MTH 324, Lab #3, Fall 2022 Name _____

Instructions: Follow along with the tutorial portion of the lab. Replicate the code examples in R on your own, along with the demonstration. Then use those examples as a model to answer the questions/perform the tasks that follow. Copy and paste the results of your code to answer questions where directed. Submit your response file and the code used (both for the tutorial and part two). Your code file and your lab response file should each include your name inside.

Creating statistical graphs in R

We are going to look at two different methods for creating statistical graphs in R. The first will use base R methods, and the second will be to use the package ggplot2. They work very differently, so we'll limit the kind of graphs we make to some standards. The cheat sheet link [4] below includes a cheat sheet for ggplot. It will be really useful as we expand our knowledge. The lecture on statistical graphs includes links to some of what we will cover in this lab, as well as a couple of resources that use other packages. I've also included a link below to an overview of some other common graphics packages that you might find useful for certain applications.

We'll use the same data file from the last lab, so start by importing that into R. We're going to pick up from where we left on in Lab 2. The Location color should be a factor, and the first column of Household identifiers should be removed.

```
library(readxl)
data <- read_excel("daemen/324lab2data.xlsx")

data <- data[-1]

data$Location <- factor(data$Location)
table(data2$Location)
```

Base R

To make a bar graph in base R using barplot() we need to first summarize the data in a frequency table. While you can make a quick-and-dirty graph from just the table, good graphs need titles and axis labels, so you'll need to specify a bit more.

```
barplot(table(data2$Location))
barplot(table(data2$Location),
        main = "Number of Households in Each Neighborhood",
        xlab = "Neighborhoods",
        ylab = "Counts",
        col = "darkgreen",
        horiz = FALSE)
```

If you'd rather, set horiz=TRUE to change the orientation. If the title is too long, add \n where you want the line to break.

We can make a histogram using hist().

```
hist(data2$Debt,
     main="Household Debt",
     xlab="Debt Amount in Dollars", ylab="Relative Frequency",
     col="darkmagenta",
     freq=FALSE
)
```

If we don't like the number of bars in the graph, we can adjust it with the breaks option.

We can create a boxplot of the same data with boxplot().

```
boxplot(data$Debt,
        main = "Household Utilities",
        xlab = "Utilities Bill per month in Dollars",
        col = "orange",
        border = "brown",
        horizontal = TRUE,
        notch = FALSE
)
```

Experiment with the settings to see which ones you prefer.

If we'd prefer a smoother density plot to a histogram, we can use the density() function.

```
plot(density(data2$Debt), main="Kernel Density of Debt")
polygon(density(data2$Debt), col="red", border="black")
```

The second line fills in the plot with shading.

We can also make a stemplot in R, but the output is to the console in plain text.

```
stem(data2$Debt)
```

We can compare two numerical values against each other in a scatterplot.

```
plot(data2$Debt, data2$Utilities, main="Debts vs. Utilities Cost",
     xlab="Debt in Dollars", ylab="Utilities Cost in Dollars", pch=19)
```

The pch option lets you change how the points are displayed.

To create a comparative boxplot, you would need to separate out the numerical variable into separate vectors based on the discrete or categorical one. That will need some more programming skills to extract from a dataframe, so we'll return to this when we need it a bit later in the course if we need it, but other graphing packages may make this unnecessary.

ggplot2

This is a popular graphing package in R. It's really powerful and can make a lot of different kinds of graphs, but it works a bit differently than base R graphics. For one thing, it's designed to work on dataframes and not bare vectors.

You can get a cheat sheet for working with ggplot2 here: https://github.com/rstudio/cheatsheets/blob/main/data-visualization-2.1.pdf but we'll go over the basics. This is still a great resource for other graph types, remembering the syntax, or using options.

The so-called grammar of graphics of ggplot is to first specify the data to be used and any common aesthetics, then specify the plot type, then add any statistical features, themes, position adjustments, legends and other finer details.

First, specify the data. In most cases, you will also specify variables here, unless you are overlaying other variables in later graphs.

In our examples, we'll work more or less backwards and start with the scatterplot.

```
44  library(ggplot2)
45  ggplot(data=data2,aes(x=Debt, y=Utilities))
46
```

This will set the variable range, but the graph will be blank. We add new elements with + sign. Let's make the scatterplot. The kind of plot we make is called a geom (short for geometry).

```
46  ggplot(data=data2,aes(x=Debt, y=Utilities))+geom_point()
```

This will create our scatterplot.

Turns out, we can make a comparative boxplot very easily with ggplot.

```
48  ggplot(data=data2,aes(Location, Debt))+geom_boxplot()
49
```

We can just as easily make one-variable plots.

```
50  ggplot(data=data2,aes(x=Debt))+geom_boxplot()
51  ggplot(data=data2,aes(x=Debt))+geom_histogram()
52  ggplot(data=data2,aes(x=Debt))+geom_density()
```

We can add color by using the fill option inside the geom parentheses. The labs() function also lets you add axis labels and a title. You can change the number of bins in the histogram by specifying the binwidth.

```
53
54 ggplot(data=data2,aes(x=Debt))+geom_histogram(binwidth =500,fill="blue")+
55   labs(title="Household Debt",x="Debt in Dollars",y="Frequency")
56
```

We can make a bar graph of factored or categorical data.

```
59 ggplot(data=data2,aes(x=Location))+geom_bar(fill='navy')+
60   labs(title="Household Neighborhoods",x="Neighborhoods",y="Frequency")
```

We can also easily create a normal probability plot using geom_qq().

```
62 ggplot(data=data2)+geom_qq(aes(sample=Debt))
63
```

Recall that the data should follow a straight line if the data is approximately normal.

The package ggplot is extremely popular and there are many packages out there that add functionality including methods of animating plots, or other features.

The last feature worth mentioning is that you can create side-by-side plots or in a table. In base R, the command is shown below. In ggplot, there is a faceting feature you can also use, for instance, if we wanted to make separate histograms for each neighborhood.

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

If you use this function, though, it will stay on until your turn it off by resetting it to c(1,1) or clear the Environment.

Tasks

1. Create a histogram of First Income. Add axis labels and a descriptive title (you should add these to all the graphs). Paste graph below. Change the number of bins and compare.

2. Make a dotplot of Family Size. Paste it below. What happens if you make a dot plot of Debt?

3. Make a comparative boxplot of First Income by Home Ownership. (You may need to convert the Home Ownership variable to a factor.) Try changing the theme of the graph in ggplot (see the cheatsheet).

4. Make a scatterplot of First Income by Monthly Payment, and color code the points by Home Ownership (look at the cheat sheet or online for how to add this feature to your graph). Paste the graph here. What other features could you add?

5. Make a graph of your choice that we haven't tried here, but which we talked about in class. A pie chart, for instance (in base R), or a violin plot (using ggplot) or another graph we discussed. You may need to google for specifics. Some graph types will require additional packages. Paste the graph here.

References:

1. Discovering Statistics Using R. Andy Field, Jeremy Miles, Zoe Field. (2012)
2. https://book.stat420.org/applied_statistics.pdf
3. https://scholarworks.montana.edu/xmlui/handle/1/2999
4. https://www.rstudio.com/resources/cheatsheets/
5. https://mode.com/blog/r-data-visualization-packages/
6. http://betsymccall.net/edu/CDSE/coding/R/bar_graphs.pdf
7. http://betsymccall.net/edu/CDSE/coding/R/boxplots.pdf
8. http://betsymccall.net/edu/CDSE/coding/R/histogram.pdf
9. http://betsymccall.net/edu/CDSE/coding/R/normal_plots.pdf
10. https://sites.harding.edu/fmccown/r/
11. https://rstudio-pubs-static.s3.amazonaws.com/7953_4e3efd5b9415444ca065b1167862c349.html