Derek Holsapple

Assignment 2

7/16/2017

MAJOR FINDINGS OF DATA CLEANING

The following data has been used by StatLib---Datasets Archive

<http://lib.stat.cmu.edu/datasets/>

[**bodyfat**](http://lib.stat.cmu.edu/datasets/bodyfat)

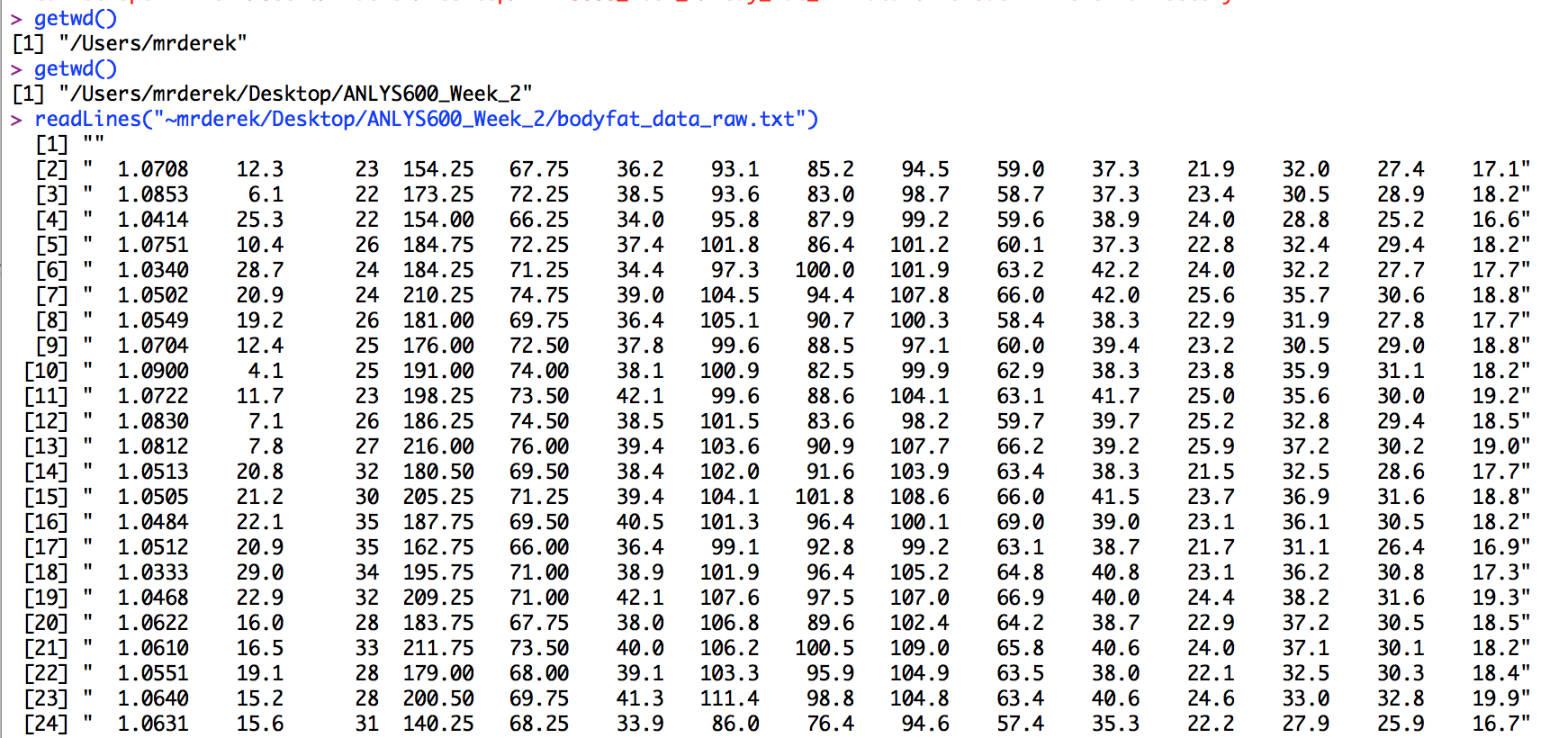
Lists estimates of the percentage of body fat determined by underwater weighing and various body circumference measurements for 252 men. Submitted by Roger Johnson (rwjohnso@silver.sdsmt.edu) [2/Oct/95](35 kbytes)

The file selected was and downloaded into the folder

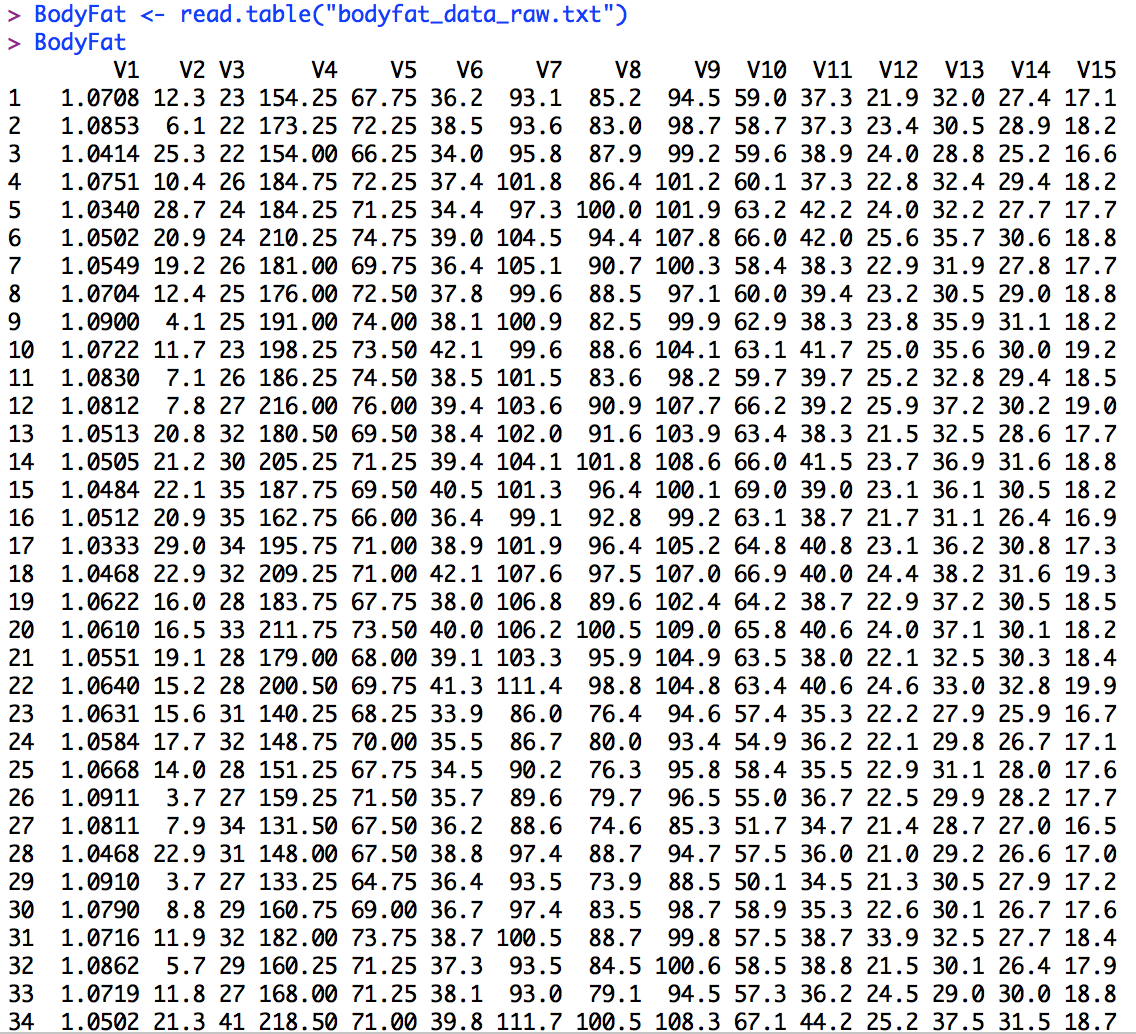
"~mrderek/Desktop/ANLYS600\_Week\_2/bodyfat\_data\_raw.txt"

Here is a screen shot of the Raw Data. Another component with the column names is in a separate document. The data seems to be relatively clean but needs column names along with proper cleaning to determine the validity. We will now start the search for data entry errors

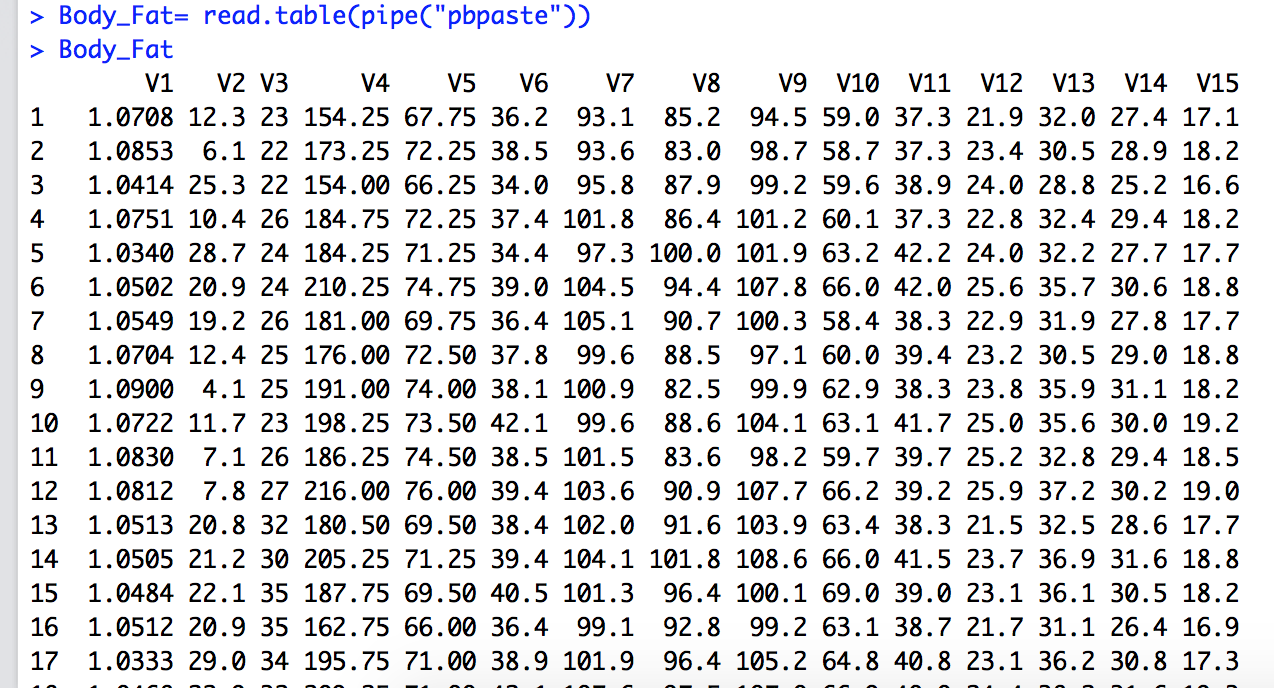
My first task after downloading the raw data was to set up my directory. I used getwd() to determine where my current directory was set up. I then changed my directory to the file of where my data is located. To see that I have readable data I used the function readLines() to determine what kind of data cleaning practices must be used.



I then created the variable “BodyFat” and used the function “read.table” for the table below. It seems the data converted nicely with limited amounts of indexing required. Next I will rename the columns to reflect the correct associated names found in the main file.



Alternatively I could have used a different reference to execute the same behavior of “read.table” if for some reason I was having compatibility issues. Utilizing the clipboard operating off of a Macintosh operating system data can be collected by the copy paste method by using the command (pipe(“pbpaste”)). This is a fast and easy shortcut if you did not want to download the whole file or reduce indexing time. I will utilize the table above for assignment compatibility.



In the code below I created columns for the associated data. The data will now be easier to read and reference. The names of the columns were found in the file downloaded. The column names are longer than the names assigned to the data table. I wanted to be able to see all the column names and chose to reduce the size of the original column names for referencing. For example, the column name “BodyFat” was reduced from it’s actual name of “Percent body fat from Siri's (1956) equation”. The real column names are listed below.

 1.Density determined from underwater weighing

 2.Percent body fat from Siri's (1956) equation

 3.Age (years)

 4.Weight (lbs)

 5.Height (inches)

 6.Neck circumference (cm)

 7.Chest circumference (cm)

 8.Abdomen 2 circumference (cm)

 9.Hip circumference (cm)

 10.Thigh circumference (cm)

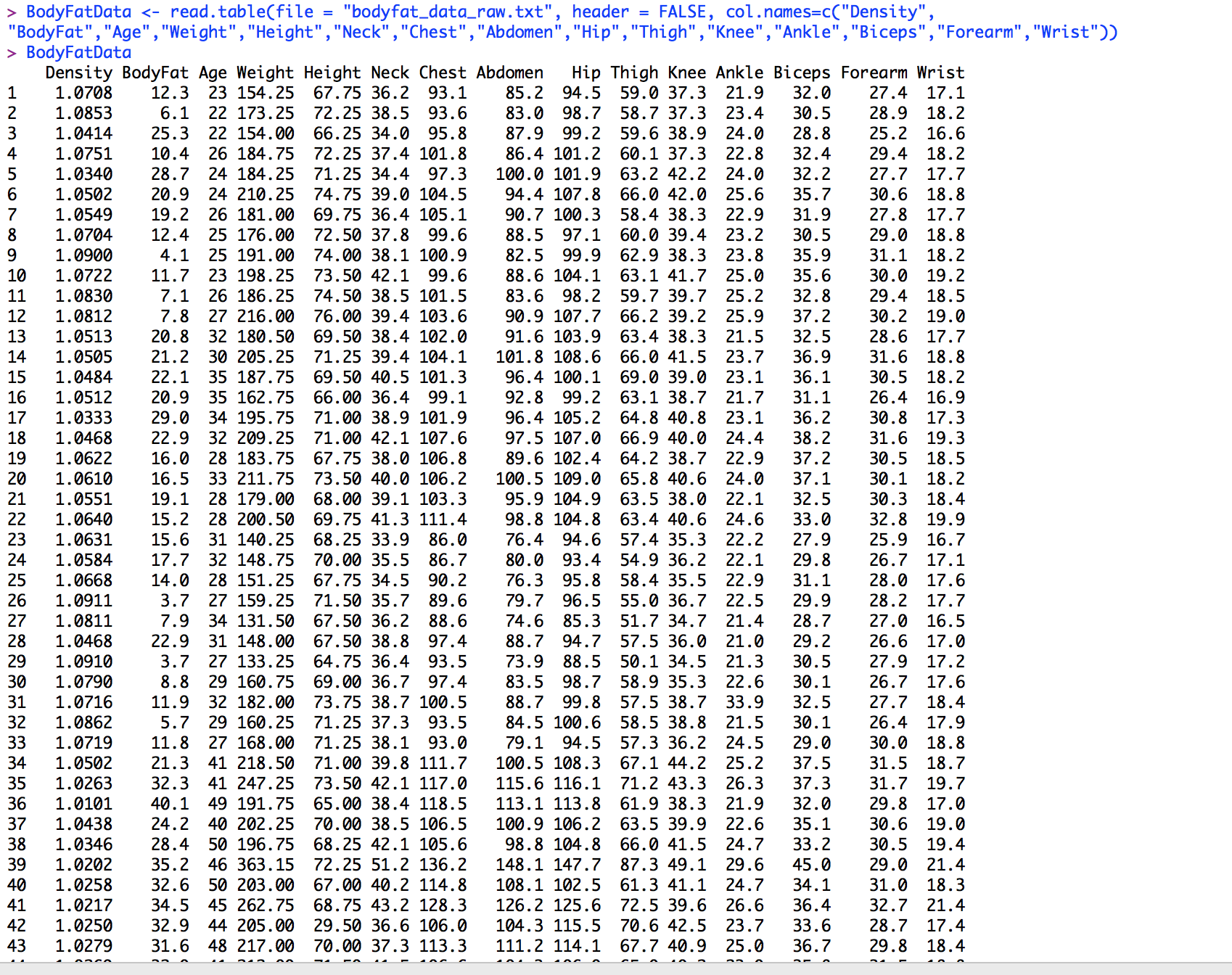
 11.Knee circumference (cm)

 12.Ankle circumference (cm)

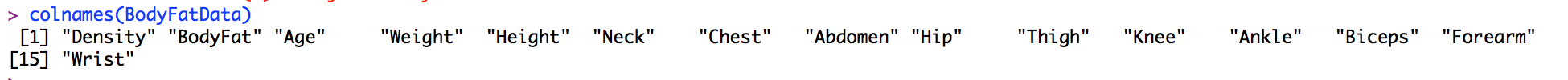
 13.Biceps (extended) circumference (cm)

 14.Forearm circumference (cm)

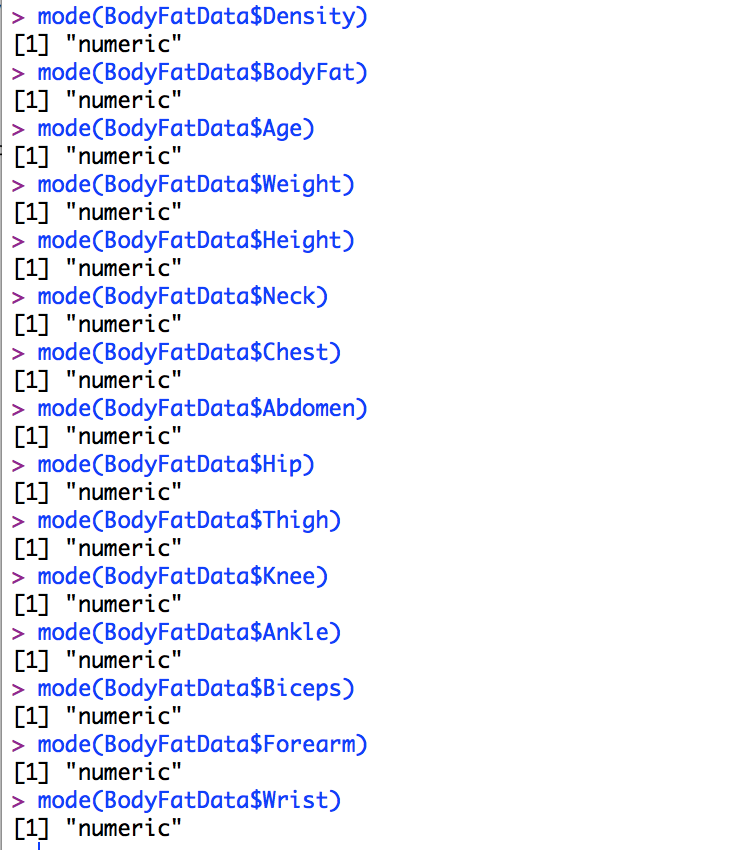
 15.Wrist circumference (cm)



To check my work, I use the function below to ensure that the column names to the associated data table are correct.



Now I must examine the content of the Data. I must make sure that all the storage mode reads “numeric” so that statistical analysis can be conducted later. It seems all the data in each column will be stored as numeric data.

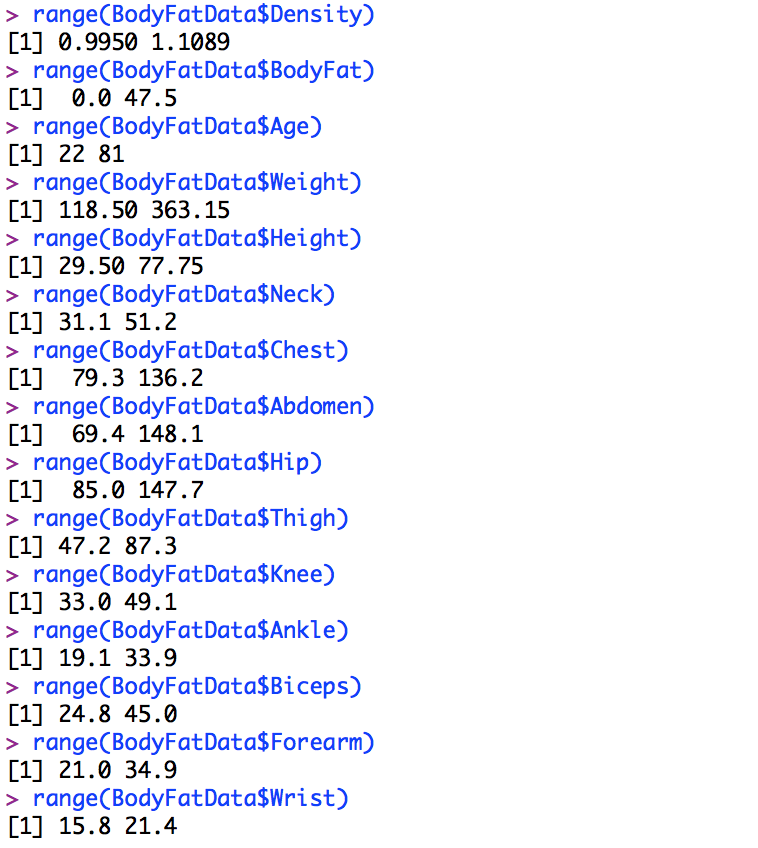


Next I must determine the object class of each column.

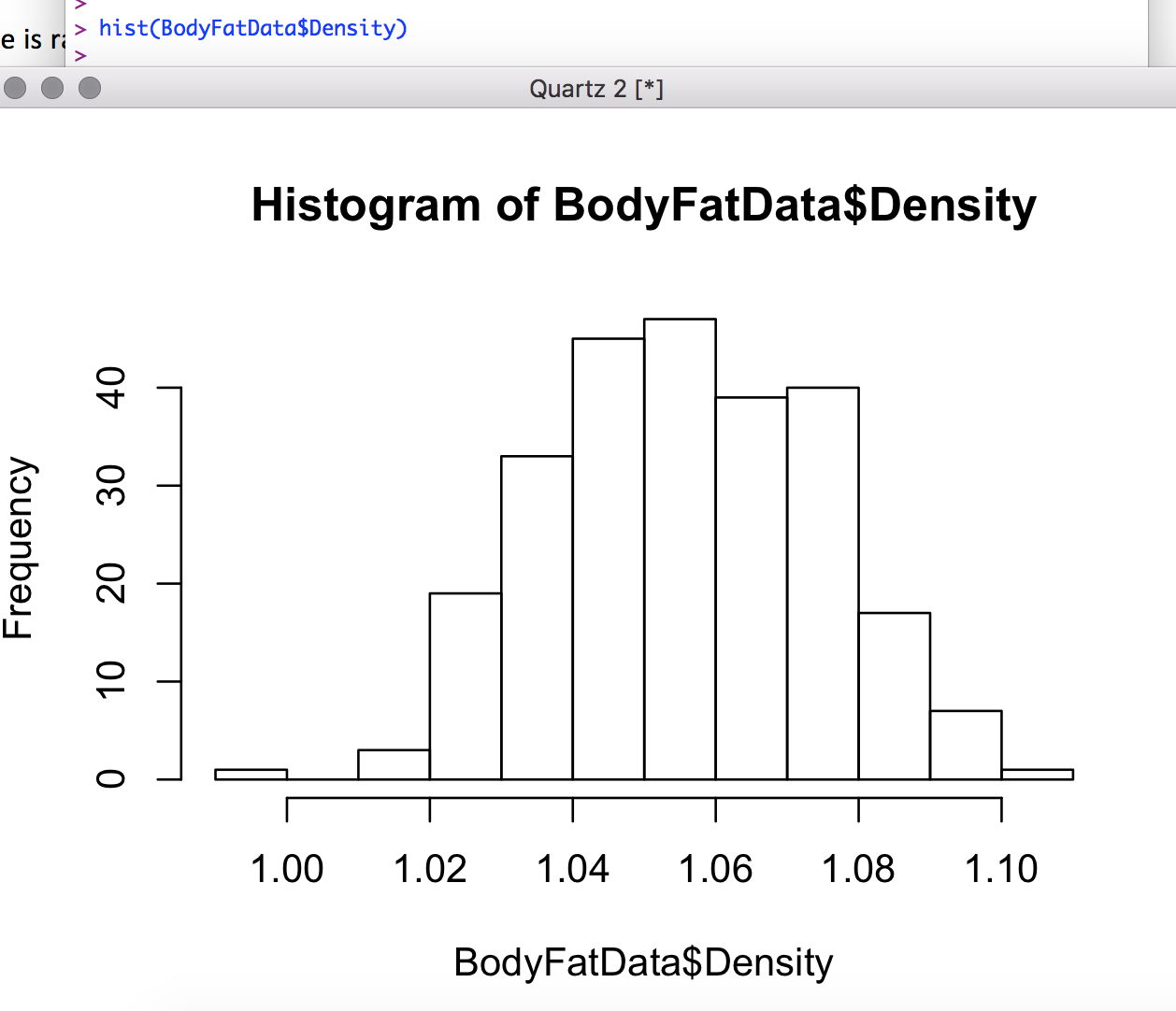


The data looks great. No indexing required to change any categorical or integer data.

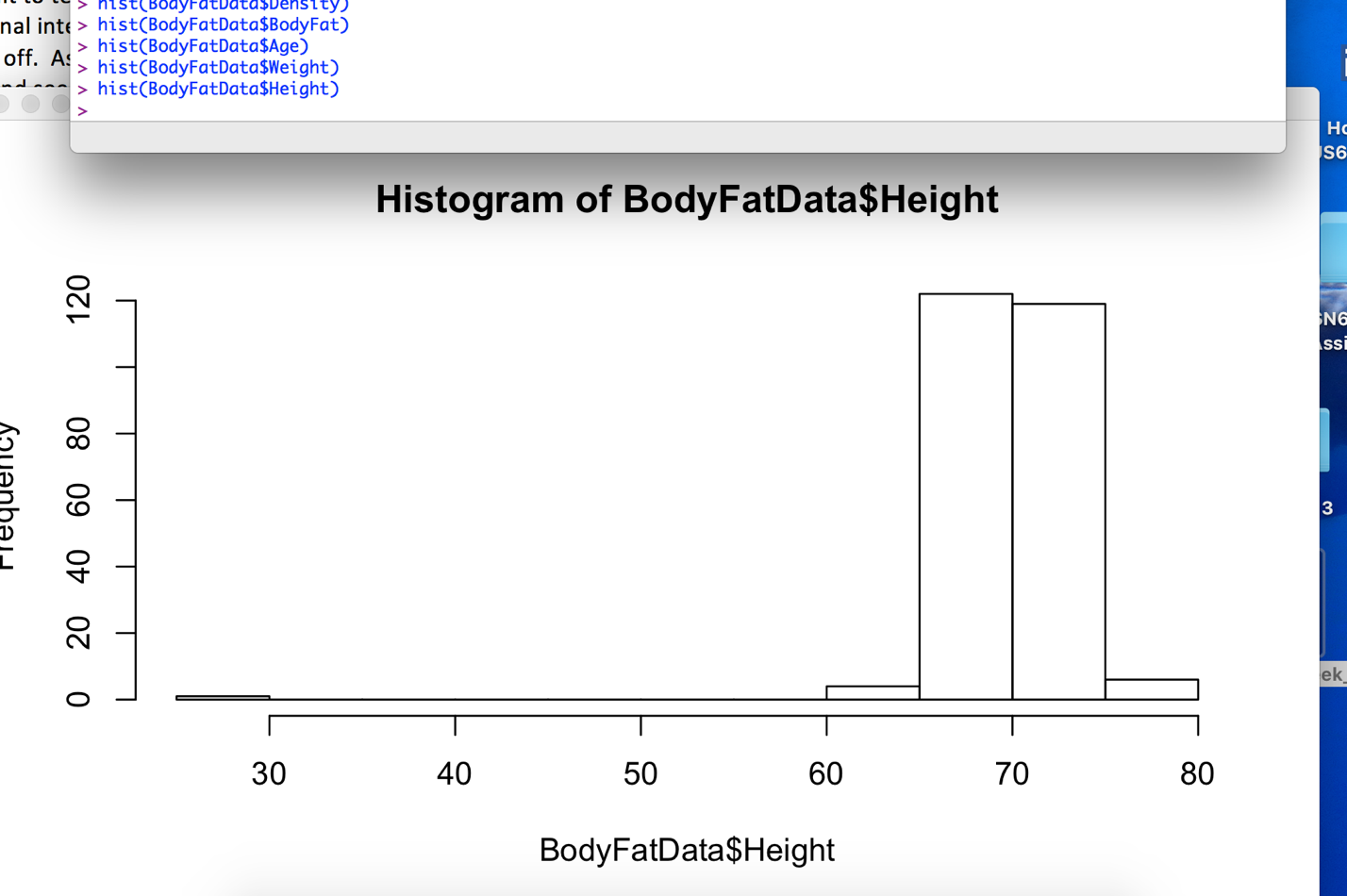
I must now look at all the columns ranges to see if there was a possible human data entry error. It is difficult to tell if data is off because I do not know what abnormal is. Because I am not a nutritionist I do not know if the ranges seem normal or off. As a statistician I can search for outliers. I can create histograms of the following columns and see if there are any possible outliers from data entry errors.

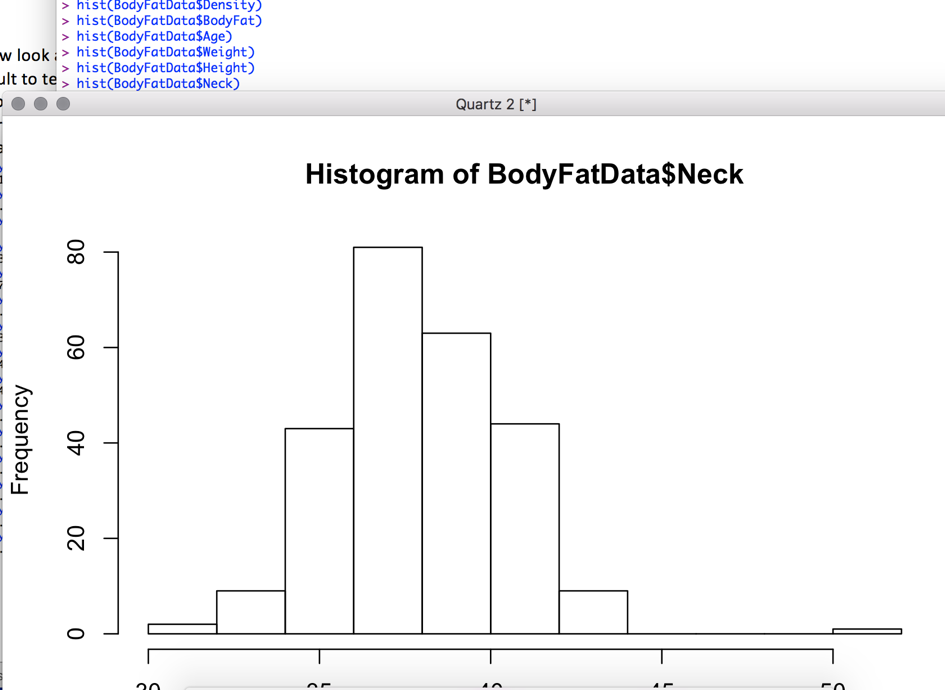


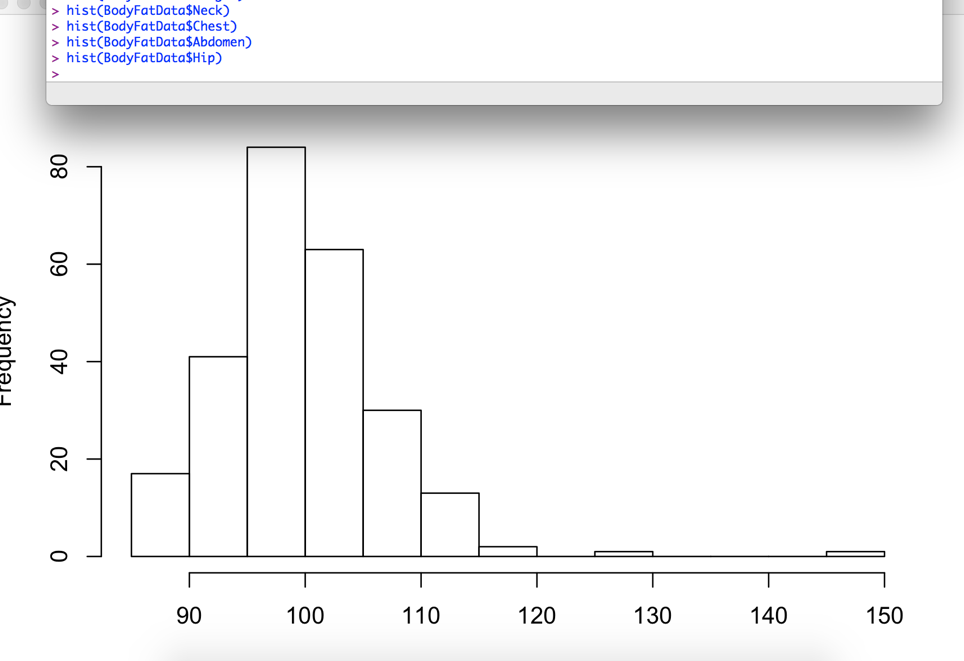
Since we assume there is random sampling involved in this data collection experiment we can expect the data to represent a relatively normal distribution in all histograms. If an outlier appears we can investigate more to determine if it was a data entry error or a real outlier. For the column “Density” we see the distribution is normal with no outliers. For the sake of this assignment I will not give screenshots of the other 14 histograms. If I find a discrepancy I will show that histogram.



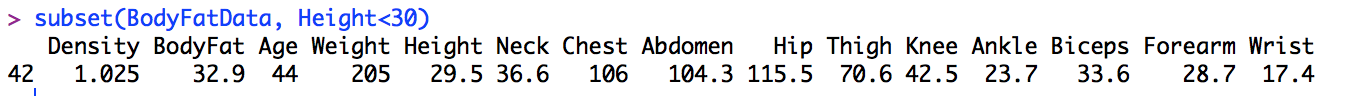
In the data, we found a discrepancy. When running the histogram for “Height” there seems to be an outlier that will require more investigation. I also found a possible outlier for the “Thigh” and “Hip” categories. These two are not as extreme of outliers as much as “Height”. I believe its safe to say that height can be attributed to data entry error due to less than 30in is close to 2 1/2ft. The other possible outliers “Hip” and “Neck” are plausible and will not be indexed for this dataset. The largest hip circumference in the world is 822cm which is well below our possible outliers. A body builder can also have a neck circumference size above 55cm.



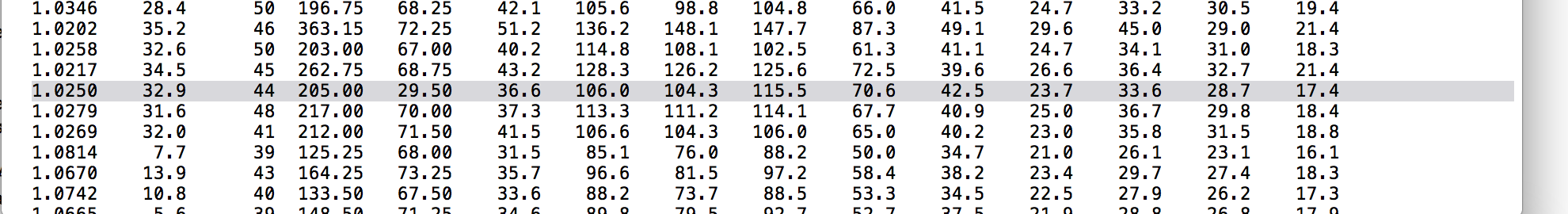




I now must localize the problematic piece of data from the “Height” column. To do this I will run the function subset() and look for the line that displays “Height” less than 30in.



We find that only one subject falls into this category. Other data seems consistent except for “Height”. This person would be an interesting specimen if he existed. Very short being 2 1/2ft and weighing 205lbs. We will investigate further by looking at the raw data to confirm that this was entered 29.5in.

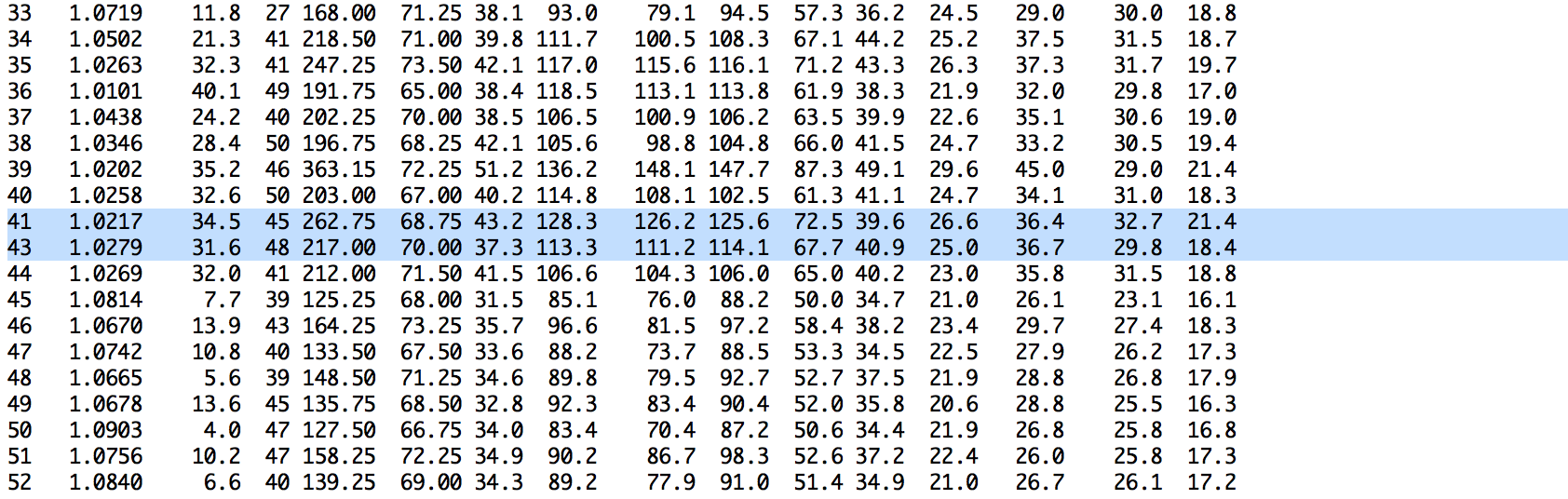


In the raw data, we find that indeed 29.5 was entered for subject 42. It is safe to say that there was data entry error for this case. We will now have to eliminate this subject from the experiment.

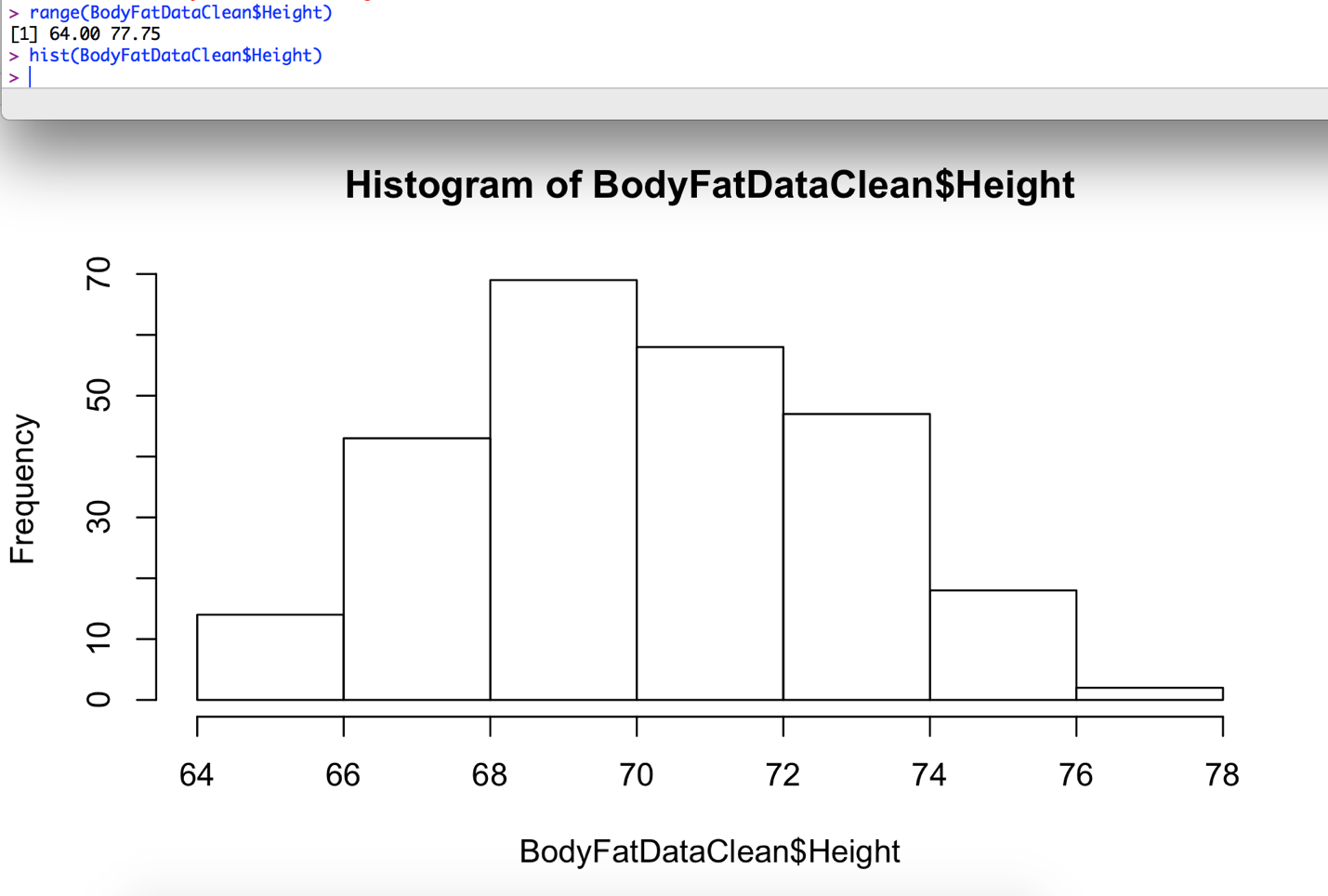
To delete row 42 we use the function below



To check that row 42 is gone we scroll through the data to find a missing 42 row.

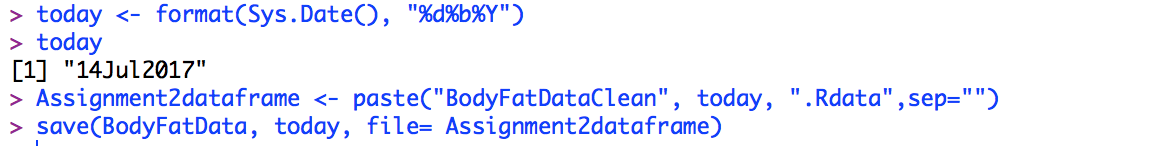


With Subject 42 now gone we will run our range, and histogram again with the new data set.

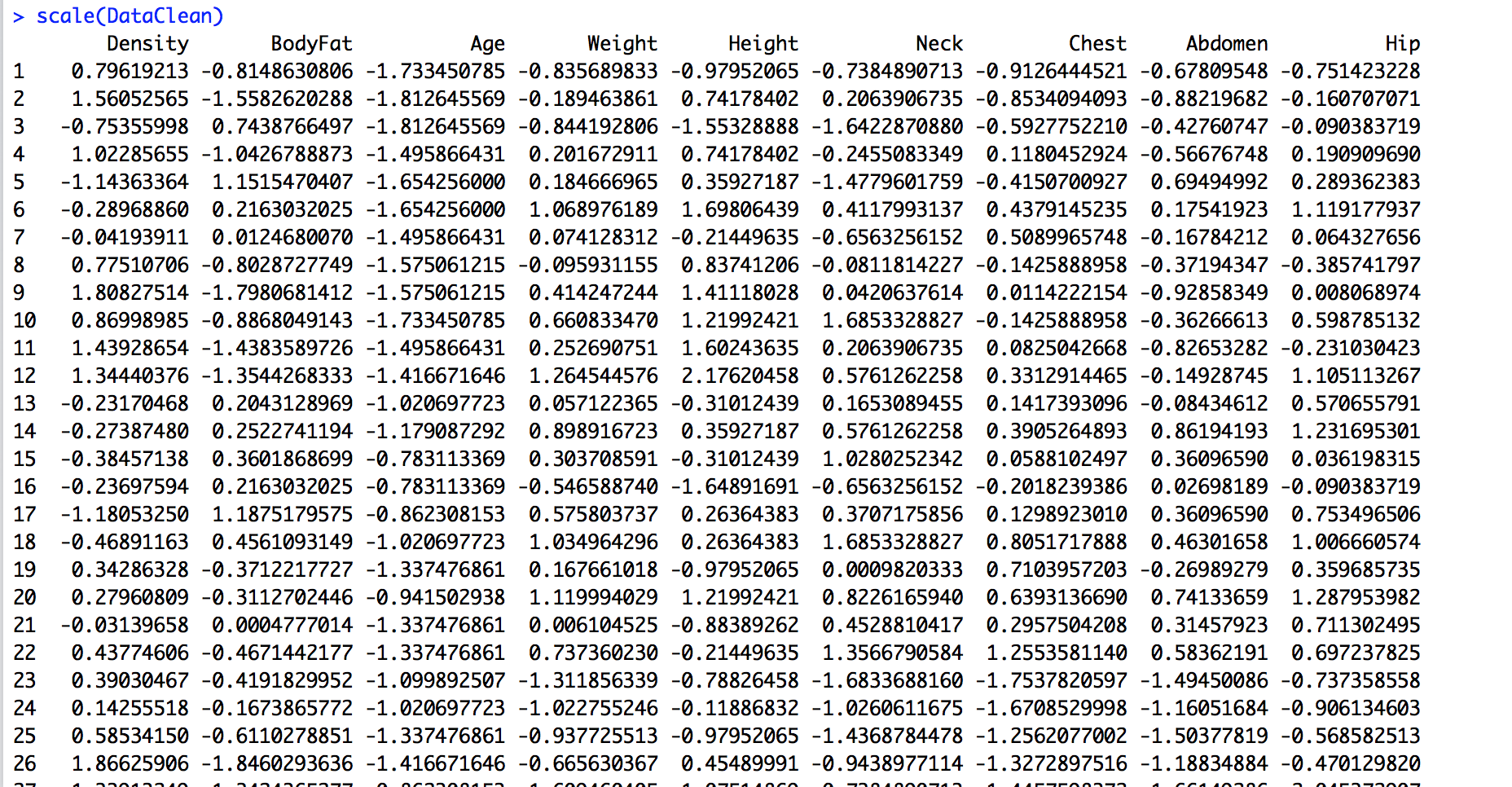


The histogram looks normally distributed with no signs of outliers. Looks like we are good to keep moving forward. We will add a note onto the final data that we eliminated subject 42 due to discrepancies described above.

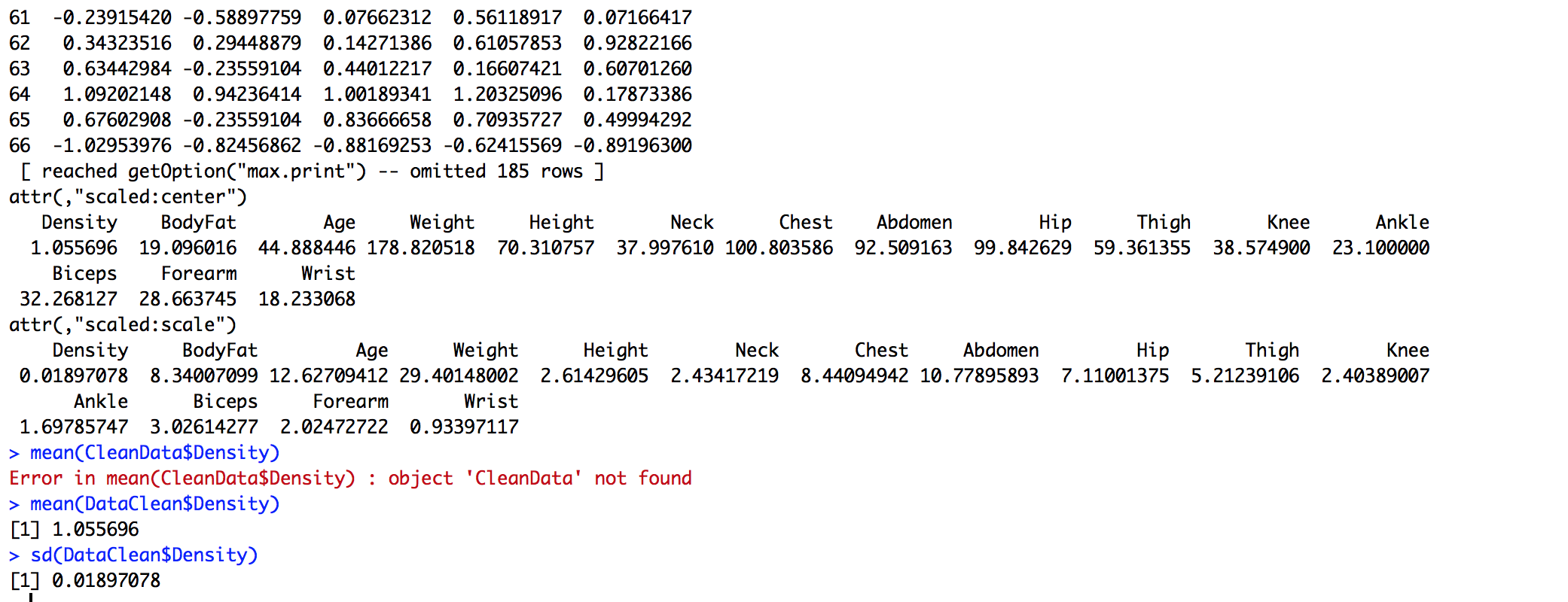
The data is now cleaned and I will now save this data set for later analysis.



We are now ready to start normalizing the data to take a look at the distributions. In the data frame below we have found the Z-Score transformation using the scale() function. The Z-score transformation. This is a standardization of the data. The scale() function by default calculates the mean and standard deviation of the whole vector and then scales each data by values subtracting the mean and dividing by the standard deviation.



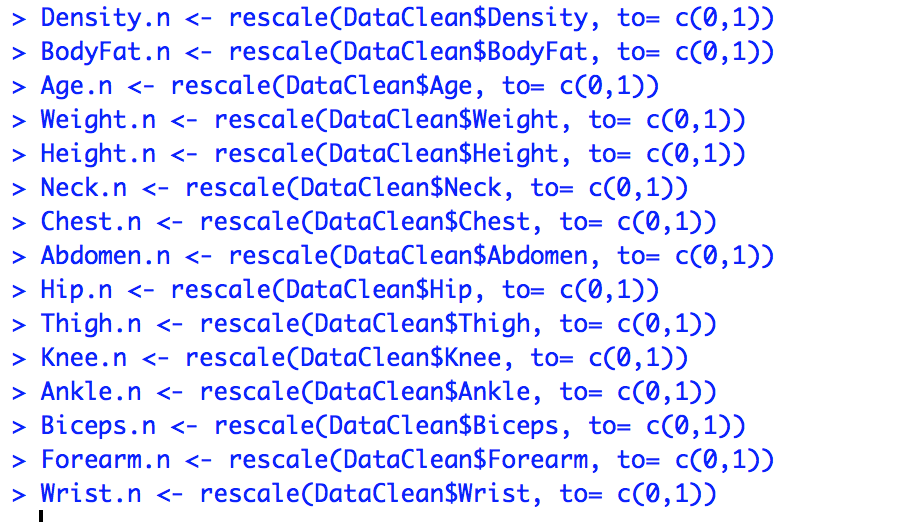
To check the mean and standard deviation to know our function operated correctly we simply check the mean of the column “Density” and the Standard deviation.



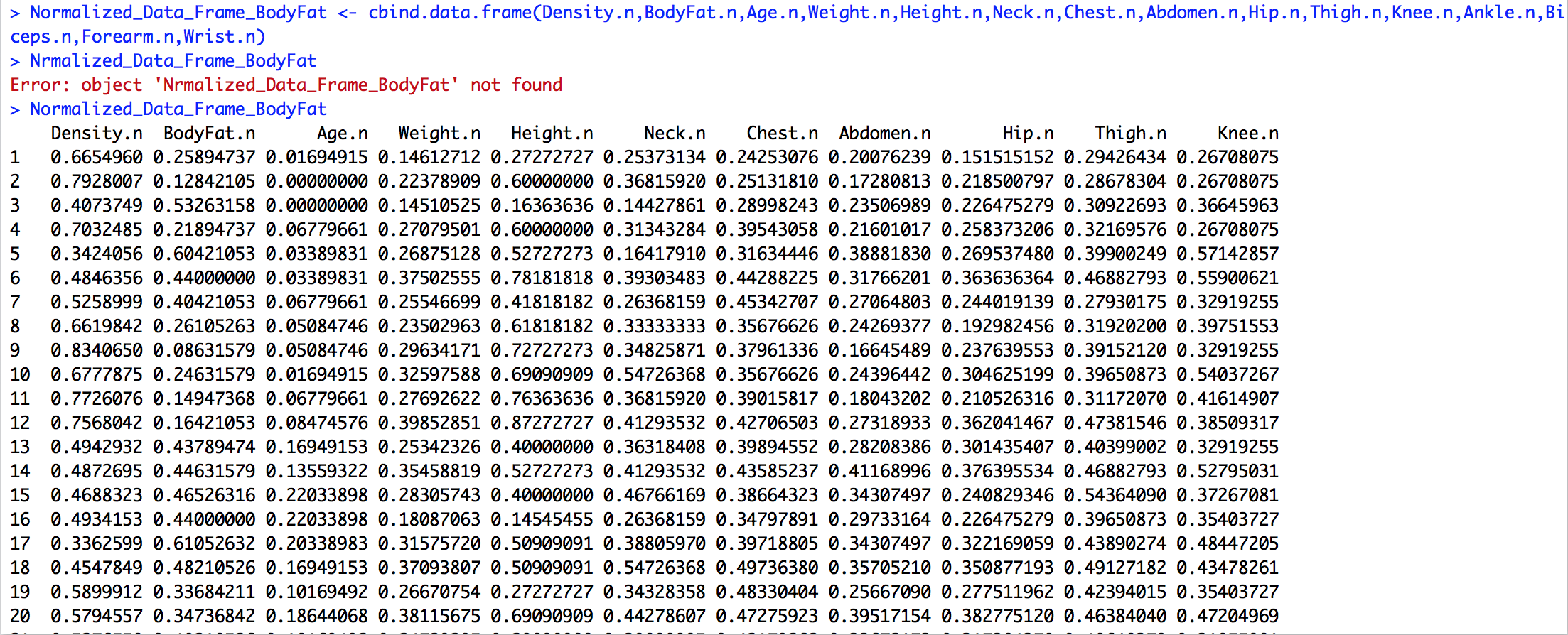
I now am going to normalize the data starting with the Density column. Using the rescale function we will pull the Density column from our cleaned data to normalize the data on a scale of 0-1.

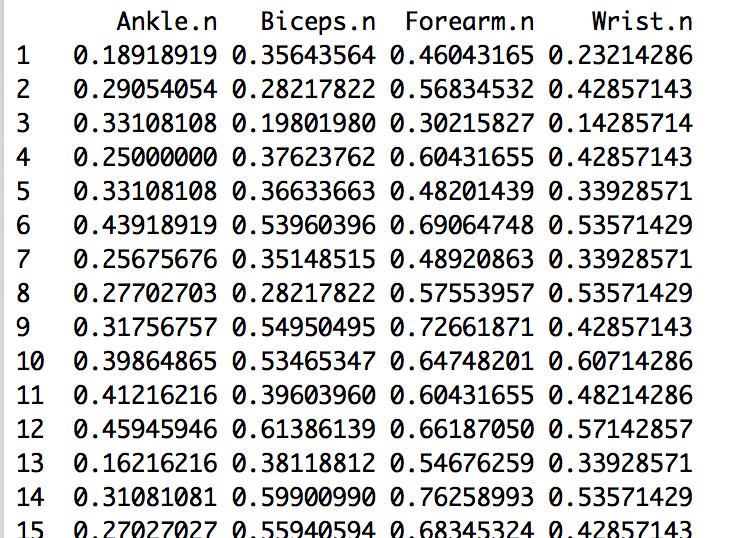


I then created vectors for all the rescaled data.

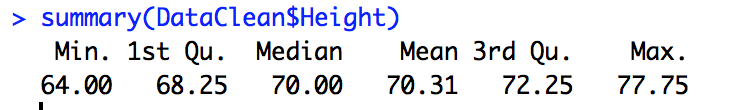


I then binded these vectors together to create a data frame of the normalized data points 0-1 of each column for cross comparison.

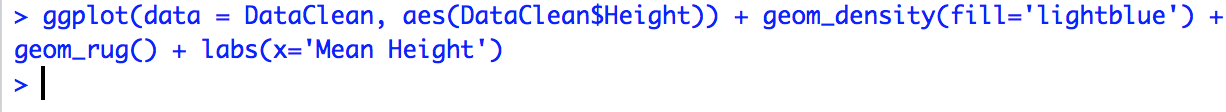


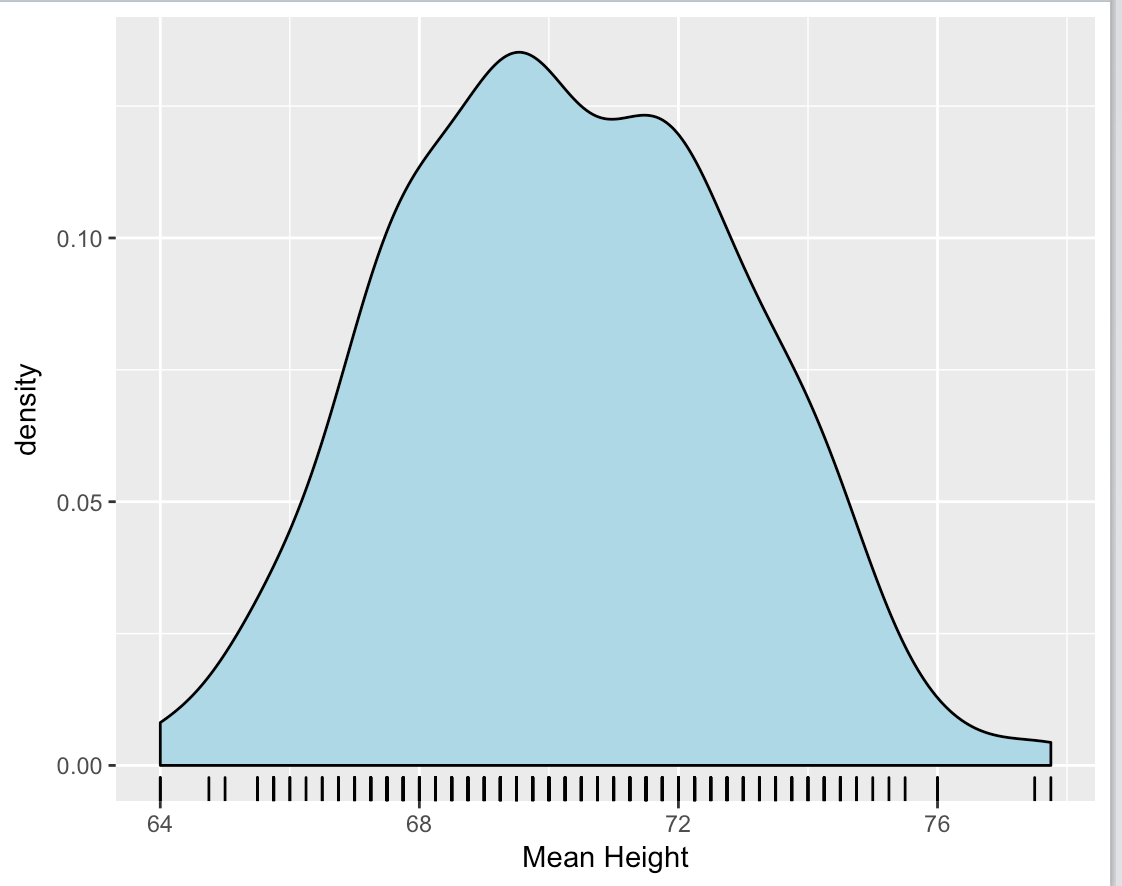


The next data analysis that I will conduct will be to perform binning on the variable “Height”. First I will conduct summary statistics on the Clean Data related to the variable “Height”.

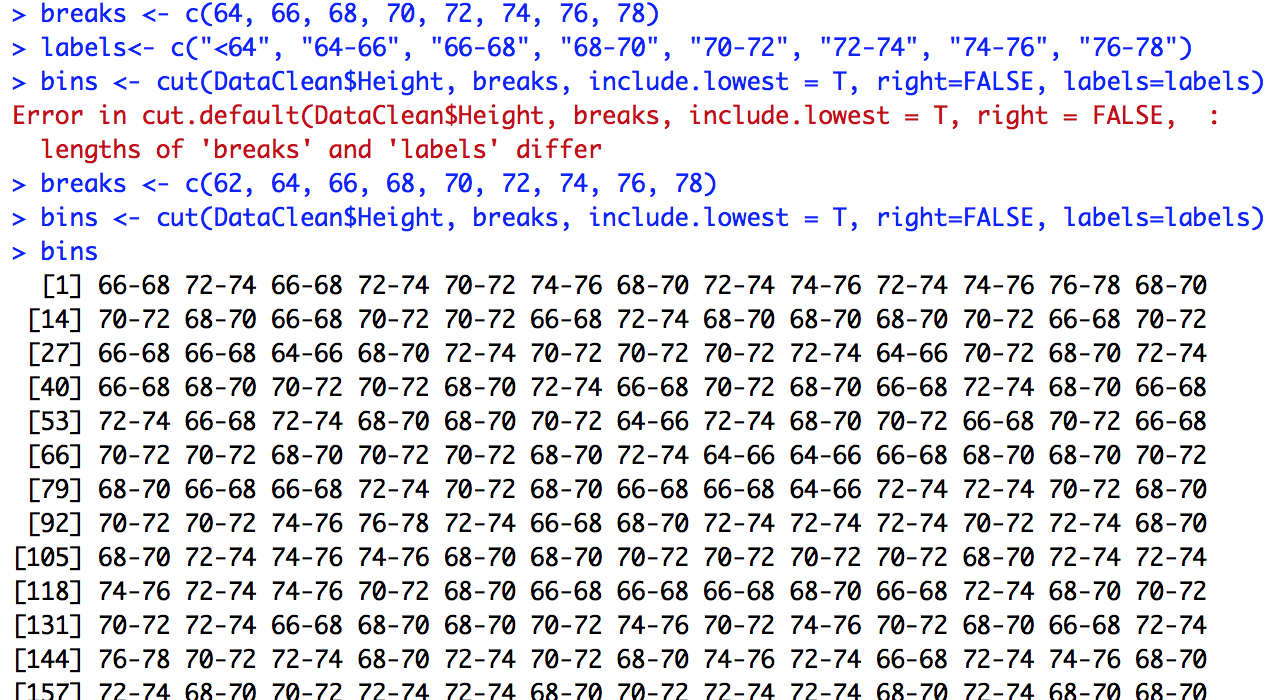


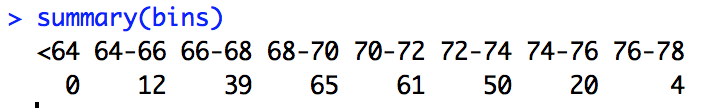
We then can see the summary statistics plotted using ggplot.





Since we now know the summary statistics we can create bins to see where all the data falls. The minimum height is 64.00 and maximum is 77.75. We need to create bins reflecting this range. We now create the variable breaks associated with the range data. I will break the data by increments of 2. I also create the variable labels to classify the different breaks into categories. We then read the data using the function summary(bins). Next we will graph this data.





This is the plot of the variable height binned in categories in increments of 2 inches. The data looks like it is normally distributed with no anomalies. It looks very similar to the ggplot() from our original statistical breakdown. This is just one more way we can look at the statistical data.

../../Screen%20Shot%202017-07-15%20at%2011.14.36%20AM.png

