# MATH203 - Introduction to R-Studio - Continued

Tutorial Notes - James McVittie

## 1 Downloading R-studio - Troubleshooting

R-studio is a more user friendly version of R and after downloading some computers may require that the user also installs the R program itself. If R-studio runs on your computer and there is no prompt to download the R program, skip this section. R can be downloaded by going to the following website for the Windows download:

https://cran.r-project.org/bin/windows/base/

or for a MAC operating system to the following website:

https://cran.r-project.org/bin/macosx/.

Please ensure that you are downloading the correct version for the operating system on your computer.

## 2 More Code and Statistics

### 2.1 Skewness

From the last handout, the dataset known as "JamesTut1" was imported into R-studio where we calculated the mean and the median. Here we will add on to these calculations by trying to understand the meaning of skewness. As a reminder of the definition of skewness see page 54 of text or the equivalent definition below:

**Definition 1.** *Data is said to be* skewed to the right *if the median is less than the mean. Data is said to be* skewed to the left *if the median is greater than the mean. Data is said to be* symmetric *if the mean is equal to the median.*

Using the dataset "JamesTut1", we will determine the skewness of the height variable data and the skewness of the weight variable data. The code for calculating the mean of height is the following:

```
mean(JamesTut1$height)
```

which outputs 176.3 and the code for calculating the median of height is the following:

median(JamesTut1$height)

which outputs 175. Since the median is less than the mean then the data is skewed to the right. This means the right tail of the distribution has more extreme observations and has pulled the mean to the right.

Similarly for the weight variable, the mean is 155.6 and the median is 153.5 (Exercise: type the code to obtain these values). Since the median is less than the mean then the distribution of the weight variable is also skewed to the right.

## 2.2 Median, Quantiles and Boxplot

Now, we will be using a different dataset known as "JamesTut2" which contains one variable called "x" with 20 values between 1 and 20. We will show calculating by hand how to obtain all the important values in a boxplot and then show that R-studio reproduces all the same values in a boxplot. Using the instructions from the first handout we read in the dataset and obtain the following 20 values for x:

$$1, 2, 4, 3, 7, 5, 9, 11, 15, 17, 19, 13, 13, 12, 16, 9, 14, 13, 16, 2$$

then using the sort() function on R-studio we can put them all in order

$$1, 2, 2, 3, 4, 5, 7, 9, 9, 11, 12, 13, 13, 13, 14, 15, 16, 16, 17, 19$$

Now to calculate the median, we systematically go from the right and the left to obtain the middle number. Since there is an even number of observations, we take the average of the two middle numbers: 11 and 12. Thus, the median is 11.5. To calculate the first quartile, we calculate the median of the first ten ordered numbers. Since there are ten, we take the average of the middle two numbers: 4 and 5. Thus, the first quartile is 4.5. Similarly, to calculate the third quartile, we calculate the median of the last ten ordered numbers. Since there are ten, we take the average of the middle two numbers: 14 and 15. Thus, the third quartile is 14.5.

Now that we know the upper quartile and the lower quartile, we can calculate the interquartile range (IQR):

$$IQR = Q_U - Q_L = 14.5 - 4.5 = 10$$

We can see from the ordered data that the maximum value is 19 and the minimum value is 1. Then, using R-studio, we can plot these values in a box-plot using the command:

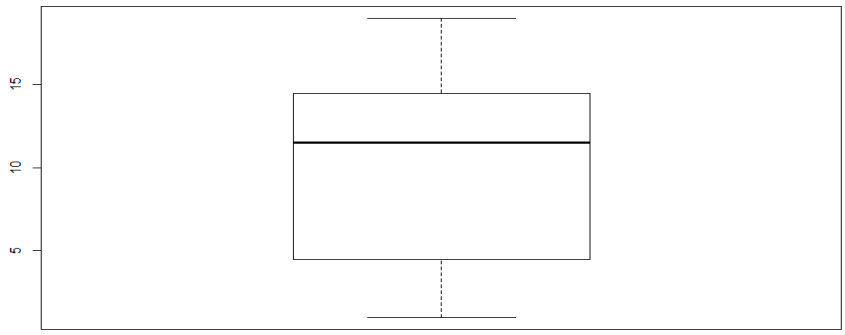
$$boxplot(JamesTut2\$x)$$

(NOTE TYPO FROM LAST TUTORIAL: The whiskers of the boxplot are drawn from each hinge to the MOST EXTREME MEASUREMENT INSIDE the inner fence where the inner fences are calculated as:

$$\text{Lower inner fence} = Q_L - 1.5(\text{IQR})$$

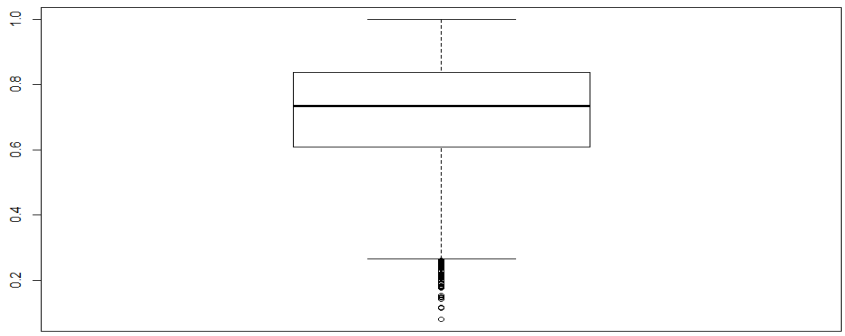
$$\text{Upper inner fence} = Q_U + 1.5(\text{IQR})$$

and so this can be seen in the boxplot). This means that when drawing the boxplot, the whiskers can be different lengths and give an indication to the skewness of the data. The boxplot of the data can be found below:

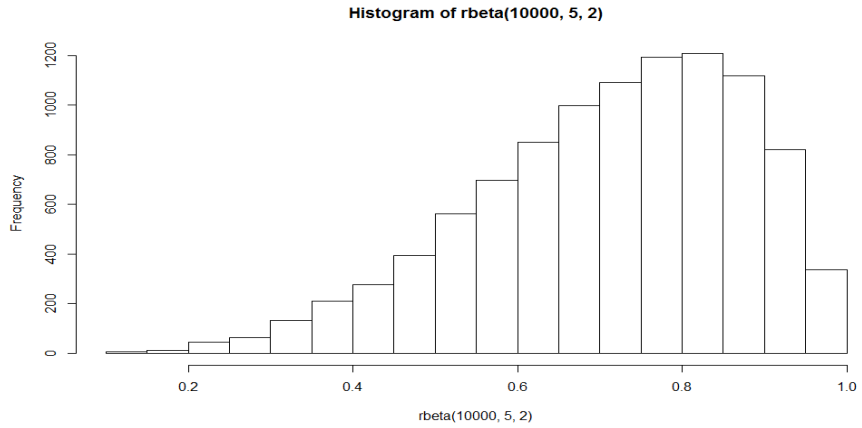



## 2.3 Shapes of Histograms and Boxplots - A Comparison

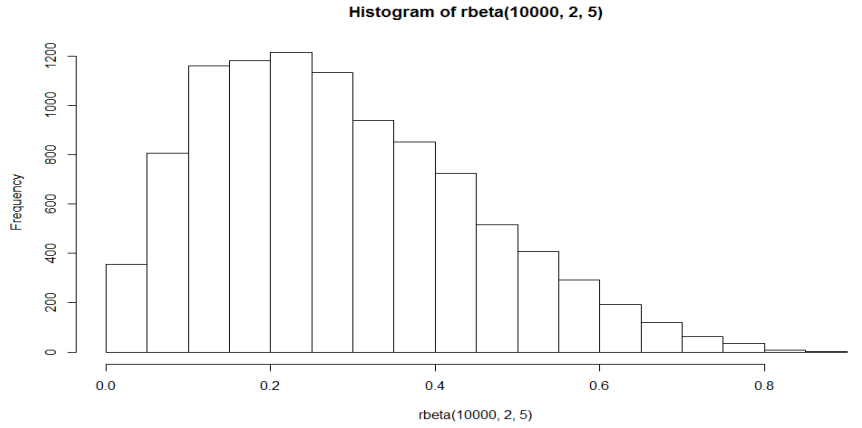
In the previous example, we were able to produce a boxplot of the data. In this section, we examine how to guess the approximate shape of the boxplot from a histogram and vice versa with simulated data (Note: R code to generate the data will not be included as it is beyond the scope of this course.). First, we will begin with a boxplot and try to understand the skewness of the data and the potential histogram.



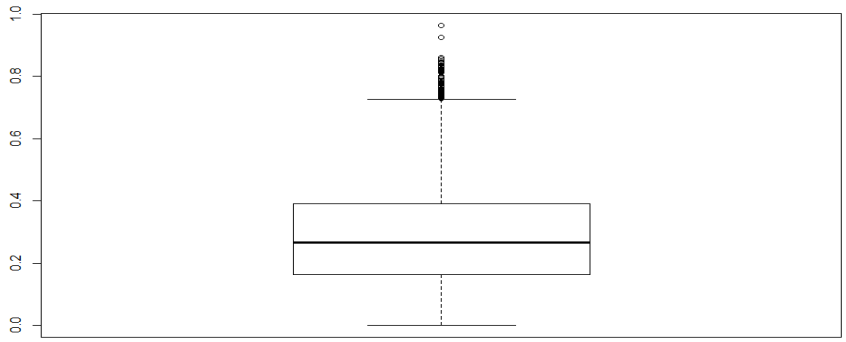
From the above boxplot, we see that the left tail is very long relative to the right tail with many observations outside of the left inner fence. This would imply that the left tail is very long and the data is left skewed. By visualizing the histogram of the data, we see that this is exactly the correct conclusion:




Now, we will begin with a histogram and try to visualize the boxplot. The histogram for the new set of data is below:




The distribution here is clearly right skewed as the right tail of the histogram takes on large values. In visualizing the boxplot, we would expect that the right whisker to be much longer than the left whisker on the boxplot as we have many more extreme observations towards the right of the distribution. This can be confirmed by examining the following plot:




Exercise: Given either a symmetric boxplot or a symmetric histogram, draw the corresponding histogram or boxplot. Will the corresponding plots also be symmetric?

## 2.4 Adding labels to the R-studio Plots

In this section, we will present how to relabel the plots obtained in R-studio. Using the "JamesTut1" datafile, create a scatter plot (i.e. points on the graph) with the height as the x-axis and weight as the y-axis using the code:

$$plot(JamesTut1\$height, JamesTut1\$weight)$$

Now in the plot, we see the default code for each of the axes. Suppose now we wish to change the x-axis to "Height of Patients". Then we add the following command to the plot function:

$$\text{xlab} = \text{"Height of Patients"}$$

so the plot function with the additional command becomes:

$$plot(JamesTut1\$height, JamesTut1\$weight, \text{xlab} = \text{"Height of Patients"})$$

To change the y-axis label, we add the following command

$$ylab = \text{"Weight of Patients"}$$

to the plot function. So to change both the axis labels, we have the following code:

$$plot(JamesTut1\$height, JamesTut1\$weight, \text{xlab} = \text{"Height of Patients"}, \text{ylab} = \text{"Weight of Patients"})$$

To add a title, we add the extra option

$$\text{main} = \text{"A plot of Height Versus Weight"}$$

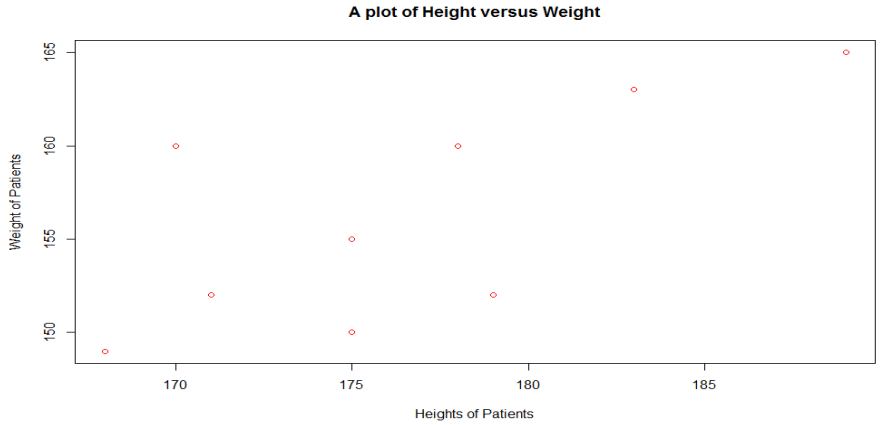
into the plot function. Finally, to change the colour of the data points in the plot function to Blue or Red, we can add the option

$$\text{col} = \text{'Blue'}$$

or

$$\text{col} = \text{'Red'}$$

A more appropriate graph that visualizes the relationship between height and weight with all the extra labels using red points can be seen below:

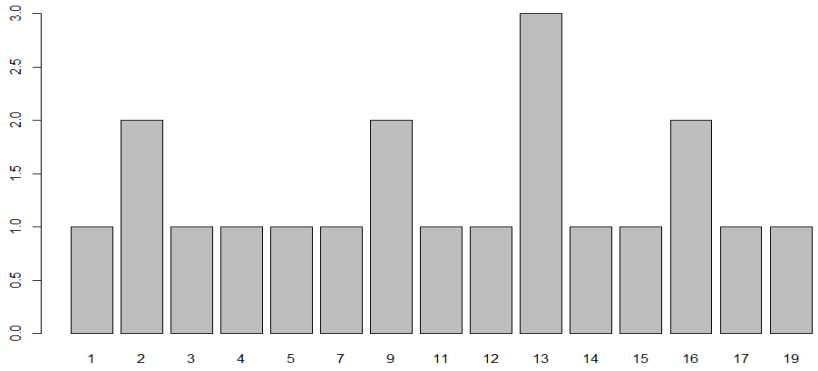



## 3 Two New Graphs - Bar Graph and Pie Chart

The first new graph we will be creating is the bar graph. Unlike the histogram which sets bins and counts the number of observations in each bin, the bar graph creates bars for each individual observation value. So given data "1,2,3,4,5" then the histogram with bin for values between 1 and 5 would have frequency 5 whereas the bar graph would have 5 separate bars each with frequency 1. Using the JamesTut2 dataset, we produce a bar chart of the variable x below using the code

barplot(table(JamesTut2$x))

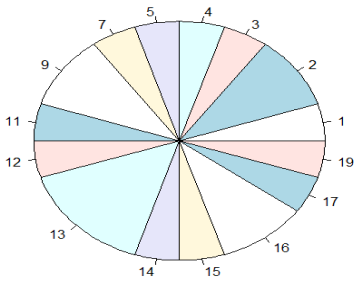
where the extra table() command is to ensure the data is in the correct format. The command above would produce the following output:




Another useful graph is the pie chart which visually represents using a pie, how many times each observation was represented in terms of the whole dataset. This can be obtained using the following code:

```
pie(table(JamesTut$x))
```

where again the table() command is so the data is in the correct format. The pie command would produce the following plot:




## 4 Final Comments

We can now create pie charts, bar charts, histograms, scatterplots and boxplots. In projects and reports in many different fields, these visual representations of the data allow researchers to understand the shape of their data rather than using the values of the mean or variance for instance. Producing plots is very useful in practice; however, please focus on understanding the meaning of each of the plots and what each of them represents.