Judge Ratings
- X-axis label: Integrity Rating
- Y-axis label: Density
- lwd: 2

- Color: blue4

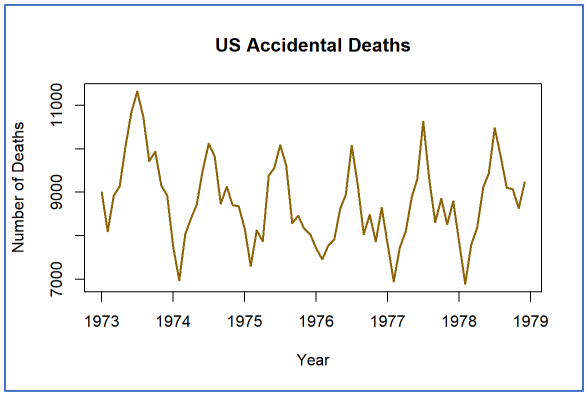
Include the density plot in the deliverable document for this lab.

4 Line Graph

Line graphs display the frequency of some value in a linear form that makes trend detection easier. These graphs are handy with "time series" data, that is, data gathered over a long period. For example, consider the following from the *airmiles*[7] data frame, which charts the number of passenger miles on US commercial airlines from 1937 to 1960.

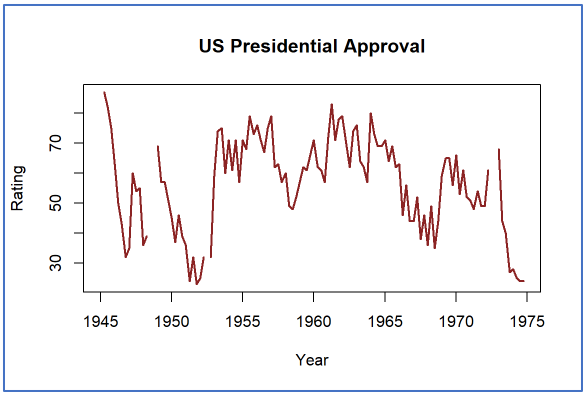


The following figure shows the number of accidental deaths in the United States from 1973 until 1979 by month taken from the *USAccDeaths*[8] data frame. This line graph clearly shows a seasonal difference where there are more accidental deaths in the summer months than winter, and detecting this type of variation is one of the strengths of a line graph.



As one final example, the following figure shows the approval rating for US Presidents from 1945 until 1975 taken from the *presidents*[9] data frame. This line graph shows a very high approval rating in 1945 (Roosevelt at the end of WWII) with dips in the early 1950s (Truman and the Korean Conflict) and about 1974 (Watergate

and Nixon's resignation). Notice that missing data in the data frame causes two gaps in the line (the late 1940s and 1973).



4.1 Demonstration: Line Graph

Using R's plot function, generate a line graph for a variable. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Line Graph
2  plot(ldeaths,
3    main = "Lung Disease Deaths",
4    xlab = "Month",
5    ylab = "Number",
6    type="l",
7    lwd = 2,
8    col = "red"
9  )
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: The plot function processes the *ldeaths*[10] data frame. Since this data frame only contains a time series, there is no need to indicate a variable name.

Lines 3-5: Specify the main title and labels for the x-axis and y-axis.

Line 6: This specifies the type of line graph wanted. The possible values are:

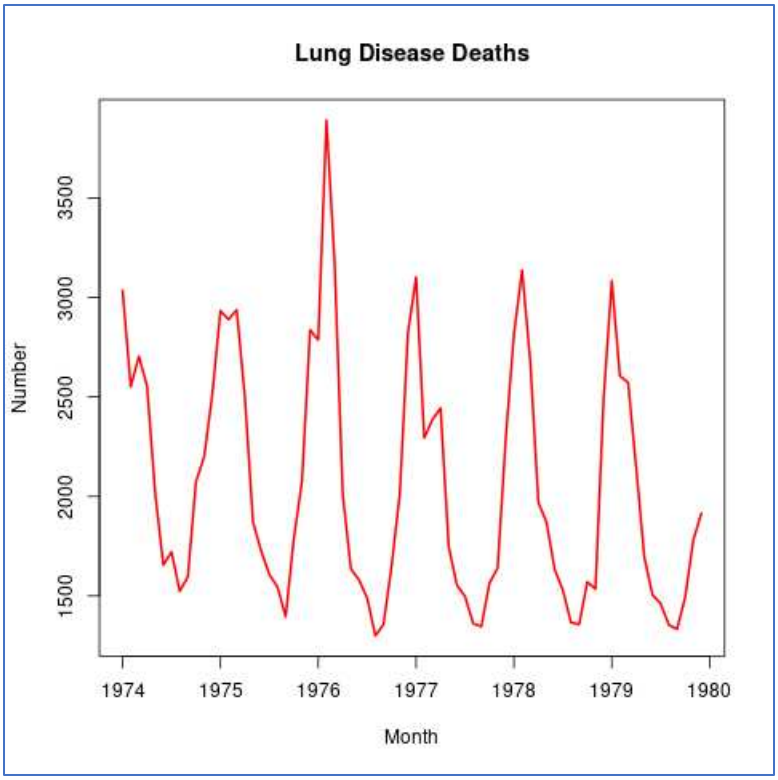
- p for points
- l for lines (this is a lower-case "L")
- b for both
- c for the lines part alone of "b"
- for both' overplotted' (that is, a lower-case "O," not zero)
- h for 'histogram-like' (or 'high-density') vertical lines
- s for stair steps (that is, a lower-case "S")
- S for other steps (that is, a capital "S")
- n for no plotting

Line 7: The lwd parameter sets the width of the line. The default value is one, so this line specifies a double-width line to make it easier to see.

Line 8: This sets the line's color to red to set it off from the axis and text.

Line 9: This parenthesis closes the plot function started on line 2.

Following is the result of the script. Line graphs are handy for detecting a change in variables, especially over time, indicating that there are more deaths in the winter months than in summer.



4.2 Activity: Line Graph

Using the *UKgas*[11] data frame, create a line graph with these specifications:

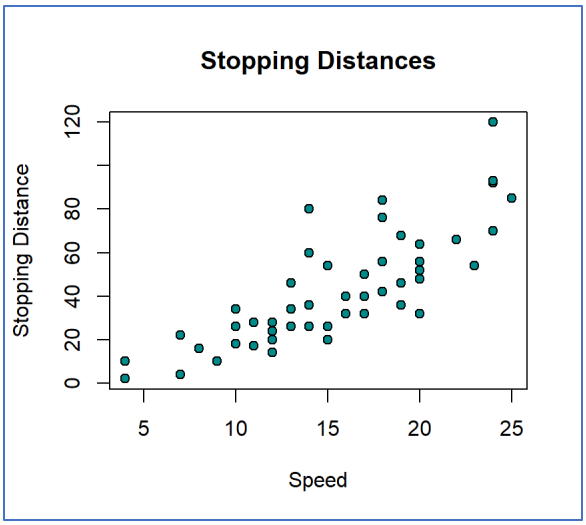
- Plot: UKgas
- Title: UK Quarterly Gas Consumption
- X-axis label: Year
- Y-axis label: Millions of Therms
- Type: l (this is a lower-case "L")
- lwd: 2
- Color: firebrick4

Include the line graph in the deliverable document for this lab.

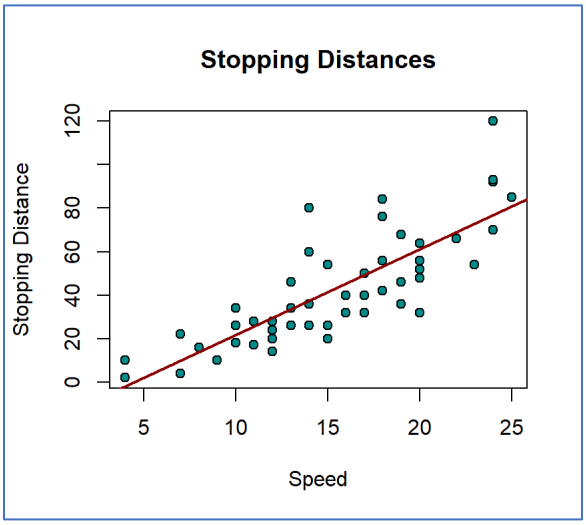
5 Plot

5.1 Introduction

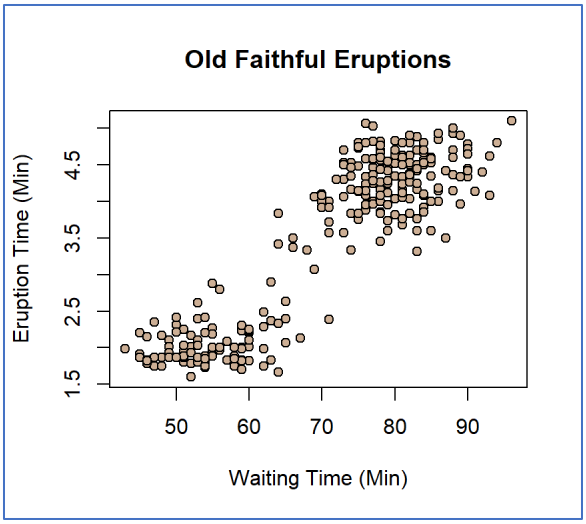
Plots (often called "scatter plots") show how two different variables are related. Scatter plots are often used in connection with correlation, where they visually indicate the correlation between two variables. For example, the following figure is the plot of stopping distance vs. speed from the *cars*[12] data frame.



The previous figure shows that as a car's speed increases, the stopping distance also increases, which would be expected. (Note: this data was gathered on cars in the 1920s.) A line of best fit is often included with a plot to visualize the relationship between the two variables better, as illustrated in the following figure.



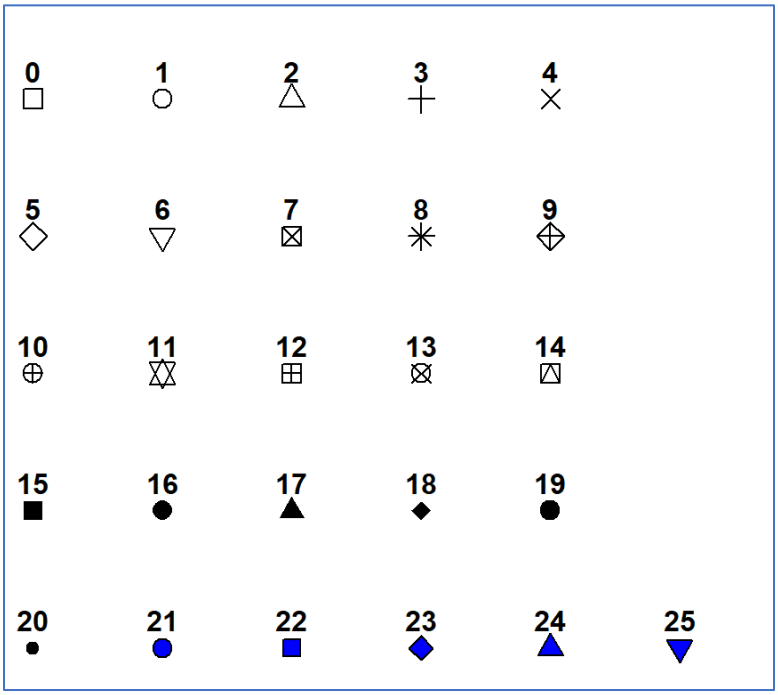
The following figure is a plot created from the faithful[13] data frame as a second example. It shows the eruption time for the Old Faithful geyser as a function of the waiting time between eruptions.



The previous figure shows that as the time between eruptions increases, the time that the eruption lasts also increases. Notice that this scatter plot also suggests that the data is bimodal since there are two clusters of points, and a researcher would want to explore that matter before doing much else with the data.

5.2 Plot Point Types

R has 25 different symbols that could be used for the points on a plot, as shown in the following chart.



Symbols 1-20 are a single color, but symbols 21-25 have two colors, and the background and line color can be specified. Using black for the line color (the default) and background color with high contrast to the plot area makes the points easier to see on a graph. The point symbol used in a plot is specified in the *pch=* attribute.

5.3 Demonstration: Plot

To generate a plot for a variable, use R's plot function. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Simple Plot
2  plot(swiss$Fertility ~ swiss$Education,
3    main = "Swiss Indicators",
4    xlab = "Education",
5    ylab = "Fertility",
6    pch = 21,
7    bg = "chartreuse3",
8    col = "black"
9  )
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: This starts the plot function. Like many R functions, a plot requires a formula input as y ~ x. The dependent variable (the y-axis) is first in the formula, and the independent variable (the x-axis) is second. For this plot, two variables from the *swiss*[14] data frame are used. Education is the independent variable on the x-axis, while Fertility is the dependent variable on the y-axis. The researcher answered, "Does education level affect the number of children people have?"

Lines 3-5: Specify the title and labels for the x-axis and y-axis.

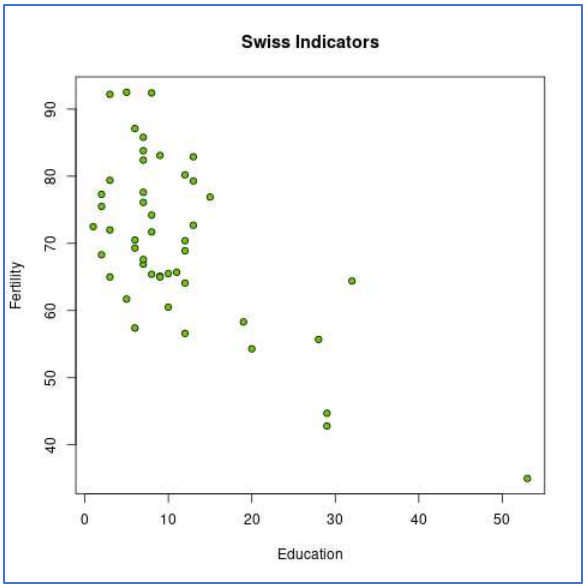
Line 6: The *pch* attribute is the type of point used on the graph. As illustrated above, R has several pre-defined types of points, and for this graph, type number 21 is selected.

Line 7: The background color of the points is specified using the *bg* attribute. In this case, it is a shade of chartreuse, a yellow-green hue.

Line 8: The line color of the circle for each point is specified using the col attribute.

Line 9: This parenthesis closes the plot function started on line 2.

Following is the result of the script.



A line of best fit is often included in a plot to visualize the relationship between the variables better. R's *abline* function generates a line of best fit. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Line of Best Fit
2  plot(attitude$complaints ~ attitude$rating,
3    main = "Attitude Data",
4    xlab = "Overall Rating",
5    ylab = "Complaints",
6    pch = 21,
7    bg = "khaki2",
8    col = "black"
9  )
10 abline(lm(attitude$complaints ~ attitude$rating),
11   lwd = 2,
12   col = "darkseagreen4"
13 )
```

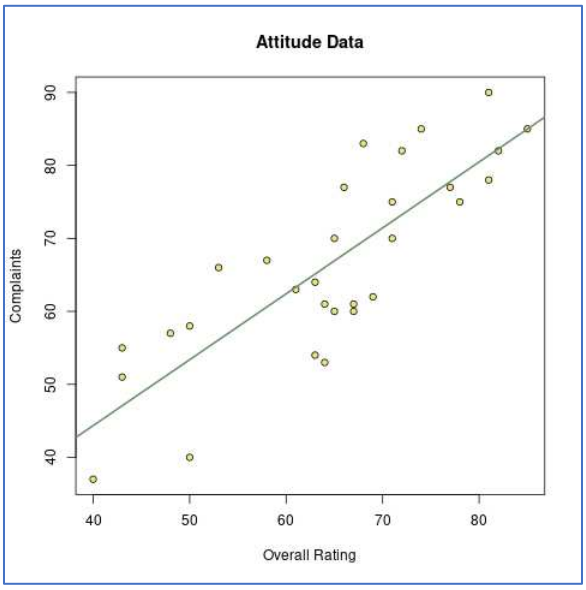
Lines 1-9: This demonstration plots two variables from the *attitude*[15] data frame. R*ating* is the independent variable on the x-axis, and *complaints* is the dependent variable on the y-axis. The researcher answered, "Do employees with higher overall ratings handle complaints better?" The other parameters of this plot function are defined for the previous plot and are not further discussed here.

Line 10: This starts the *abline* function that ends on line 13. *Abline* draws a line "from point A to point B" (which is why the function is named "abline") on an existing plot. In this case, the line drawn is calculated with the *lm* (linear model) function, which calculates the slope and y-intercept of the line of best fit for the two specified variables.

Lines 11-12: These are the parameters for the line of best fit as drawn on the plot.

Line 13: This parenthesis closes the abline function started on line 10.

Following is the result of the script.



5.4 Activity: Plot

Using the *rock* data frame, create a plot with *rock$area* (y-axis) as a function of *rock$peri* (x-axis). The graph should include a line of best fit and meet the following specifications.

Plot:

- Title: Rock Area by Perimeter
- X-axis label: Perimeter
- Y-axis label: Area
- Pch: 21
- Bg: goldenrod3
- Color: black

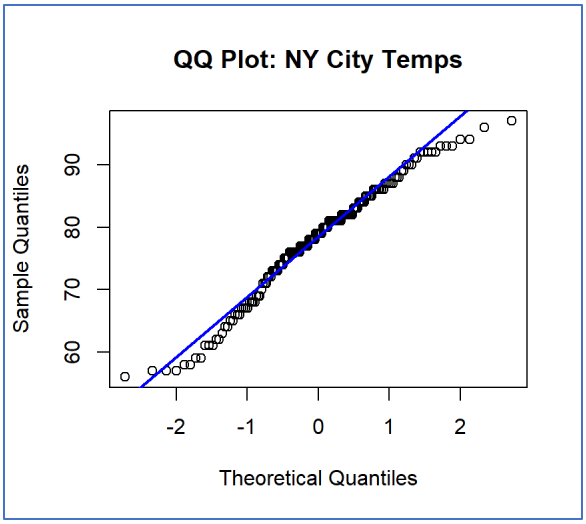
Abline:

- lwd: 2
- Color: violetred3

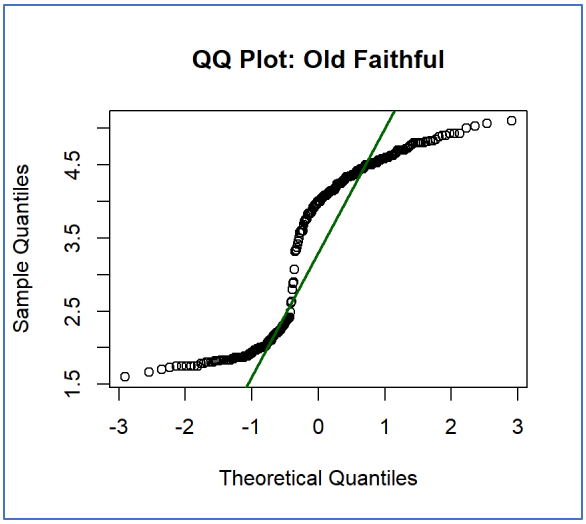
Include the plot in the deliverable document for this lab.

6 Q-Q Plot

Researchers in the exploratory phase of a project often need to know whether a data frame is normally distributed. Creating a histogram or density plot is very helpful, but a Q-Q ("Quantile-Quantile") Plot is a typical method to determine if a data frame is normally distributed. The following figure is a Q-Q plot of the New York City temperatures in the *airquality* data frame.

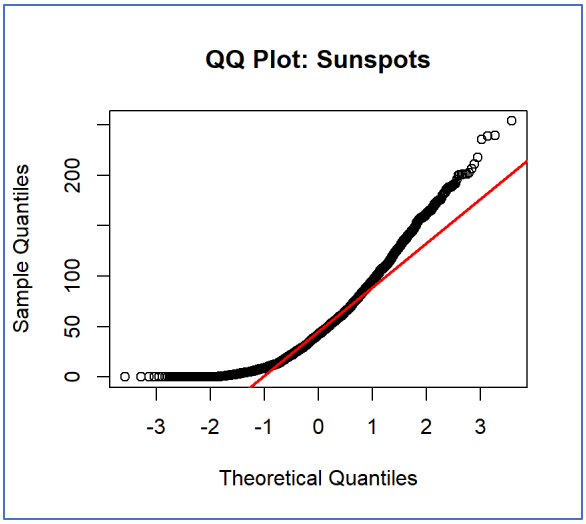


A perfect normal distribution would generate a straight-line Q-Q plot, and the blue line in the previous figure is ideal. Interpreting a Q-Q plot is more art than science, but the data is normally distributed if most values are near the ideal line. As plotted earlier in this tutorial, the following figure is a Q-Q plot for the Old Faithful eruption.



The above plot shows a typical bi-modal pattern. On the left side of the plot is a reasonably flat area from -3 to -0.5 on the x-axis. The plot skips upward and creates a second reasonably flat area between 0.5 and 3. Imagine two parallel lines running through the lower and upper parts of the plot and then notice that the green "ideal" line does not get very close to the slope of either of those lines. This Q-Q plot shows that the data is not normally distributed.

The following figure shows a curved Q-Q plot that is typical for a data frame that is skewed. In this case, the plotted *sunspots*[16] data has a heavy positive skew indicated by the long "tail" on the left side of the plot. That skew should be verified with a histogram or density plot.



6.1 Demonstration: Q-Q Plot

To generate a Q-Q plot for a variable, use R's qqnorm and qqline functions. Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # Q-Q Plot
2  qqnorm(chickwts$weight,
3    main = "QQ Plot: Chick Weights"
4  )
5  # Q-Q Line
6  qqline(chickwts$weight,
7    lwd = 2,
8    col = "blue"
9  )
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

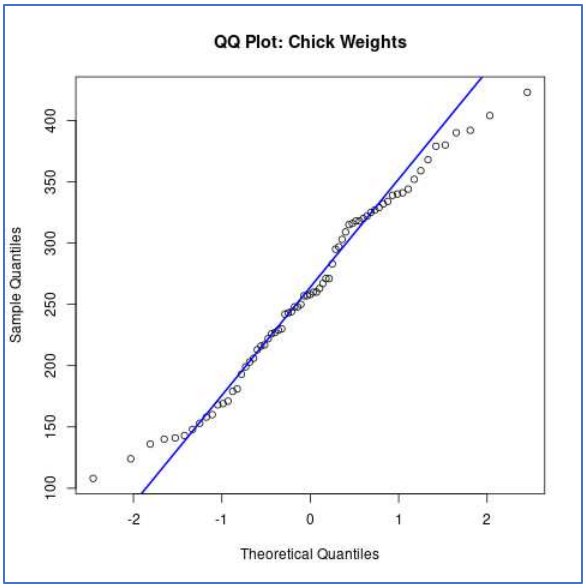
Lines 2-4: This executes the qqnorm function and passes that function the weight variable from the *chickwts*[17] data frame. This function draws the Q-Q Plot. The only parameter needed is main, which adds the title to the plot.

Line 5: This parenthesis closes the qqnorm function started on line 2.

Lines 6-9: This executes the qqline function and passes that function the weight variable from the *chickwts* data frame. This function draws the straight line that indicates a perfect Q-Q plot. The only parameters passed to the function are setting the size to 2 and the color to blue.

Line 9: This parenthesis closes the qqline function started on line 6.

Following is the result of the script.



6.2 Activity: Q-Q Plot

Using the *swiss* data frame, create a Q-Q plot of the *Fertility* variable. The plot should have the title "Q-Q Plot: Swiss Fertility" and a Q-Q line with a width of 2 and blue color. Include the plot in the deliverable document for this lab.

7 Deliverable

Complete the activities in this lab and consolidate the responses into a single document. Name the document with your name and "Lab 8," like "George Self Lab 8," and submit that document for grading.

---

[1] The airquality data frame contains information from New York air quality measurements. The data frame includes these variables: airquality$Ozone, airquality$Solar.R, airquality$Wind, airqualtiy$Temp, airquality$Month, and airquality$Day.

[2] The beaver data frame reports the body temperature of two Castor canadensis beavers in north-central Wisconsin. The data frame includes these variables: beaver1$day, beaver1$time, beaver1$temp, and beaver1$activ.

[3] The rock data frame provides measurements on 48 petroleum Rock samples. The data frame includes these variables: rock$area, rock$peri, rock$shape, and rock$perm.

[4] The Morley data frame includes measurements on the speed of light done in 1879. The data frame includes these variables: morley$Expt, morley$Run, and morley$Speed.

[5] The USJudgeRatings data frame contains lawyers' ratings of state judges in the US Superior Court. The data frame includes these variables: USJudgeRatings$CONT, USJudgeRatings$INTG, USJudgeRatings$DMNR, USJudgeRatings$DILG, USJudgeRatings$CFMG, USJudgeRatings$DECI, USJudgeRatings$PREP, USJudgeRatings$FAMI, USJudgeRatings$ORAL, USJudgeRatings$WRIT, USJudgeRatings$PHYS, and USJudgeRatings$RTEN.

[6] The USArrests data frame contains statistics about the arrests per 100,000 residents for assault, murder, and rape in each of the 50 US states in 1973. Also given is the percent of the population living in urban areas. The data frame includes these variables: USArrests$Murder, USArrests$Assault, USArrests$UrbanPop, and USArrests$Rape.

[7] The airmiles data frame contains the number of passenger miles flown by commercial airlines in the United States for each year from 1937 to 1960. The data frame is a time series and contains no other variables.
[8] The USAccDeaths data frame contains the monthly totals of accidental deaths in the USA. The data frame is a time series and contains no other variables.
[9] The presidents data frame contains the (approximate) quarterly approval rating for the President of the United States from the first quarter of 1945 to the last quarter of 1974. The data frame is a time series and contains no other variables.
[10] The ldeaths data frame contains the number of monthly deaths recorded from bronchitis, emphysema, and asthma in the UK from the years 1974 until 1979. The data frame is a time series and contains no other variables.
[11] The UKgas data frame contains the quarterly UK gas consumption from 1960Q1 to 1986Q4, in millions of therms. The data frame is a time series and contains no other variables.
[12] The cars data frame contains information about the speed and stopping distances for cars. Note that the data was recorded in the 1920s. The data frame includes these variables: cars$speed and cars$dist.
[13] The faithful data frame contains observations about the Old Faithful Geyser in Yellowstone National Park. The data frame includes these variables: faithful$waiting, the waiting time between eruptions, and faithful$eruptions, the eruption time in minutes.
[14] The swiss data frame contains data about Swiss fertility and socioeconomic indicators from 1888. The data frame includes these variables: swiss$Fertility, swiss$Agriculture, swiss$Examination, swiss$Education, swiss$Catholic, and swiss$Infant.Mortality.
[15] The attitude data frame contains information from a survey of the clerical employees of a large financial organization. The data frame includes these variables: attitude$ratings, attitude$complaints, attitude$privileges, attitude$learning, attitude$raises, attitude$critical, and attitude$advance.
[16] The sunspots data frame contains the monthly mean sunspot numbers from 1749 to 1983. The data was collected at Swiss Federal Observatory, Zurich until 1960, then Tokyo Astronomical Observatory. The data frame is a time series and contains no other variables.
[17] The chickwts data frame contains the results of an experiment conducted to measure and compare the effectiveness of various feed supplements on the growth rate of chickens. The data frame includes these variables: chickwts$weight and chickwts$feed.

Lab 9: Parametric Testing

1 Introduction

Researchers should always begin a project with a hypothesis and then gather data to see if the hypothesis supports an underlying theory. Continuous data gathered as part of the research project is analyzed using parametric techniques, and two of the most used tests are described in this lab. This lab starts with a discussion of "hypothesis" since that is key to both parametric and nonparametric testing.

2 Hypothesis

A hypothesis is an attempted explanation for some observation and is often used as a starting point for further investigation. For example, imagine that a physician notices that babies born to women who smoke seem to weigh less than women who do not. That could lead to a hypothesis: "smoking during pregnancy is linked to lighter birth weights." As another example, imagine that a restaurant owner notices that tipping seems higher on weekends than weekdays. That might lead to a hypothesis that "the size of tips is higher on weekends than weekdays." After creating a hypothesis, a researcher would gather data and then statistically analyze that data to determine if the hypothesis is valid. Additional investigation may be needed to explain why that observation is accurate. There are usually two related competing hypotheses in a research project: the Null Hypothesis and the Alternate Hypothesis.

- Null Hypothesis. This hypothesis is sometimes described as the "skeptical" view; the explanation for some observed phenomena was mistaken. For example, the null hypothesis for the smoking mother observation mentioned above would be "smoking does not affect a baby's weight." The tipping null hypothesis would be "there is no difference in tipping on the weekend."
- Alternate Hypothesis. This hypothesis is suggested as an explanation for the observed phenomenon. In the case of the mothers mentioned above, the alternative hypothesis is that smoking causes a decrease in birth weight. This hypothesis is called "alternate" because it differs from the status quo, encapsulated in the null hypothesis.

For the most part, researchers will never conclude that the alternate hypothesis is true. There are always confounding variables that are not considered but could cause the observation. For example, in the smoking mothers example mentioned above, even if the evidence indicates that babies born to smokers weigh less, the researcher could not conclusively state that smoking caused that observation. Perhaps non-smoking mothers had better health care; perhaps they had better diets, exercised more, or several other reasonable explanations unrelated to smoking. For that reason, the result of a research project is typically reported with one of two phrases:

- The null hypothesis is rejected. The null hypothesis will be rejected if the evidence indicates a significant difference between the status quo and whatever was observed. For the "tipping" example above, if the researcher found a significant difference in the amount of money tipped on weekends compared to weekdays, then the null hypothesis (tipping is the same on weekdays and weekends) would be rejected.

- The null hypothesis cannot be rejected. If the evidence indicates no significant difference between the status quo and whatever was observed, the researcher would report that the null hypothesis could not be rejected. For example, if there was no significant difference in the birth weights of babies born to smokers and non-smokers, then the researcher failed to reject the null hypothesis.

Often, a research hypothesis is based on a prediction rather than observation, and that hypothesis can be tested. Imagine a hypothesis: "walking one mile a day for one month decreases blood pressure." A researcher could test this by measuring the blood pressure of a group of volunteers, having them walk a mile every day for a month, and then measuring their blood pressure at the end of the experiment to look for a significant difference.

3 ANOVA

An Analysis of Variance (ANOVA) is used to analyze the difference in more than two groups of normally distributed samples. For example, imagine a professor testing a hypothesis that tutoring improves students' grades. A class is split into three groups: one group was not required to attend tutoring, a second group was required to attend tutoring once a week, and a third group was required to attend tutoring more than once a week. The null hypothesis is that "The amount of tutoring does not significantly change students' scores on the final exam." The alternate hypothesis is that "More frequent tutoring significantly changes students' scores on the final test." After the final exam was graded, an ANOVA could be administered. If the test scores for those three groups of students had a significant difference, then the null hypothesis would be rejected in favor of the alternate hypothesis.

3.1 Demonstration: ANOVA

The R ANOVA function requires the two variables being compared to be input in a linear model (lm) formula in the form of y ~ x, where y is the dependent variable (outcomes), and x is the independent variable (the groups producing the outcomes). Also, the data source is specified with a *data=* parameter. In the case of the professor's Tutoring Efficacy hypothesis mentioned in the previous paragraph, the students' final exam scores would be the dependent variable, and the three tutoring groups would be the independent variable.

Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # ANOVA
2  anova(lm(Speed ~ Expt, data = morley))
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: The ANOVA function is used to process the linear model (lm) generated by the *Speed* (outcomes) and *Expt* (groups) variables in the *morley*[1] data frame.

The ANOVA function returns much information, most of which is beyond the scope of this lab.

```
Analysis of Variance Table

Response: Speed
          Df Sum Sq Mean Sq F value    Pr(>F)
Expt       1  72581   72581  13.041 0.0004827 ***
Residuals 98 545444    5566
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

While an ANOVA function returns information that would be useful in a more thorough statistical analysis, this lab is only concerned with the p-value, 0.0004827, which is labeled *Pr(>F)* and is found near the end of line three (not counting the title). Following that p-value, R helpfully prints a code to aid in determining the significance of the result, three asterisks in this case. The last line in the results then lists the meaning of the significance codes used. *P*-values that fall between 0 and 0.001 are marked with three asterisks, as in this case, so it is significant at the 0.1% level (0.001), the most significant level.

3.2 Activity: ANOVA

Using the *npk*[2] data frame, calculate an ANOVA for the *yield* output when grouped by *block*. Include the *Pr(>F)* value in the deliverable document for this lab. Only the *Pr(>F)* value should be reported, not the entire result.

4 T-test

A t-test analyzes the difference between two groups of normally distributed samples, unlike an ANOVA, where more than two groups are compared. For example, imagine that the spending habits of two similar groups of people are compared; do the residents of Tucson spend more when dining out than the residents of Phoenix? The null hypothesis is, "People in Tucson and Phoenix spend the same when dining out." The alternate hypothesis is, "People in Tucson and Phoenix do not spend the same when dining out." Imagine that the dining bills of 100 people from both cities were recorded, and it was discovered that the mean bill in Phoenix is $15.13 and in Tucson is $12.47. The null hypothesis will be rejected if a t-test determines a significant difference in those two numbers.

4.1 Demonstration: T-test

A t.test requires the two variables being compared to be input as a formula in the form of y ~ x, where y is the dependent variable (outcomes), and x is the independent variable (the groups producing the outcomes). Also, the data source is specified with a *data=* parameter. In the case of the Dining Spending hypothesis mentioned in the previous paragraph, the size of the dining bills would be the dependent variable, the outcome, and the two groups of diners, the two cities, would be the independent variable.

Copy and paste the R code in the right-hand box below into the Snippets text box at https://rdrr.io/snippets/ and tap the "Run" button.

```
1  # T-Test (Independent)
2  t.test(extra ~ group, data = sleep)
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: The t.text function is used with the *sleep*[3] data to determine if there is a significant difference in the extra sleep between the two groups.

The t.test function returns much information, most of which is beyond the scope of this lab.

```
Welch Two Sample t-test

data:  extra by group
t = -1.8608, df = 17.776, p-value = 0.07939
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.3654832  0.2054832
sample estimates:
mean in group 1 mean in group 2
           0.75            2.33
```

While a t.test function returns information that would be useful in a more thorough statistical analysis, this lab is only concerned with the p-value, 0.07939, found at the end of line two (not counting the title line).

4.2 Activity: T-test

Using the *npk* data frame, calculate the t.test for the *yield* output when grouped by *N*. Include the p-value in the deliverable document for this lab. Note that only the p-value should be reported, not the entire result.

5 Deliverable

Complete the activities in this lab and consolidate the responses into a single document. Name the document with your name and "Lab 9," like "George Self Lab 9," and submit that document for grading.

---

[1] The Morley data frame includes measurements on the speed of light done in 1879. The data frame includes these variables: morley$Expt, morley$Run, and morley$Speed.

[2] The npk data frame contains the result of a classical N, P, K (nitrogen, phosphate, potassium) experiment on the growth of peas conducted on 6 blocks. The data frame includes these variables: npk$block, npk$N, npkP, npkK, and npk$yield.

[3] The sleep data frame contains data which shows the effect of two soporific drugs (increase in hours of sleep compared to control) on 10 patients. The data frame includes these three variables: sleep$extra, sleep$group, and sleep$ID.

Lab 10: Non-Parametric Testing

1 Introduction

Researchers often begin a project with a hypothesis and then gather data to see if the hypothesis supports an underlying theory. Categorical data gathered as part of the research project is analyzed using non-parametric techniques, and two of the most used tests are described in this lab. (Note: the concept of "hypothesis" is discussed in Lab 9.)

2 Kruskal-Wallis H

This test is used to determine if there are any significant differences in three or more data groups that are not normally distributed, often categorical. Imagine that a researcher wanted to determine if there was a difference in the smoking habit by age group. The subjects were interviewed and asked how many packs they smoked per week. This data was skewed to the right since a few subjects smoked heavily, but most were non-smokers or only smoked a few packs per week. The subjects were also divided into age groups: <20, 20-29, 30-39, 40-49, >49. The researcher would then use a Kruskal-Wallis H test to see if there was a significant difference in the smoking habit by age group since the dependent variable (packs smoked) was not normally distributed.

2.1 Demonstration: Kruskal-Wallis H

The R kruskal.test function requires the two variables being compared to be input in as y ~ x, where y is the dependent variable (outcomes), and x is the independent variable (the groups producing the outcomes). Also, the data source is specified with a *data=* parameter. In the case of the smoking example mentioned in the previous paragraph, the number of packs smoked would be the dependent variable, and the age group would be the independent variable.

```
1  # Kruskal-Wallis H
2  kruskal.test(Ozone ~ Month, data = airquality)
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: The kruskal.test function is used with the *airquality*[1] data to determine if there is a significant difference in the ozone readings between the measured months.

The kruskal.test function returns much information, most of which is beyond the scope of this lab.

```
        Kruskal-Wallis rank sum test

data:  Ozone by Month
Kruskal-Wallis chi-squared = 29.267, df = 4, p-value = 6.901e-06
```

While a kruskal.test function returns information that would be useful in a more thorough statistical analysis; this lab is only concerned with the p-value, 6.901e-06, which is found at the end of line two (not counting the title). Because this is less than 0.05 (5%), it would be considered a significant result. Thus, the null hypothesis would be rejected (there was no difference in Ozone by Month). Notice that this test does not indicate which

month had the most significant ozone reading or if all months had some significant variance from the mean, just that there is a significant difference between the months.

2.2 Activity: Kruskal-Wallis H

Using the *ChickWeight*[2] data frame, calculate kruskal.test for the *weight* output when grouped by *Diet*. Include the P-value in the deliverable document for this lab. Note that only the P-value should be reported, not the complete results.

3 Mann-Whitney U

This test determines any significant differences in two data groups that are not normally distributed, often categorical. Imagine that a movie producer wanted to know if there was a difference in how the audience in two different cities responded to a movie. The null hypothesis is, "There is no difference in movie-goers' opinions between these two cities." The alternate hypothesis is that "Movie-goers' opinions are significantly different by city." As the audience members left the theater, they would be asked to rate the movie on a scale of one to five stars. The ratings for the two cities would be collected, and then a Mann-Whitney test would be used to determine if the difference in ratings between the cities was significant.

3.1 Demonstration: Mann-Whitney U

R uses the wilcox.test for several different types of non-parametric tests. It will automatically compute a Mann-Whitney U test when the dependent variable is numeric, and the independent variable is binary (only two levels). The wilcox.test function requires the two variables being compared to be input in the form of y ~ x, where y is the dependent variable (outcomes), and x is the independent variable (the two groups producing the outcomes). Also, the data source is specified with a *data=* parameter. In the case of the movie analysis example mentioned in the previous paragraph, the count of each movie rating would be the dependent variable, the outcome, while the city would be the independent, or grouping, variable.

```
1  # Mann-Whitney U
2  wilcox.test(uptake ~ Treatment, data = CO2)
```

Line 1: Any line in R that starts with a hashtag is a comment and will be ignored.

Line 2: The wilcox.test function is used with the *CO2*[3] data to determine if there is a significant difference in the CO2 uptake for chilled or not chilled plants.

The wilcox.test function returns much information, most of which is beyond the scope of this lab.

```
        Wilcoxon rank sum test with continuity correction

data:  uptake by Treatment
W = 1187.5, p-value = 0.006358
alternative hypothesis: true location shift is not equal to 0

Warning message:
In wilcox.test.default(x = c(16, 30.4, 34.8, 37.2, 35.3, 39.2, 39.7,
      :
  cannot compute exact p-value with ties
```

The warning message can be ignored. It indicates that some values in the Treatment variable are repeated ("tied"), but that is expected since more than one plant in this study got the same treatment. While a wilcox.test function returns information that would be useful in a more thorough statistical analysis, this lab is only concerned with the p-value, 0.006358, found at the end of line two (not counting the title). Because this is less than 0.05 (5%), it would be considered a significant result.

3.2 Activity: Mann-Whitney U

Using the *mtcars*[4] data frame, calculate Mann-Whitney U for the *disp* output when grouped by *am*. Include the P-value in the deliverable document for this lab. Note that only the P-value should be reported, not the complete results.

4 Deliverable

Complete the activities in this lab and consolidate the responses into a single document. Name the document with your name and "Lab 10," like "George Self Lab 10," and submit that document for grading.

---

[1] The airquality data frame contains information from New York air quality measurements. The data frame includes these variables: airquality$Ozone, airquality$Solar.R, airquality$Wind, airqualtiy$Temp, airquality$Month, and airquality$Day.

[2] The ChickWeight data frame was generated from an experiment on the effect of diet on early growth of chicks. The data frame includes these variables: ChickWeight$weight, ChickWeight$Time, ChickWeight$Chick, and ChickWeight$Diet.

[3] The CO2 data frame contains information about carbon dioxide uptake in grass plants from an experiment on the cold tolerance of the grass species Echinochloa crusgalli. The data frame includes these variables: CO2$Plant, CO2$Type, CO2$Treatment, CO2$conc, and CO2$uptake.

[4] The mtcars data frame was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models). The data frame includes these variables: mtcars$mpg, mtcars$cyl, mtcars$disp, mtcars$hp, mtcars$drat, mtcars$wt, mtcars$qsec, mtcars$vs, mtcars$am, mtcars$gear, and mtcars$carb.