STAT 260 - Introduction to R Assignment 1

Introduction to R Assignment 1

- To download R: `http://www.r-project.org/`
- When you first open R, there will be a single window open, the R Console. You can enter commands in this window, and the output will appear here as well.
- It may be wise to open a Script Window. In the PC version of R, you do this by selecting File, then New Script. In the Mac version, select File, then New Document. The script window is useful because you can write many lines of code here and save your work as you go along.
- Commands in the script window do not run automatically. Put your cursor on the command line that you want to run, hit Ctrl+R on a PC (Command+return on a Mac) to run. You can also highlight several commands and run them the same way.
- To display the help file for any command, simply enter a question mark before the command name. For example, to see help for the command **hist**, I would enter **?hist**.
- If you are not sure of the name of a particular command, R allows for fuzzy matching. Enter two question marks followed by the search term. For example, enter **??histogram** to see a list of possible commands related to histograms.

Entering Data

- For now, we will enter numerical data as vectors. We will deal with tables/matrices and non-numerical data later.
- For a small number of observations, it is easy to enter the data directly. Suppose we have a set of five steel bolts, and we measure their weights and lengths:

Weight	12.3	12.5	12.7	11.5	15.3
Length	1.9	2.2	2.1	1.8	2.5

- It is good practice to use a descriptive name for your vectors, rather than a simple letter. Names must not contain spaces, but may contain periods. For example, I can use **bolt.weight** as a the name for my weight vector.
- To create my weight vector, I use the following command:

  bolt.weight = c(12.3, 12.5, 12.7, 11.5, 15.3),

  and to create the vector of bolt lengths I enter

bolt.length = c(1.9, 2.2, 2.1, 1.8, 2.5)

- We can read the symbol = as meaning "is defined as". The *c* before the data is the command R uses to create a vector.
- For a large number of observations, we can scan in the data. Suppose I want to create a vector called **odd.numbers** that consists of the first 10 odd numbers. I could enter the following:

odd.numbers = scan()
1
3
5
7
9
11
13
15
17
19

- Hitting enter on a blank line stops the process of reading in numbers. At that point, R will tell you how many numbers it has read in.
- The **scan()** command is especially useful if you are copying and pasting a column of data from a spreadsheet. The command will also accept data copied and pasted from a row spreadsheet; there is no need for commas to separate numbers with this command, only the spaces between the numbers.
- If we wish to display our vectors now in the computer's memory, we just enter the vector name. When I enter **bolt.weight**, I get the following output:
`[1] 12.3 12.5 12.7 11.5 15.3`
- The `[1]` at the beginning of the line is a counter. If the data had several lines worth of observations, there would be a counter at the beginning of each line to tell the reader which observation is at the start of that line.

Numerical Summaries

- To find the mean of our bolt weight data, we enter the command:
mean(bolt.weight)
- The commands **var(bolt.weight)** and **sd(bolt.weight)** give us the sample variance and sample standard deviation (respectively) for our weight observations.
- The command **cor(bolt.weight,bolt.length)** returns the sample correlation coefficient for the bivariate bolt data.

Visual Summaries

- To create a histogram of the bolt data, enter the command **hist(bolt.weight)**. You can use various options to make your histogram even better looking. The options **main** and **xlab** allow you to enter a customized title for your histogram and x-axis.
- With the command below, the title of the histogram will read "Bolt Study", and below the x-axis it will ready "weight of bolts (in g)".
  hist(bolt.weight, main="Bolt Study", xlab="weight of bolts (in g)")
- The **boxplot** command is used to create boxplots.
- Enter **boxplot(bolt.weight)** to create a boxplot of the bolt data.
- If I wish to compare two (or more) sets of data with boxplots, I can create side-by-side boxplots. For example, if I enter **boxplot(bolt.weight, odd.numbers)**, R will put boxplots for both data sets on the same axes.
- To label the boxplots on the x-axis as bolts and odd numbers, enter the command:
  boxplot(bolt.weight, odd.numbers, names=c("bolts","odd numbers"))
- If I wish to know what values R is using for the boxplots, I can use the summary command. When I enter the command **summary(bolt.weight)** I get the following output:

```
Min.   1st Qu.  Median  Mean   3rd Qu.  Max.
12.10  12.30    12.50   12.44  12.60    12.70
```

- The bottom of the box is at the first quartile (12.30) and the top of the box is at the third quartile (12.60). The line in the middle of the box is at the median (12.50). The smallest observation is 12.10, and the largest observation is 12.70.
- For the Windows version of R, if you right-click on your graphic that you have created, you can select Copy as metafile. Then, in a Word or Open Office document, you can paste it in using Ctrl+V. On a Mac, select your graph, enter Command+C and then paste it into a Word document using Command+V.
- The **plot** command is used to create scatterplots.
- Enter **plot(bolt.weight,bolt.length)** to create a scatterplot of the bolt data; bolt.weight is plotted on the x-axis, while bolt.length is plotted on the y-axis. The options **main**, **xlab**, and **ylab** allow you to specify the main title, x-axis label, and y-axis label:
  plot(bolt.weight,bolt.length, main = "Bolt Study", xlab = "Weight (g)", ylab = "Length (cm)")