Hw 1: R Tutorial

This tutorial serves as a beginner's introduction to R. We walk through basic steps for reading in data, plotting, and obtaining descriptive statistics. The skills developed here will be used for the duration of the course. Additional functions will be introduced throughout the semester as needed. BE VERY CAREFUL to type all commands presented here exactly as they appear. Small errors in syntax, spelling, or capitalization will cause errors!

Please type your solutions and organize them in a clear manner. *There are several questions posed in each part. Be sure to address each question in your solutions. For each part, you should provide, (1) the R code, (2) the R-output, and (3) answers to questions posed for that part.*

---

1. **Data Entry:** We will be using the airfares.txt data set on the website site of an unrelated textbook (A Modern Approach to Regression with R, Sheather). This dataset examines the relationship between distance and cost for seventeen different flights. We will explore data entry using three different methods. You should be comfortable using any of these methods throughout the semester and may wish to keep this assignment as a reference.

   (a) We will first enter data using the "clipboard approach. First, open the dataset on the website at `https://raw.githubusercontent.com/moh3nnn/stat870datasets/master/airfares.txt`. Next, highlight and copy the data from this file (`ctrl+c`), including the variable names. In `R`, type the following command

   Data entry by copy and paste:

   - For a PC: `air.data1 = read.delim("clipboard", header=T)`
   - For a Mac: `air.data1 = read.delim(pipe("pbpaste"), header=T)`

   - You have defined a data set called `air.data1`. Notice the name is appropriate for the data topic and the `.data` portion reminds you that this is a data set.
   - The `'clipboard'` (or `'pbpaste'`) portion tells R to read the information you stored on the clipboard by highlighting and copying the data from the website.
   - The `header=T` portion tells R that the first row contains titles for each column.

   When you press enter, nothing should appear. The data set has simply been stored as `air.data1`. To see if the data was read in properly, type the following command and hit enter.

```
head(air.data1,2)
```

Does this display the entire data set? If not, what portion is displayed? What would you expect from `tail(air.data1,3)`? Try it out and provide the output to confirm your answer.

(b) Another approach for entering data would be to do so by hand. This method is not recommended for large data sets for obvious reasons! First we need to create three vectors for the variables. Then we need to put them together in a dataframe. We will give this dataset a different name so we do not confuse it with our first part. We will also change the variable names slightly to prove a point later on.

Manually defining a dataset

```
city=seq(1,17,1)
fare=c(360,360,207,111,93,141,291,183,309,300,90,162,477,84,
       231,54,429)
distance=c(1463,1448,681,270,190,393,1102,578,1204,1138,184,502,
           1828,179,818,90,1813)
air.data2 = data.frame(city, fare, distance)
```

- The command `seq()` creates a sequence of numbers starting at the first entry (1) and stopping at the second entry (17). The third entry tells `R` what to count by (one in this case).
- Note the `c()` syntax was used to create vectors for each variable
- The `data.frame()` was used to combine the vectors into one data set.

Type `air.data2` to view the new data set and confirm it was read in correctly. Then, try the command `dim(air.data2)`. Report your results and explain what you think it might mean. Finally, use the function `seq()` to create a vector from zero to twenty counting by two's. Report your command and the output.

(c) Let's enter the data one using one more method. We will read it directly from the website. Try the following command in R.

Entering data from a url or file path

```
air.data3=read.delim("https://raw.githubusercontent.com/moh3nnn/
                      stat870datasets/master/airfares.txt", header=T)
```

Note that this method can also be used to read a file stored on your computer by supplying the file path in place of the url in the code above. Enter `air.data3` in

the command line to ensure the data was read in correctly. Briefly comment on the strengths and weaknesses of each method and which you prefer. (Note: this is an opinion question.)

(d) Now that we have successfully read the data, we would like to access individual variables in the data set.

- Suppose we want to access the variable called `Fare`. Type `Fare` into the R console and report the resulting error. REMEMBER THIS ERROR! This is a common frustration in R and the solutions are provided in the next two steps.
- This error occurred because R does not view each variable as individual objects. They are located within the data frame `air.data1` (or `air.data3`). To access variables in the data frame, use the `$` syntax. Try the following command and report your results.

```
air.data1$Fare
```

- Using the `$` syntax for accessing variables in a data frame requires a lot of extra typing. An alternative is to `attach` the data frame so that each variable is its own object. Try the following.

```
attach(air.data1)
```

  Nothing should appear when you press enter, but now try typing `Fare` and report what appears.
- Finally, try typing in `fare` and report the result. Why did we not need to use `attach()` for the variable defined in the `air.data2` dataset?

Takeaways:

- There are multiple methods for reading in data (copy + paste, manually, from a website or url). Choose the most appropriate for the given situation.
- Name data sets appropriately (ex. consider the .data naming convention).
- Don't forget to attach the data!!!
- R is case sensitive. For example, `fare` and `Fare` were two different variables.

2. **Subsetting**

In some cases, you may wish to access select portions of a dataset. For example, perhaps you would like to only consider flights where `Fare` is greater than $300. This is called subsetting and the syntax requires square brackets to accomplish the task. Let's explore this with the following steps.

(a) If we would like to look at `Fare` values that are equal to \$300, we need to look at the variable `Fare`. In the R Console type the following and provide the output. Explain what you see.

```
Fare==300
```

Notice that we used a single equal to sign for defining/naming our data set. Now, it is important to use two equal to signs. This tells R that we are checking a condition (Is the value of `Fare` equal to 300?) rather than assigning something a name (`Fare` is the number 300). If `Fare` had text as values, we would put quotes around the value of interest (`Fare=="some text"`).

(b) Suppose we are interested in the value of `Distance` when `Fare` is greater than \$300. We can do this by subsetting `Distance` with square brackets under the condition that `Fare` is greater than \$300. Try the following and provide the output. Explain what you see.

```
Distance[Fare>300]
```

(c) Suppose we are not only interested in the value of `Distance` when `Fare` is greater than \$300, but all the other variables as well. We can obtain a subset of the entire data frame. First, notice that `Fare` only has one dimension (length of 17 observations). However, the data frame `air.data1` has two dimensions, 17 observations and 3 variables as seen in `dim(air.data1)`. In the square bracket notation, we now need two pieces of information. For example, type the following and report your output for each.

```
air.data1[1,2]
air.data1[1,3]
air.data1[2,2]
air.data1[ ,2]
air.data1[1, ]
```

What does the first entry in the bracket notation provide? Row or column information? What does the second entry indicate? What happens when the first entry is blank? What happens when the second entry is blank?

Using this information, we can subset the data set for values of `Fare` greater than \$300 with the following and provide the output.

```
air.data1[Fare>300, ]
```

Takeaways:

- Subsetting uses square brackets and use double equal signs to check a condition.
- Vectors only have one index.
- Data frames have two indices. The first indicates the row and the second indicates the column.

3. **Graphing**
R has many powerful graphing capabilities. The options for adding colors, titles, and labels are similar for each type of graph, so we will use scatterplots as an example. Scatterplots are used extensively in this course. They are useful for displaying the relationship between two variables using (x, y) pairs. To copy plots into your homework, simply right click on the figure to copy and paste. There are more sophisticated methods, but this simple approach should suffice for this class.

   (a) Suppose we wish to create a scatterplot of `Fare` versus `Distance`. Try the following and provide the resulting plot.

```
plot(Fare~Distance)
```

   Notice that the plot is plain and includes only default labels.

   (b) To include labels on the plot (which is ALWAYS requested) add the following options and provide the resulting graph.

```
plot(Fare~Distance, xlab = "Distance (miles)", ylab="Fare
   (dollars)", main = "Airfare According to Distance")
```

   (c) You can also add color to your plots. The site `http://kktg.net/sgr/otw-portfolio/new-fig-2/` has an extensive list of color options in R. Simply use the navigation at the bottom of the page to scroll through all 7 color charts. Plotting symbols can also be changed from the open circles using the `pch` argument. `http://www.sthda.com/english/wiki/r-plot-pch-symbols-the-different-point-shapes-available-in-r` has a nice list of graphical parameters, including values for `pch`. Try adding the following color and plotting symbol, or chose one of your liking, and provide the output.

```
plot(Fare~Distance,xlab="Distance (miles)", ylab="Airfare
     (dollars)",main="Airfare According to Distance",
                    col="cyan4",pch=8)
```

(d) Run the commands below on at a time. Copy the output for each and write 1-2 sentences commenting on the output. If you are feeling ambitious, feel free to play with the titles and colors as well.

```
boxplot(Fare)
hist(Distance)
```

4. **Summary Statistics**
R has many built-in functions to calculate basic summary information about variables. Here we look at a few of those functions for quantitative variables.

(a) We can calculate summary statistics for quantitative variables such as the mean, standard deviation, and five number summary. Let's calculate these statistics for `Distance`. Try the following and provide your output.

```
mean(Distance)
  sd(Distance)
summary(Distance)
```

(b) Finally, suppose we wish to summarize `Distance` only for values of `Fare` greater than \$300. Find the mean and sd using the commands above along with your knowledge of subsetting. Repeat for values of `Fare` less than or equal to \$300. (Hint: Use `<=`) Report the results. Compare and contrast the two groups in a couple of sentences.