STATS 419 Survey of Multivariate Analysis

Week 03 Assignment 03_datasets_revisited

Jailee Foster
(jailee.foster@wsu.edu)

Instructor: Monte J. Shaffer

21 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); # clockwise
```

```
##      [,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

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.axis=1, cex.main=1)
```

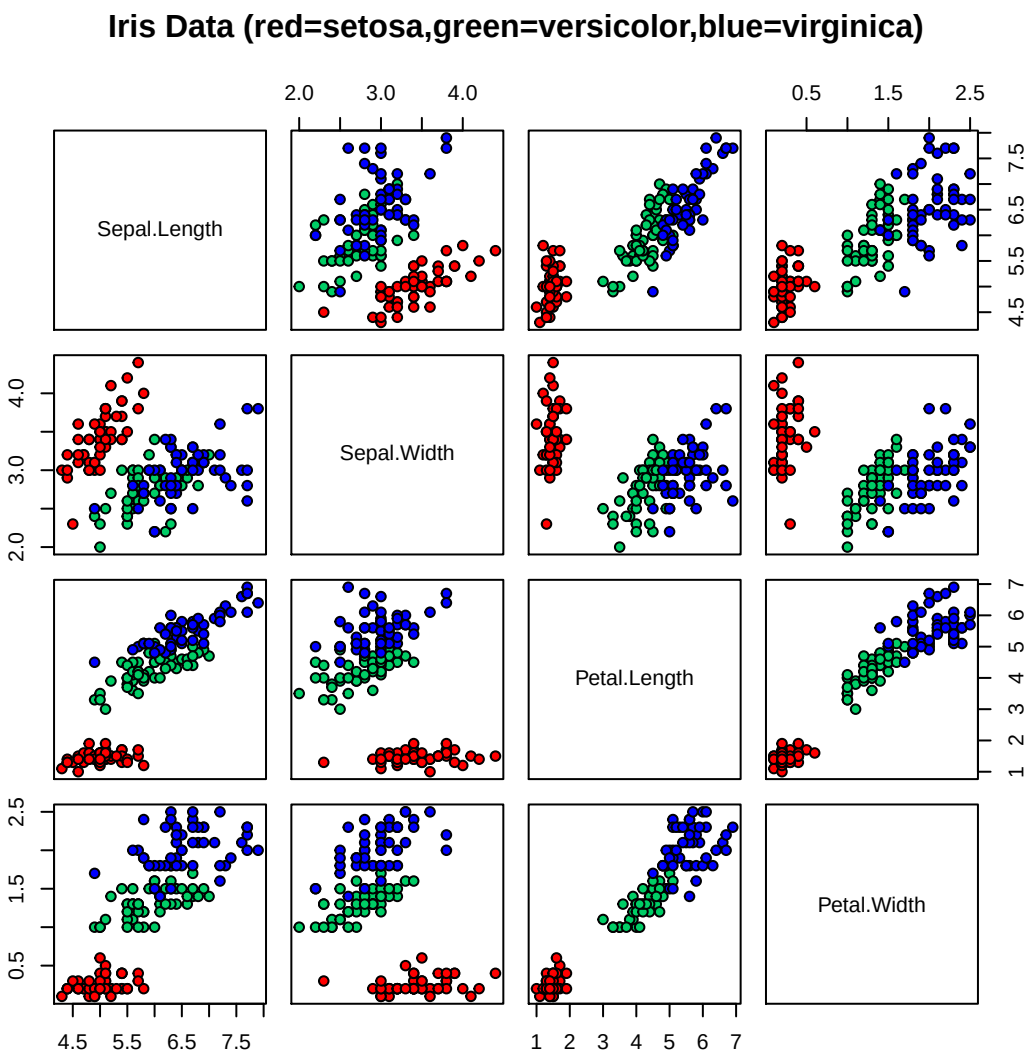


Figure 1: Scatterplot of IRIS Dataset: Wikipedia(2020)

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. [1]

3 Personality

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))
```

3.1 Cleanup Raw Data Set

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"]),]
```

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="|")
```

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.

4 Variance and Z-Scores

Write functions for doSummary and sampleVariance and doMode.

```
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

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)
```

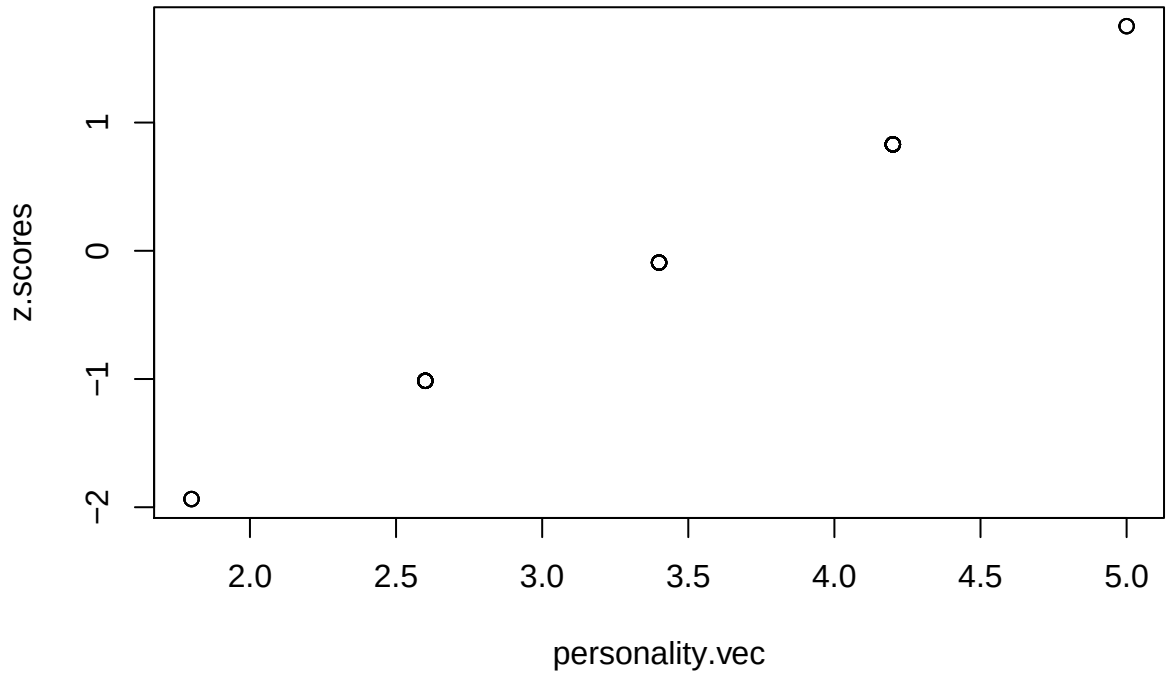
```
##  [1] -0.09220326  0.82982933 -1.01423585  0.82982933 -1.01423585 -1.01423585
##  [7]  0.82982933 -1.01423585 -0.09220326  0.82982933  0.82982933 -0.09220326
## [13] -0.09220326  0.82982933  1.75186192 -0.09220326  1.75186192 -0.09220326
## [19] -1.93626843 -1.01423585 -1.01423585 -1.01423585  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
```

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)
```



5 Will vs. Denzel

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

5.1 Will Smith

```
nmid = "nm0000226";
  will = grabFilmsForPerson(nmid);
  plot(will$movies.50[,c(1,6,7:10)]);
```

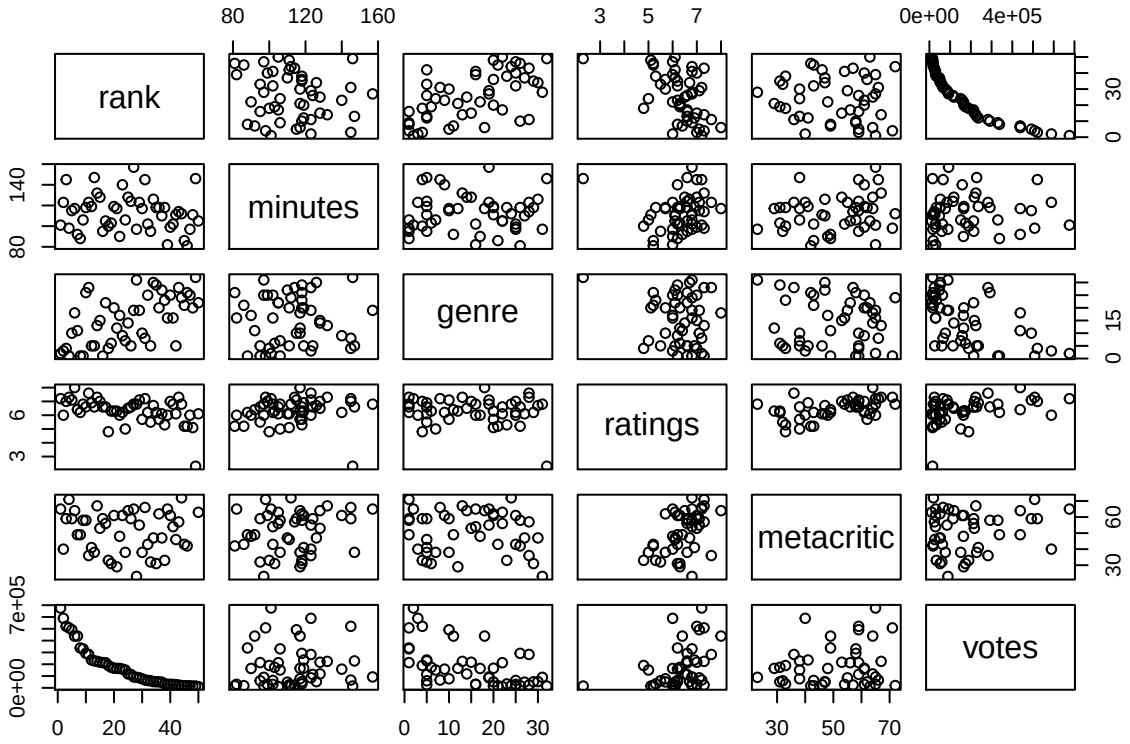


Figure 2: Will Smith scatterplot: IMDB(2020)

```
boxplot(will$movies.50$millions);
```

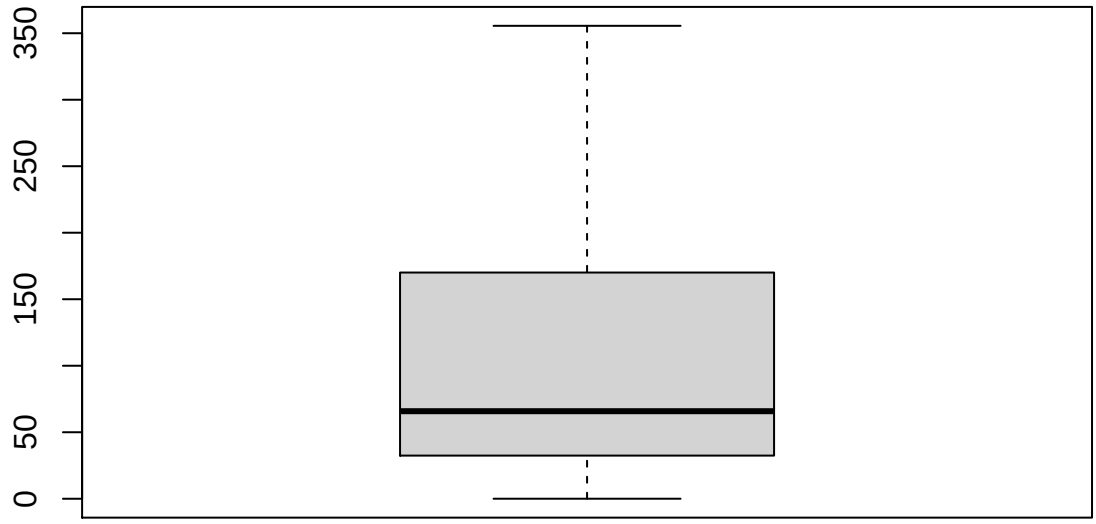


Figure 3: Will Smith boxplot raw millions: IMDB(2020)

```
   widx =  which.max(will$movies.50$millions);
will$movies.50[widx,];
```

```
##    rank   title      ttid year rated minutes                       genre ratings
## 15   15 Aladdin tt6139732 2019    PG     128 Adventure, Family, Fantasy       7
```

```
##    metacritic  votes millions
## 15         53 216922   355.56
```

```
    summary(will$movies.50$year);  # bad boys for life ... did data change?
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1993    2001    2006    2007    2014    2020
```

5.2 Denzel Washington

```
nmid = "nm0000243";
    denzel = grabFilmsForPerson(nmid);
    plot(denzel$movies.50[,c(1,6,7:10)]);
```

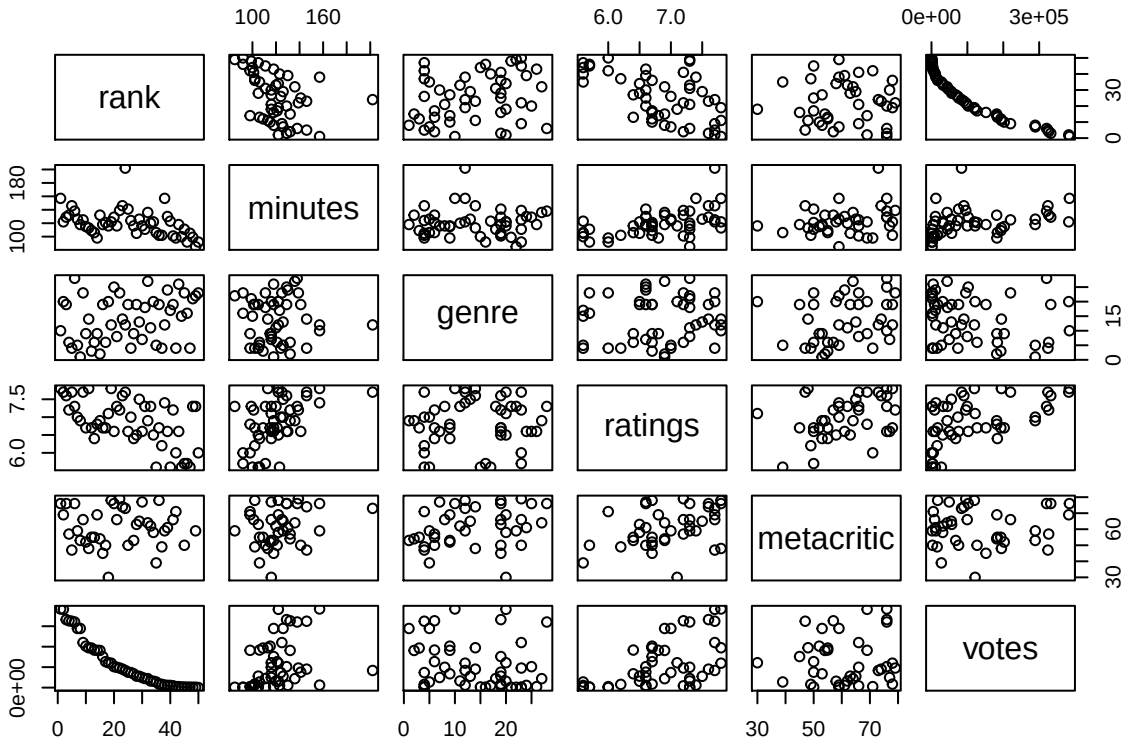


Figure 4: Denzel Washington scatterplot: IMDB(2020)

```
    boxplot(denzel$movies.50$millions);
```

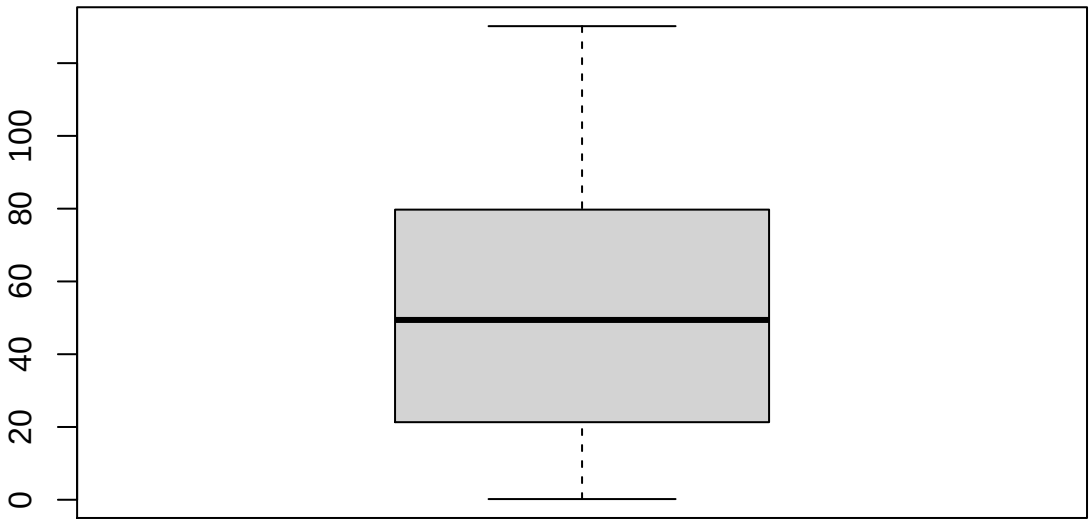


Figure 5: Denzel Washington boxplot raw millions: IMDB(2020)

```
        didx =  which.max(denzel$movies.50$millions);
    denzel$movies.50[didx,];
```

```
##   rank             title      ttid year rated minutes                    genre
## 1    1 American Gangster tt0765429 2007     R     157 Biography, Crime, Drama
##   ratings metacritic  votes millions
## 1     7.8         76 384284   130.16
```

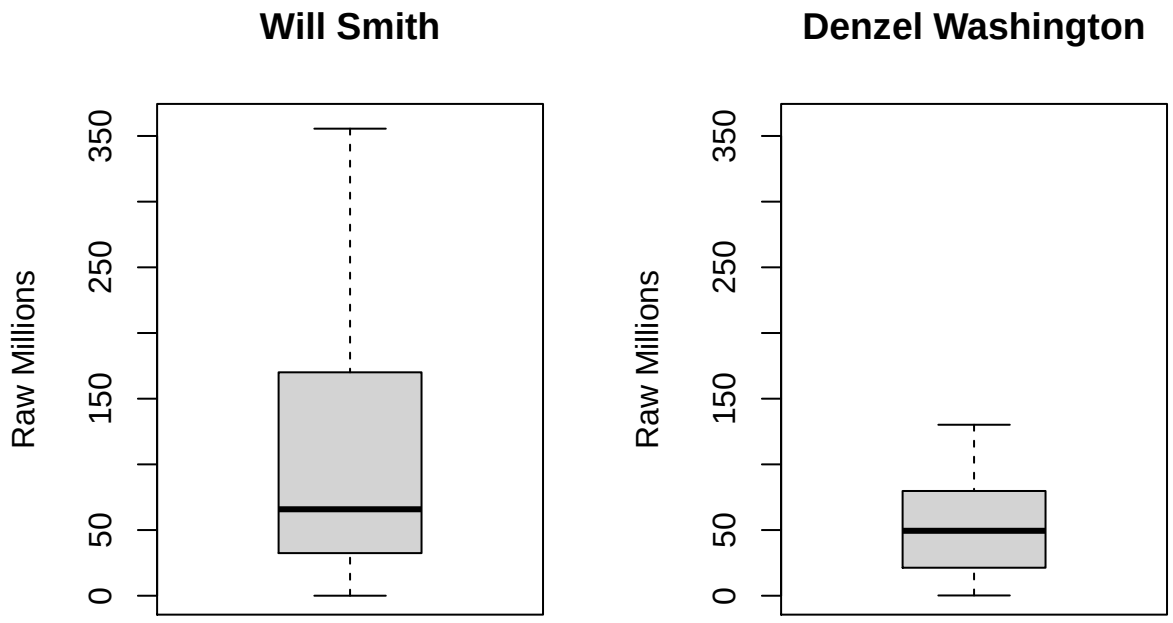
```
        summary(denzel$movies.50$year);
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1981    1993    1999    2000    2008    2018
```

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" );
```




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

5.3.2 Top 50 Movies Using Adjusted Dollars (2000)

```
local.path = "/Users/jaileefoster/Desktop/stat419/_git_/WSU_STATS419_FALL2020/";
source(paste0(local.path, "functions/functions-inflation.R"), local=T);

result= inflation()
```

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

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

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
```

```
local.path = "/Users/jaileefoster/Desktop/stat419/_git_/WSU_STATS419_FALL2020/";
source(paste0(local.path, "functions/functions-inflation.R"), local=T);

result= inflation()
```

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

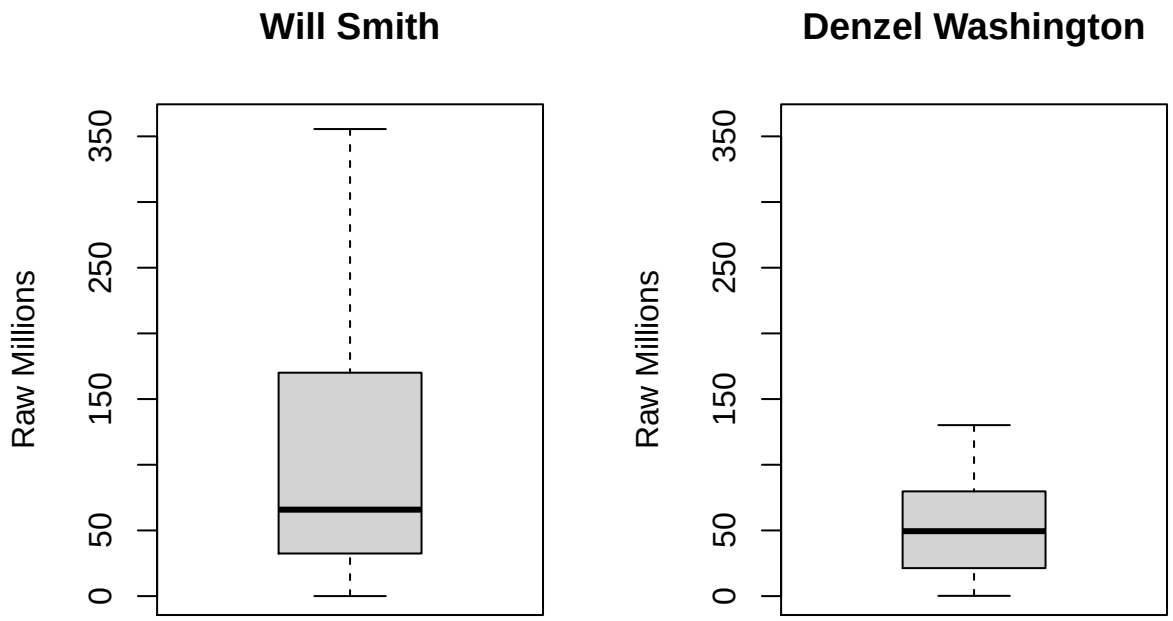
```
denzel.df = as.data.frame(denzel)
#denzel.df = as.data.frame(denzel)

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
```

```
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" );
```




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

5.3.3 Total Votes (Divide by 1,000,000 to scale)

5.3.4 Average Ratings

5.3.5 Year? Minutes?

5.3.6 Metacritic (NA values)

References

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