STATS 419 Survey of Multivariate Analysis

Week 03 Assignment 03_datasets_revisited

Jailee Foster
(jailee.foster@wsu.edu)

Instructor: Monte J. Shaffer

22 September 2020

```
library(devtools); # devtools is required for function source_url() to work
my.source = 'github';
github.path = "https://raw.githubusercontent.com/jaileefoster/WSU_STATS419_FALL2020/";
source_url(paste0(github.path, "master/functions/libraries.R"));
```

1 Matrix

Create the "rotate matrix" functions as described in lectures. Apply to the example "myMatrix".

```
source_url(paste0(github.path, "master/functions/functions-matrix.R"));

myMatrix = matrix(c(1, 0, 2,
                    0, 3, 0,
                    4, 0, 5), nrow=3, byrow=T)
```

Note: in the file that contains the matrix functions (functions-matrix.R), there is a transformation matrix that is used in the functions. When a matrix is multiplied by this transformation matrix, its columns are reversed.

```
transposeMatrix(myMatrix);
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    4
## [2,]    0    3    0
## [3,]    2    0    5
```

```
rotateMatrix90(myMatrix);
```

```
##      [,1] [,2] [,3]
## [1,]    4    0    1
## [2,]    0    3    0
## [3,]    5    0    2
```

```
rotateMatrix180(myMatrix);
```

```
##      [,1] [,2] [,3]
## [1,]    5    0    4
## [2,]    0    3    0
## [3,]    2    0    1
```

```
rotateMatrix270(myMatrix);
```

```
##      [,1] [,2] [,3]
## [1,]    2    0    5
## [2,]    0    3    0
## [3,]    1    0    4
```

2 IRIS

2.1 Plot the Data Set

Recreate the graphic for the IRIS Data Set using R. Same titles, same scales, same colors. See: https://en.wikipedia.org/wiki/Iris_flower_data_set#/media/File:Iris_dataset_scatterplot.svg

```
data(iris)
pairs(iris[, 1:4], main="Iris Data (red=setosa,green=versicolor,blue=virginica)",
      bg=c("red", "springgreen3", "blue")[iris$Species], col="black", pch=21, cex.labels = 1, cex.main=
```

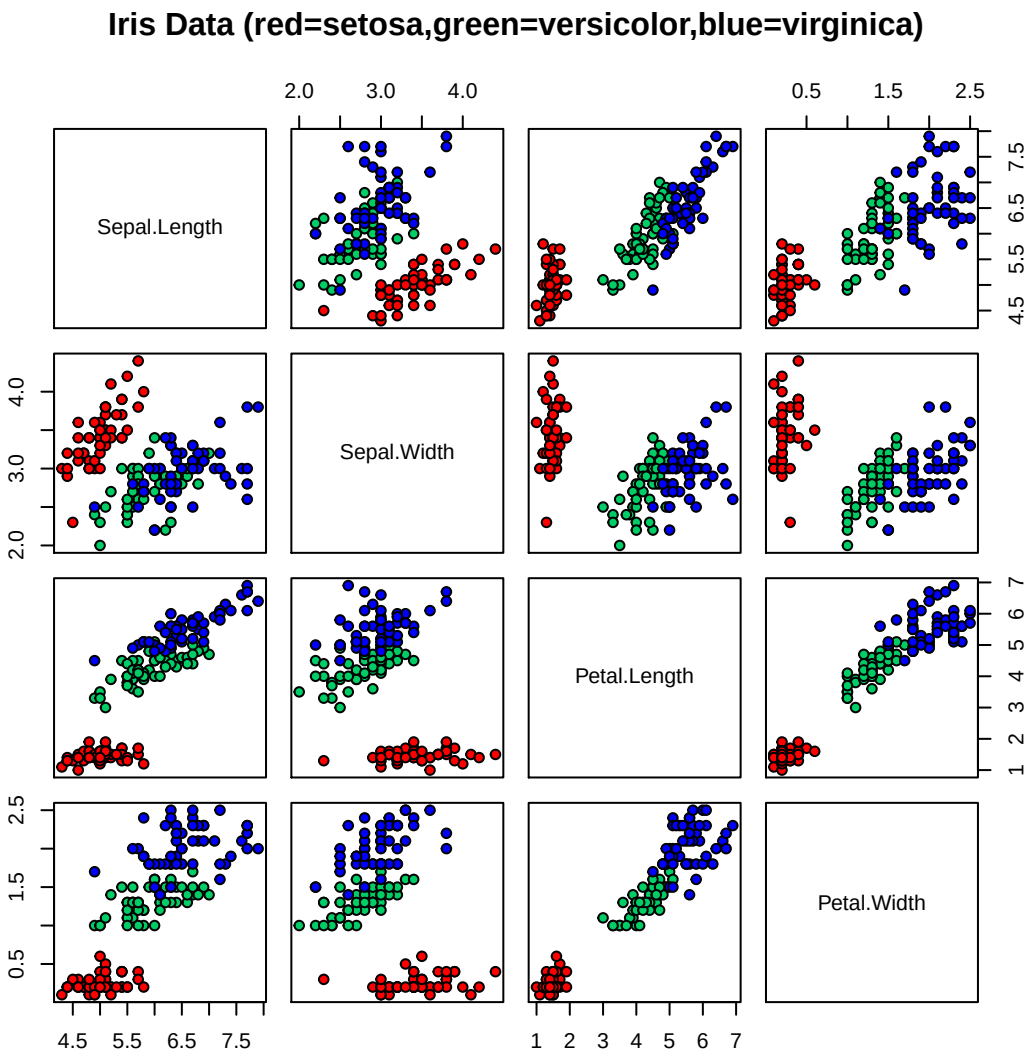


Figure 1: Scatterplot of IRIS Dataset: Wikipedia(2020)

In making the plot above, I referenced websites to add the title [1], change the font sizes [6], and change the format of the points [5].

2.2 Define the Data Set

Write 2-3 sentences concisely defining the IRIS Data Set. Be certain the final writeup are your own sentences (make certain you modify what you find, make it your own, but also cite where you got your ideas from).

The IRIS Data Set provides the following characteristics relating the iris flower: sepal length, sepal width, petal length, and petal width (all in centimeters). The data set also specifies which species each recording is. The data set itself is home to information about 50 samples each of 3 different species of iris flower, 150 samples in total. [2]

3 Personality

3.1 Cleanup Raw Data Set

Import "personality-raw.txt" into R. Remove the V00 column.

```
personality.raw = paste0(github.path, "master/datasets/personality/personality-raw.txt");
personality.data = read.csv(personality.raw, header=T, sep="|");
personality.data = subset(personality.data, select=-c(V00))
```

Create two new columns from the current column "date_test": year and week. Sort the new data frame by YEAR, WEEK so the newest tests are first ... The newest tests (e.g., 2020 or 2019) are at the top of the data frame. Then remove duplicates using the unique function based on the column "md5_email".

```
personality.data=personality.data[rev(order(as.Date(personality.data$date_test, format='%m/%d/%Y %H:%M')

date = strptime(personality.data$date_test, format='%m/%d/%Y %H:%M');
year = as.numeric(strftime(date, format="%Y"));
week = as.numeric(strftime(date, format="%W"));

personality.data$year = year;
personality.data$week = week;

personality.data = subset(personality.data, select=-c(date_test))

unique.by.email = unique(personality.data["md5_email"])
personality.data.clean = personality.data[!duplicated(personality.data["md5_email"]),]
```

The code above references a thread on Stack Overflow [3] to sort the dataframe and an article that explains how to remove duplicated data based on columns [4]. This code also references the code that Dr. Shaffer posted in the discussion board regarding cleaning the personality data set.

3.2 Export Clean Data Set

Save the data frame in the same "pipe-delimited format" (| is a pipe) with the headers. You will keep the new data frame as "personality-clean.txt" for future work (you will not upload it at this time).

```
write.table(personality.data.clean, "personality-clean.txt", sep="|")
```

3.3 Report Records

In the homework, for this tasks, report how many records your raw dataset had and how many records your clean dataset has.

```
dim(personality.data)
```

```
## [1] 838  63
```

```
dim(unique.by.email)
```

```
## [1] 678   1
```

```
dim(personality.data.clean)
```

```
## [1] 678  63
```

The raw dataset had 838 records and the clean dataset had 678 records. It is important to note that the number of unique emails found by the unique() command is also 678, implying that the duplicated() command was used properly above to remove duplicate email addresses.

4 Variance and Z-Scores

```
source_url(paste0(github.path, "master/functions/functions-variance.R"));
personality.vec = as.vector(personality.data.clean[1,])
personality.vec = as.numeric(subset(personality.vec, select=-c(md5_email, year, week)))
```

4.1 Variance

Write functions for doSummary and sampleVariance and doMode. Test these functions in your homework on the "monte.shaffer@gmail.com" record from the clean dataset. Report your findings.

```
doSummary(personality.vec)
```

```
##   length na.count mean median mode var.naive   var.2pass sd.builtin sd.custom
## 1     60        0 3.48   3.48  4.2 0.7528136 0.008786441  0.8676483 0.8676483
```

```
doMode(personality.vec)
```

```
## [1] 4.2
```

```
doSampleVariance(personality.vec, "naive")
```

```
##     sum sum.squared       var
## 1 208.8      771.04 0.7528136
```

```
doSampleVariance(personality.vec, "na")
```

```
##     sum   sum2         var
## 1 208.8 0.5184 0.008786441
```

```
zScores(personality.vec)
```

```
##  [1] -0.09220326  0.82982933 -1.01423585  0.82982933 -1.01423585 -1.01423585
##  [7]  0.82982933 -1.01423585 -0.09220326  0.82982933  0.82982933 -0.09220326
## [13] -0.09220326  0.82982933  1.75186192 -0.09220326  1.75186192 -0.09220326
## [19] -1.93626843 -1.01423585 -1.01423585 -1.01423585  0.82982933 -0.09220326
## [25]  1.75186192 -1.01423585  0.82982933 -0.09220326 -1.01423585 -1.01423585
## [31]  0.82982933 -1.93626843 -0.09220326  0.82982933  0.82982933  0.82982933
## [37] -1.01423585  0.82982933 -1.01423585  0.82982933  0.82982933  0.82982933
## [43]  0.82982933 -1.01423585  0.82982933  0.82982933 -1.01423585 -0.09220326
## [49] -1.01423585  0.82982933 -1.93626843  0.82982933 -1.01423585 -0.09220326
## [55]  0.82982933  0.82982933 -1.93626843  0.82982933 -1.01423585  0.82982933
```

Based on the results from performing the above functions on the "monte.shaffer@gmail.com" record from the clean dataset, the following conclusions can be drawn:

There are 60 entries, which correspond to the columns V01 to V60.

There are not any NA values in this data, which is expected because the data has been cleaned.

The mean and median test results are both 3.48, this suggests that the distribution of scores is symmetrical. This also suggests that the individual taking the test answer does not possess much bias, because 3.48 is almost exactly in the middle.

The mode tells us that the most common answer was 4.2.

The naïve variance was significantly larger than the two-pass variance, suggesting that there may have been a small amount of "outlier" answers that are significantly higher or lower than the other answers.

The built in function to find standard deviation, sd(), uses the naive variance method.

4.2 Z-Scores

For this "monte.shaffer@gmail.com" record, also create z-scores. Plot(x,y) where x is the raw scores for "monte.shaffer@gmail.com" and y is the z-scores from those raw scores. Include the plot in your assignment, and write 2 sentences describing what pattern you are seeing and why this pattern is present.

```
z.scores = zScores(personality.vec)
plot(personality.vec, z.scores, main = "Raw Personality Scores vs. Z-Scores", xlab = "Raw Personality S
```

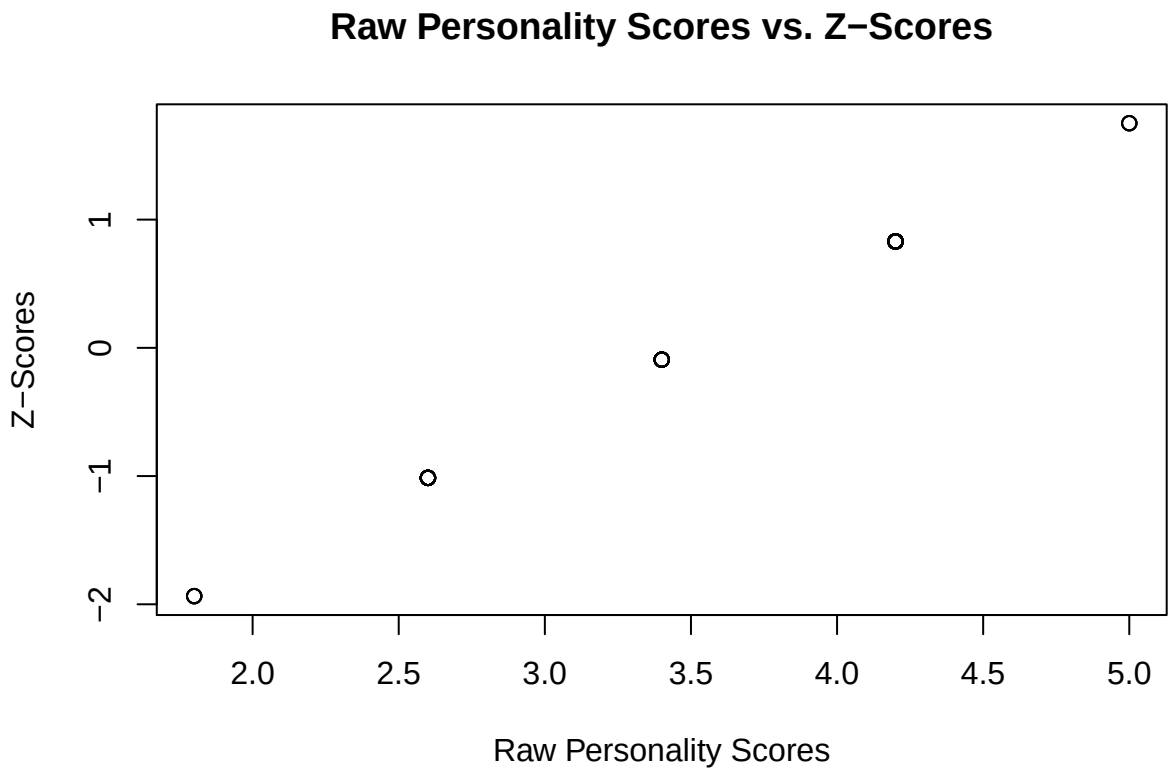


Figure 2: Raw Personality Scores vs. Z-Scores

By just looking at this plot, it is reasonable to think that there are only 5 data points. Upon further inspection, it seemed that many of the scores were repeated throughtout each of the vectors.

```
unique(z.scores)
```

```
## [1] -0.09220326  0.82982933 -1.01423585  1.75186192 -1.93626843
```

```
unique(personality.vec)
```

```
## [1] 3.4 4.2 2.6 5.0 1.8
```

To get a better idea of what was going on, I applied the unique() command to each vector and found that the personality vector only had 5 unique values, which corresponded to the 5 unique z-scores for those values. The points fall in a linear fashion, which makes sense because the same formula/ratio is used to find each z-score, there are essentially just different scalers.

5 Will vs. Denzel

```
source_url(paste0(github.path, "master/functions/functions-imdb.R"));
source_url(paste0(github.path, "master/functions/functions-inflation.R"));
```

Compare Will Smith and Denzel Washington.

5.1 Acquire necessary data from IMDB

5.1.1 Will Smith

```
nmid = "nm0000226";
  will = grabFilmsForPerson(nmid);
```

5.1.2 Denzel Washington

```
nmid = "nm0000243";
    denzel = grabFilmsForPerson(nmid);
```

5.2 Convert Raw Dollars to Dollars in 2000 to Account For Inflation

You will have to create a new variable millions.2000 that converts each movie's millions based on the year of the movie, so all dollars are in the same time frame. You will need inflation data from about 1980-2020 to make this work.

The following two chunks of code use the inflation table, result, acquired from the inflation() function to convert the "movies.50.millions" column into values that are all the same scale. The scale used is what 1,000,000 dollars was worth in the year 2000. I first created data frames from the information from grabFilmsForPerson() for each of the actors. This may have not been the most efficient way to do this, but it still worked. I then iterated through each value in the "movies.50.millions" column for each actor and expressed those numbers in dollars in relation to what 1,000,000 dollars was worth in 2000. Those values then got added to different vectors for each actor and added as new columns on the data frames called "will.millions.2000" and "denzel.millions.2000".

```
result= inflation()
```

```
## Warning in grabInflationData(): NAs introduced by coercion
```

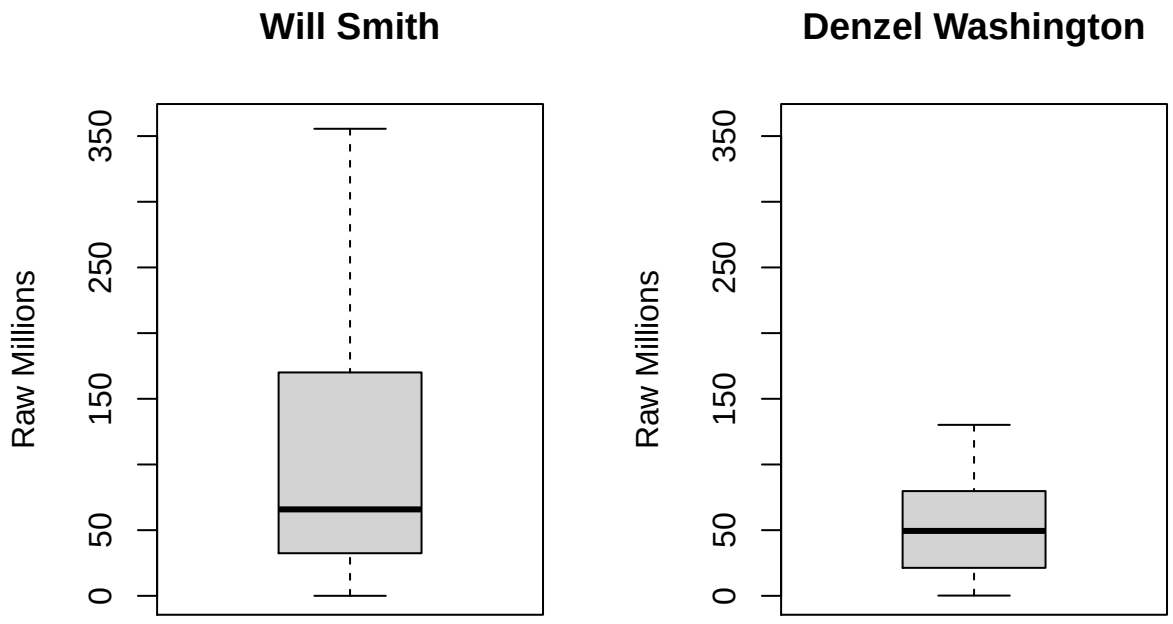
```
will.df = as.data.frame(will)

will.millions.2000 = c()

for (i in 1:50)
{
  line = as.numeric(will.df$movies.50.year) - 1919
  will.millions.2000[i] = will.df$movies.50.millions[i] * (result$dollars.2000[line[i]])/1000000
}

will.df$millions.2000 = will.millions.2000
will$millions.2000=will.millions.2000
```

```
denzel.df = as.data.frame(denzel)

denzel.millions.2000 = c()

for (i in 1:50)
{
  line = as.numeric(denzel.df$movies.50.year) - 1919
  denzel.millions.2000[i] = denzel.df$movies.50.millions[i] * (result$dollars.2000[line[i]])/1000000
}

denzel.df$millions.2000 = denzel.millions.2000
denzel$millions.2000=denzel.millions.2000
```

5.3 Side-by-Side Comparisons

5.3.1 Top 50 Movies Using Raw Dollars

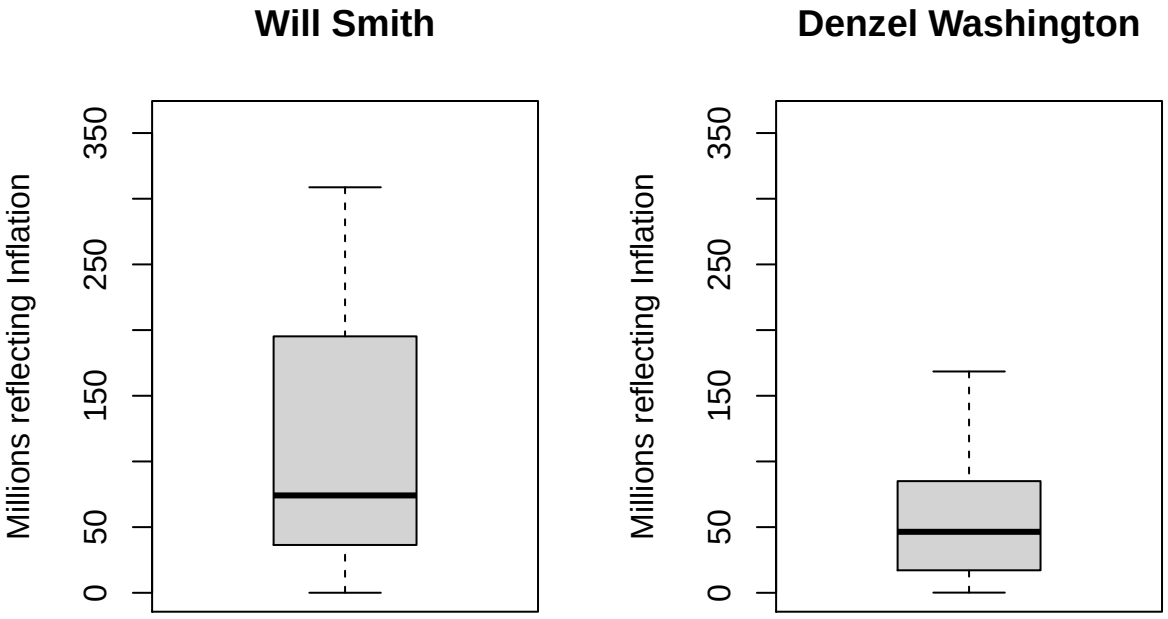
```
par(mfrow=c(1,2));

  boxplot(will$movies.50$millions, main=will$name, ylim=c(0,360), ylab="Raw Millions" );
    boxplot(denzel$movies.50$millions, main=denzel$name, ylim=c(0,360), ylab="Raw Millions" );
```



Based on these boxplots, it seems reasonable to conclude that the movies that Will Smith has acted in have brought in more money than those that Denzel Washington has acted in. The minimum amount for both actors looks very close to 0, but it looks as though Will Smith has had a few movies that have brought in a very large amount of money. The top whisker on Will Smith's plot is very long, showing that the higher movies are likely outliers. The spread of income generated by Will Smith's movies is much larger than that of Denzel Washingtons. Both box plots have medians in about the same area.

5.3.2 Top 50 Movies Using Adjusted Dollars (2000)

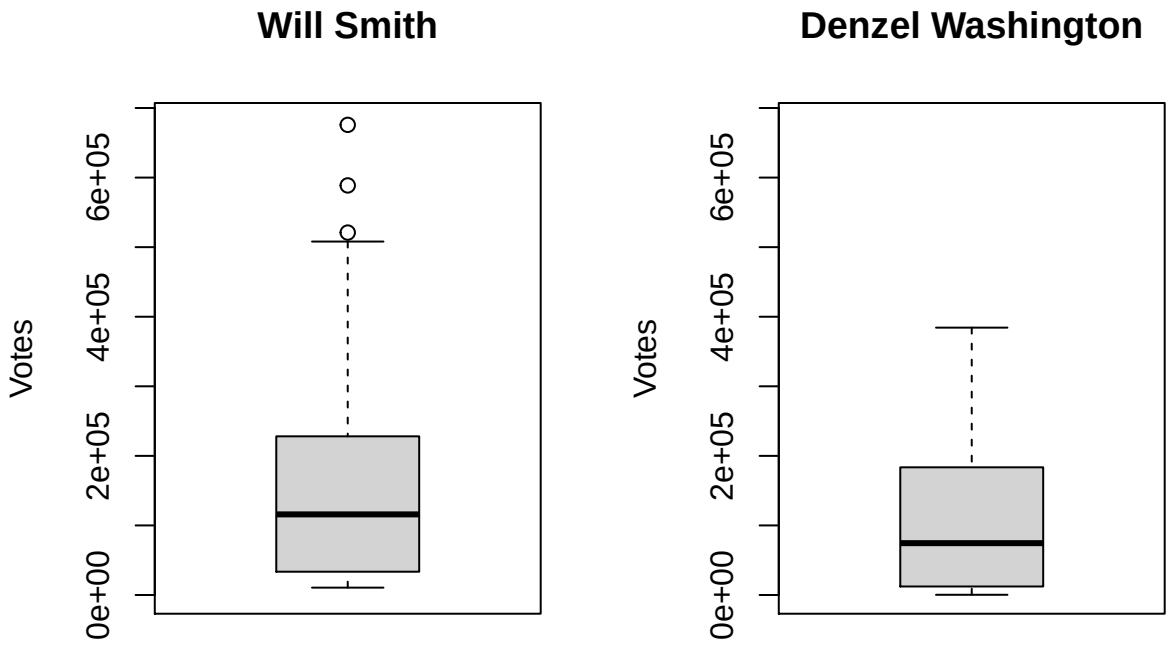
```
par(mfrow=c(1,2));

  boxplot(will$millions.2000, main=will$name, ylim=c(0,360), ylab="Millions reflecting Inflation", main
  boxplot(denzel$millions.2000, main=denzel$name, ylim=c(0,360), ylab="Millions reflecting Inflation", m
```



Based on these boxplots, I think that it is reasonable to say that Will Smith has brought in more money in the industry. Although it looks like the median for each of the actors is only about 25 million different, the maximum value (the top whisker), along with the third quartile are significantly higher for Will than they are for Denzel. The spread on the profit of movies that Will has been in is much higher than that of Denzel.

5.3.3 Total Votes

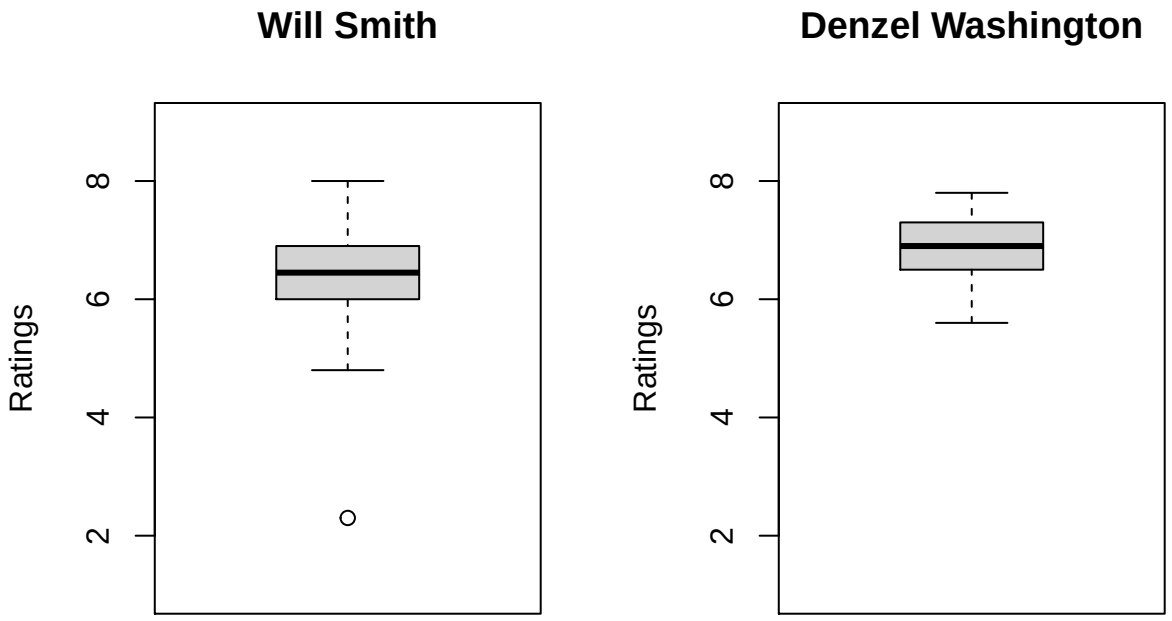
```
par(mfrow=c(1,2));

    boxplot(will$movies.50$votes, main=will$name, ylim=c(300,680000), ylab="Votes" );
    boxplot(denzel$movies.50$votes, main=denzel$name, ylim=c(300,680000), ylab="Votes" );
```



These boxplots are relatively similar. The main difference that can be seen are the outliers in Will Smith's plot, which all fall above 500,000 votes. Most of Will Smith's votes fall between 0 and 500,000, whereas Denzel Washington's only fall between 0 and 400,000. This can be found by looking at the whiskers of the plot. The median number of votes for Will Smith is slightly higher than that of Denzel Washington, but they both fall within +/- 25,000 of 100,000 votes. Based on the "box" in the plot, 50% of all movies for both actors look to have votes between about 25,000 and slightly over 200,000. Overall, Will Smith's first quartile, median, third quartile, and max are all larger than Denzel Washington's, suggesting that his movies received higher votes.

5.3.4 Average Ratings

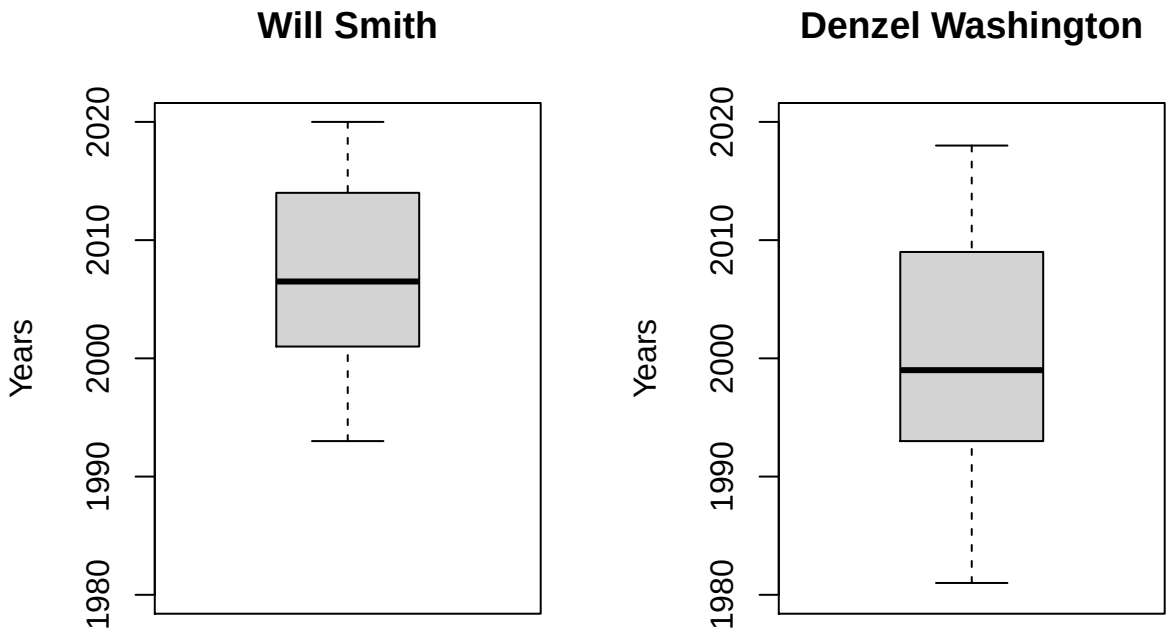
```
par(mfrow=c(1,2));

    boxplot(will$movies.50$ratings, main=will$name, ylim=c(1,9), ylab="Ratings" );
    boxplot(denzel$movies.50$ratings, main=denzel$name, ylim=c(1,9), ylab="Ratings" );
```



The plots for both actors are relatively symmetrical, suggesting there is not significant skewing in the data. Will Smith has one outlier rating slightly above a 2. Although Will Smith's maximum rating is higher that Denzel Washington's, it looks as though Denzel's ratings are more consistant because the spread of his boxplot is not as large as Will Smiths. Denzel Washington's ratings have a higher minimum, first quartile, median, and third quartile than Will Smith.

5.3.5 Year Released

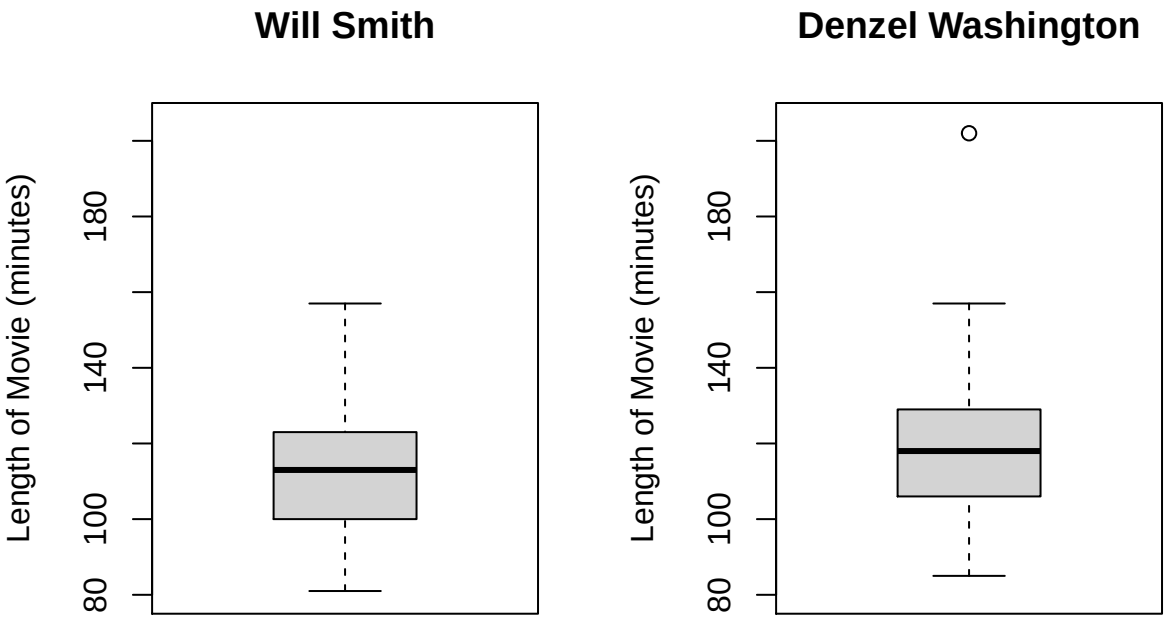
```
par(mfrow=c(1,2));

    boxplot(will$movies.50$year, main=will$name, ylim=c(1980,2020), ylab="Years" );
    boxplot(denzel$movies.50$year, main=denzel$name, ylim=c(1980,2020), ylab="Years" );
```



Based on these boxplots, I concluded that Will Smith's popular movies are generally more recent than Denzel Washington's. Denzel Washington also has much more of spread in years than Will Smith does based on the minimum and maximum shown by the whiskers. The median release year for Will Smith is somewhere around 2005, whereas the median for Denzel Washington looks to be around 1999. Both box plots are relatively symmetrical, signifying that the distribution is not overly skewed.

5.3.6 Length of Movie (in minutes)

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

  boxplot(will$movies.50$minutes, main=will$name, ylim=c(80, 205), ylab="Length of Movie (minutes)");
  boxplot(denzel$movies.50$minutes, main=denzel$name, ylim=c(80, 205), ylab="Length of Movie (minutes)")
```



These boxplots are relatively similar. The main difference that can be seen is the outlier in Denzel Washington's plot around the 200-minute mark. Other than the one outlier, it appears that the movies that both actors have been in range from about 80 minutes to 160 minutes, based on the spread of the whiskers. The median number of minutes for both actors looks to be between 100 and 125 minutes, and 50% of all movies for both actors look to be between 100 and 130 minutes based on the "box" part of the boxplot.

References

[1] de Vries, A. and J. Meys (2019). How to add titles and axis labels to a plot in r. (Accessed: September 20, 2020).
https://www.dummies.com/programming/r/how-to-add-titles-and-axis-labels-to-a-plot-in-r/.

[2] Dua, D. and C. Graff (2017). UCI machine learning repository: Iris data set.
https://archive.ics.uci.edu/ml/datasets/iris.

[3] I82Much (2011). How to sort a data frame by date. (Accessed: September 20, 2020).
https://stackoverflow.com/questions/6246159/how-to-sort-a-data-frame-by-date.

[4] Kassambra, A. (2018). Identify and remove duplicate data in r. (Accessed: September 20, 2020).
https://www.datanovia.com/en/lessons/identify-and-remove-duplicate-data-in-r/.

[5] Kassambra, A. (2019). Pch in r best tips. (Accessed: September 20, 2020).
https://www.datanovia.com/en/blog/pch-in-r-best-tips/.

[6] Martins, V. (2015). R pairs function - how to change the diagonal values font size? (Accessed: September 20, 2020).
https://stackoverflow.com/questions/30407332/r-pairs-function-how-to-change-the-diagonal-values-font-size.