# STATS 419 Survey of Multivariate Anlaysis

Week 03 Assignment

Harrison Fuller
(harrison.fuller@wsu.edu)
[]

Instructor: Monte J. Shaffer

19 September 2020

```
library(devtools);
my.source = "local";
github.path = "https://raw.githubusercontent.com/fullerharrison/WSU_STATS419_FALL2020/";
source_url(paste0(github.path, "master/functions/libraries.R"))
```

## 1 Matrix

Create the "rotate matrix" functions described in lectures from the sample matrix. 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);
```

```
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)
```

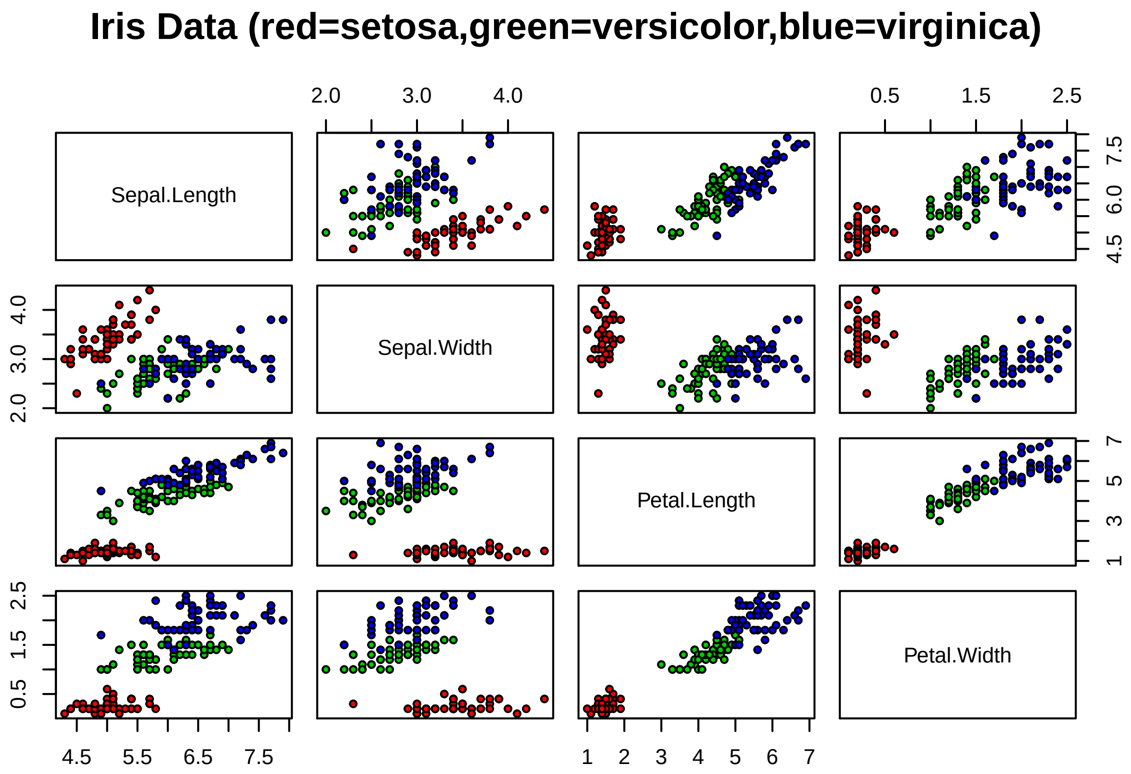
```
##      [,1] [,2] [,3]
## [1,]    2    0    5
## [2,]    0    3    0
## [3,]    1    0    4
```

## 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

```
# plot function
data("iris");

cols = c("red", "green3", "blue")
pairs(iris[1:4],
      main = "Iris Data (red=setosa,green=versicolor,blue=virginica)",
      pch = 21,
      bg = cols[unclass(iris$Species)], # unclass removes class attributes... why does this matter?
      cex = 0.7,
      cex.labels = 1.0,
      gap = 1)
```



Sentences: Right 2-3 sentences concisely defining the IRIS Data Set. Maybe search KAGGLE for a nice template. 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). NOTE: Watch the video, Figure 8 has a +5 EASTER EGG.

British biologist and statistician Ronald Fisher's 1936 paper "The use of multiple measurements in taxonomic problems" introduces a data set of 3 iris flower types (Setosa, Versicolour, and Virginica) and 5 attributes (sepal length, sepal width, petal length, petal width, and class) [1]. The data provided is one of the most widely used multivariate data sets for pattern recognition or classification with machine learning [2].

# 3 Personality

## 3.1 Cleanup RAW

Import "personality-raw.txt" into R. Remove the V00 column. Create two new columns from the current column "date_test": year and week. Stack Overflow may help: https://stackoverflow.com/questions/22439540/how-to-get-week-numbers-from-dates ... 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". 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). In the homework, for this tasks, report how many records your raw dataset had and how many records your clean dataset has.

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

[1] https://www.kaggle.com/arshid/iris-flower-dataset

[2] https://archive.ics.uci.edu/ml/datasets/iris

```
personality_raw <- read.table(paste0(github.path, "master/datasets/personality-raw.txt"), header = T, s

function(dat)
{
  test = dat %>%
    mutate(date_test = as.Date(dat$date_test, "%m/%d/%Y %H:%M")) %>% # create as a date
    arrange(desc(date_test)) %>%  # arrange by date
    mutate(V00 = lubridate::week(date_test)) %>%  # date to week
    mutate(date_test = lubridate:: year(date_test)) %>% # date to year
    rename(., year = date_test) %>% # rename year
    rename(. , week = V00) %>%  # rename week
    distinct(md5_email, .keep_all = T ) # find unique values by date
  return(test)

}
```

```
## function(dat)
## {
##   test = dat %>%
##     mutate(date_test = as.Date(dat$date_test, "%m/%d/%Y %H:%M")) %>% # create as a date
##     arrange(desc(date_test)) %>%  # arrange by date
##     mutate(V00 = lubridate::week(date_test)) %>%  # date to week
##     mutate(date_test = lubridate:: year(date_test)) %>% # date to year
##     rename(., year = date_test) %>% # rename year
##     rename(. , week = V00) %>%  # rename week
##     distinct(md5_email, .keep_all = T ) # find unique values by date
##   return(test)
##
## }
```

```
personality_clean <- cleanPersonalityData_tidy(personality_raw)
```

## 4 Varience and Z-score

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. 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.

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

doSummary = function(x)
{
  x <- as.vector(unlist(x));
  result <- data.frame(             matrix(ncol = 1,nrow = 11,
                                  dimnames = list( c("length",
                                                     "NAs",
                                                     "mean",
                                                     "median",
                                                     "mode",
                                                     "sum",
                                                     "sumSq",
                                                     "variance.naive",
                                                     "variance.two-pass",
                                                     "sd_custom",
                                                     "sd" ),
                                                   "") ));
  result["length",] = length(x);
  result["NAs", ] = sum(is.na(x));
  result["mean", ] = mean(x);
  result["median", ] = median(x);
  result["mode", ] = doMode(x);
  result["sum",] = doSampleVariance(x, method = "naive")[[1]];
  result["sumSq", ] = doSampleVariance(x, method = "naive")[[2]];
  result["variance.naive", ] = doSampleVariance(x, method = "naive")[[3]];
  result["variance.two-pass", ] = doSampleVariance(x, method = "two-pass")[[3]];
  result["sd_custom", ] = doSampleVariance(x, method = "naive")[[4]];
  result["sd", ] = sd(x);
  round(result, digits = 3);
}

doMode = function(x)
{
  # code adapted from : ^[https://www.r-bloggers.com/computing-the-mode-in-r/]
  result = c();
  table_of_values = table(x)
  table_of_max_values = max(table_of_values)
  if (all(table_of_values == table_of_max_values))
  {
    result = NA
  }
  else if(is.numeric(x))
  {
    result = as.numeric(names(table_of_values)[table_of_values == table_of_max_values])
  }
  else
  {
    result = names(table_of_values)[table_of_values == table_of_max_values]
  }
  result;
}
```

## 4.1 Variance

### 4.1.1 Naive & Traditional Two-Pass

```
doSampleVariance = function(x, method){

  if (method == "naive"){
    count <- Sum <- Sum2 <- 0
    x <- as.vector(unlist(x));
    for (i in x)
    {
      count = count + 1;
      Sum = Sum + i;
      Sum2 = Sum2 + (i* i);
    }
    if(count < 2) { return(NULL);} #

    variance = (Sum2 - (Sum * Sum) / count) / (count - 1)
    sd = sqrt(variance);

    list(Sum, Sum2, variance, sd)
  }
  else # two-pass
  {
    count <- Sum <- Sum2 <- 0;
    x <- as.vector(unlist(x));
    for (i in x)
    {
      count = count + 1;
      Sum = Sum + i;
    }
    if(count <  2) { return(NULL);} #
    mean = (Sum / count);
    for (j in x) # second pass
    {
      Sum2 = Sum2 + ((j - mean) * (j - mean));
    }
    variance = (Sum2 / (count - 1));
    sd = sqrt(variance);

    list(Sum, Sum2, variance, sd)
  }
}
```

## 4.2 Z-score

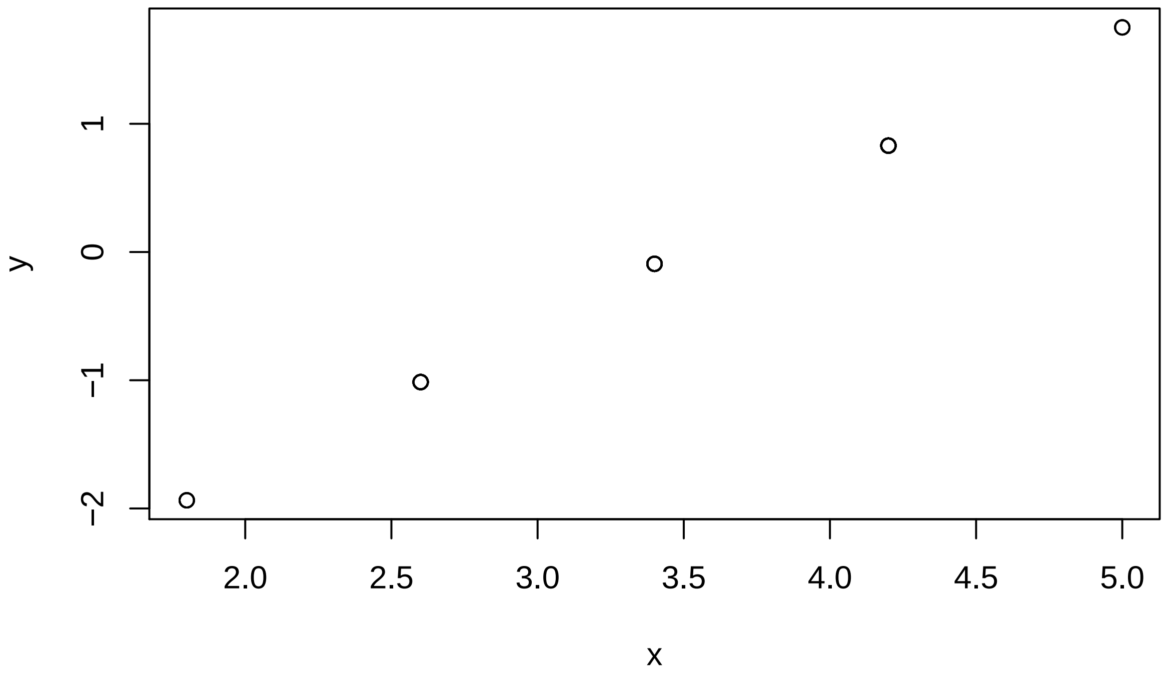
```
m.shaffer = "b62c73cdaf59e0a13de495b84030734e"

data <- personality_clean[personality_clean$md5_email == m.shaffer, 4:63];
doSummary(data);
```

```
##                        V1
## length             60.000
## NAs                 0.000
## mean                3.480
## median              3.400
## mode                4.200
## sum               208.800
## sumSq             771.040
## variance.naive      0.753
## variance.two-pass   0.753
## sd_custom           0.868
## sd                  0.868
```

```
x <- t(data);
y <- scale(t(data));

plot(x, y);
```



The plot generated from scaled “monte.shaffer@gmail.com” data indicates normal distribution of questions answered using the likert scale. The theoretical correlation is near perfect showing a linear relationship between raw scores and normalized scores.

# 5 Will vs Denzel

Compare Will Smith and Denzel Washington. [See 03_n greater 1-v2.txt for the necessary functions and will-vs-denzel.txt for some sample code and in DROPBOX: /**student_access**/unit_01_exploratory_data_analysis/week_02/imdb-example ] 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.

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

## 5.1 Will Smith

```
nmid = "nm0000226";
    will = grabFilmsForPerson(nmid);  ## can we source for dput?
  source_url("http://md5.mshaffer.com/WSU_STATS419/will");  # look at syntax
                                                            # will = data;
```

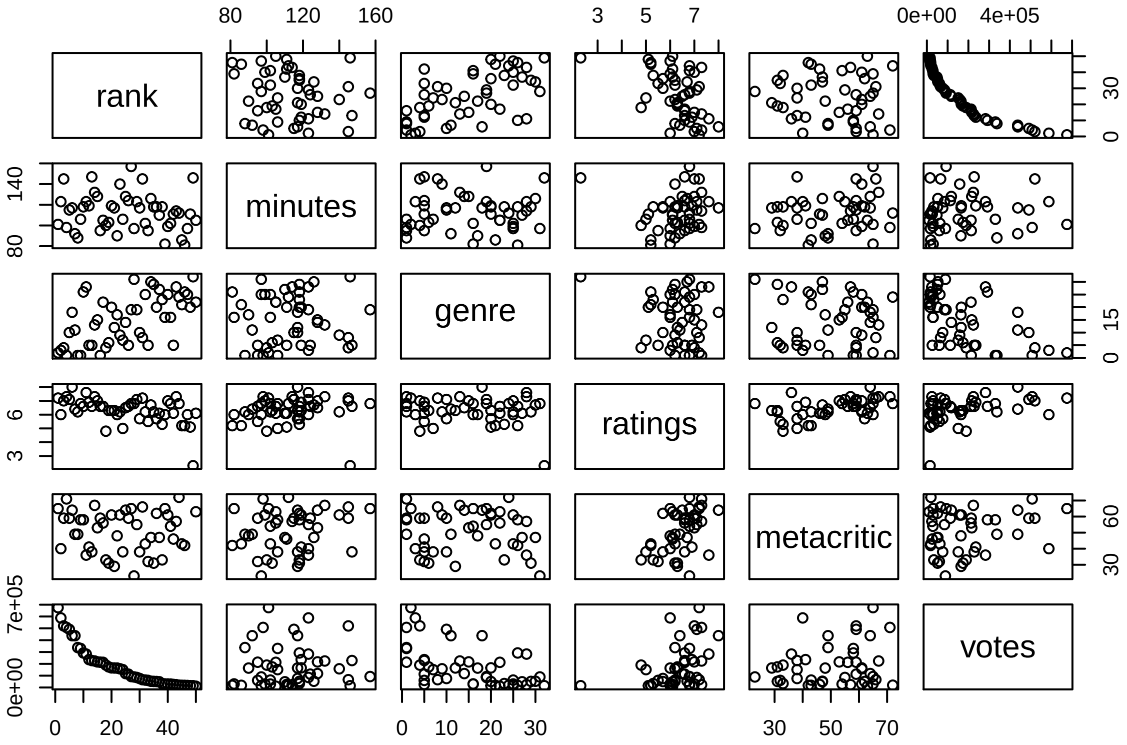


Figure 1: Will Smith Scatterplot:IMDB(2020)

```
plot(will$movies.50[,c(1,6,7:10)]);
```

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

```
    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 216296   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

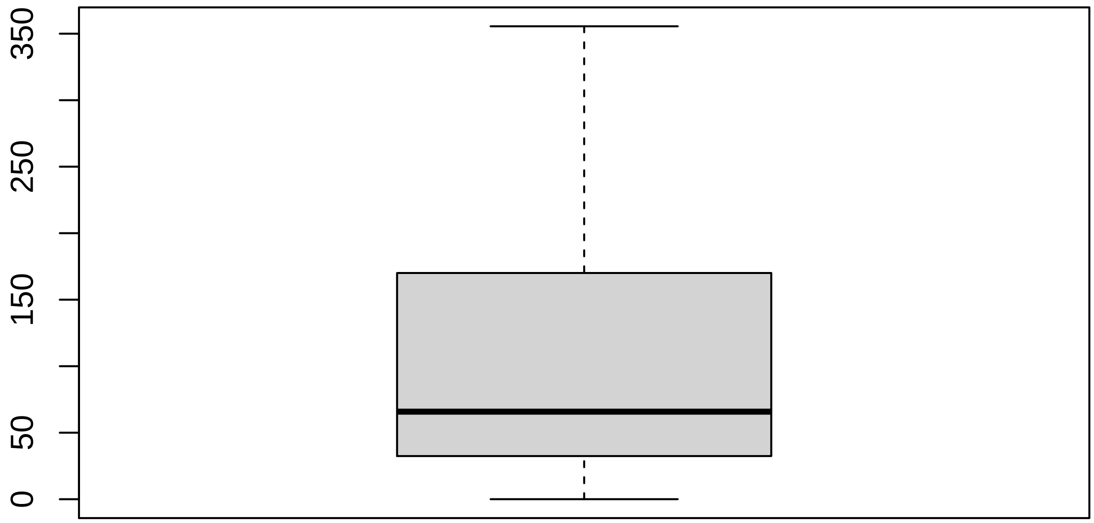


Figure 2: Will Smith Boxplot:IMDB(2020)

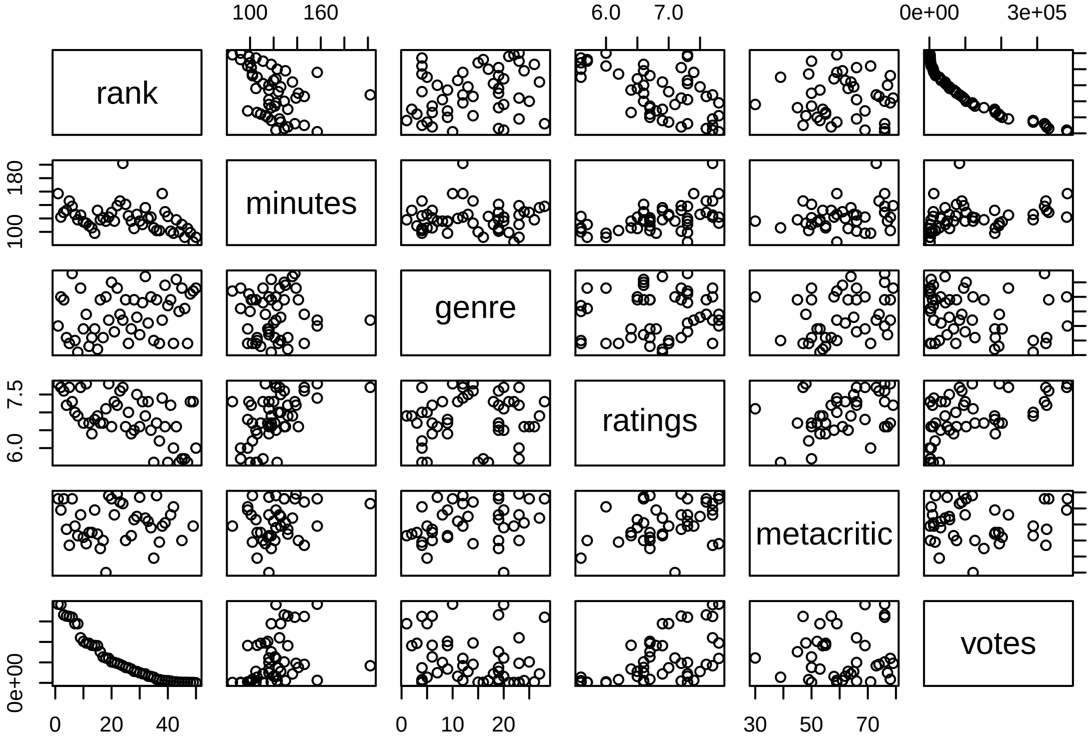


Figure 3: Denzel Washington Scatterplot:IMDB(2020)

```
nmid = "nm0000243";
    denzel = grabFilmsForPerson(nmid);  ## can we source for dput?

    plot(denzel$movies.50[,c(1,6,7:10)]);
```

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

```
        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 384212   130.16
```

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

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

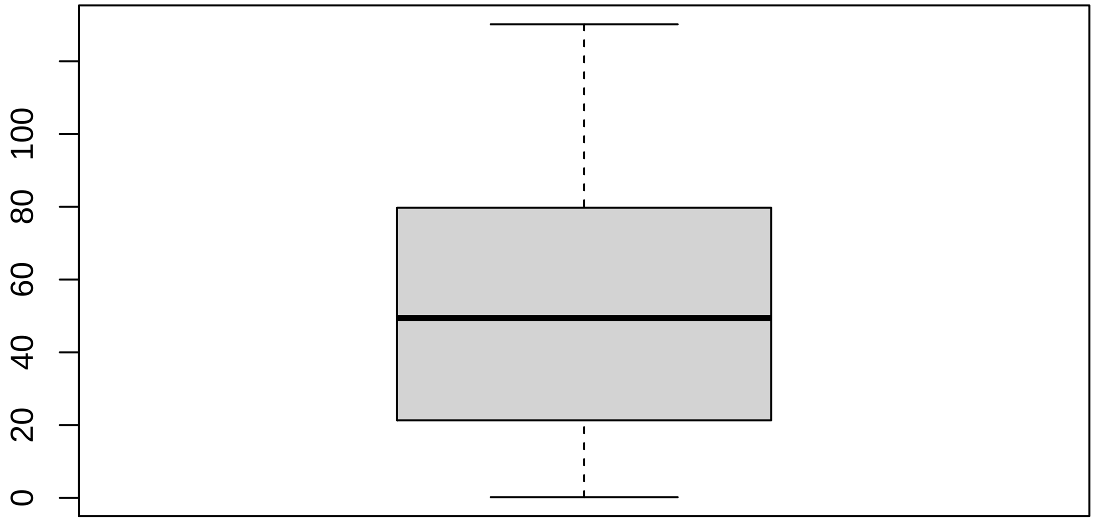


Figure 4: Denzel Washington Boxplot:IMDB(2020)

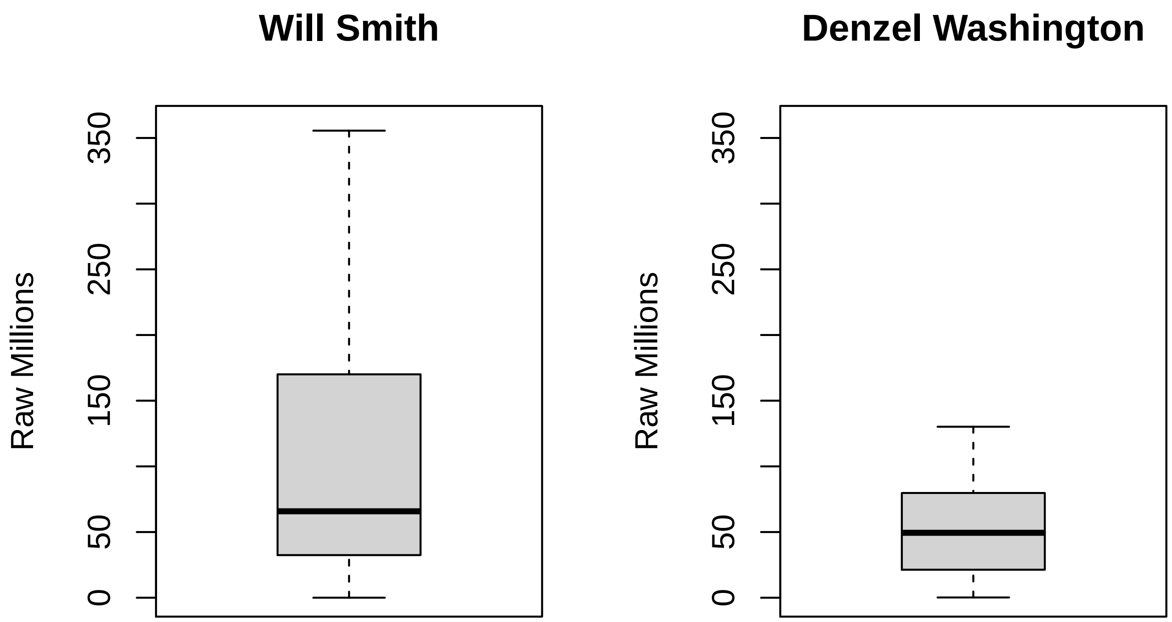


Figure 5: Will Smith and Denzel Washington Raw Millions Boxplot:IMDB(2020)

```
denzel$name
```

```
## [1] "Denzel Washington"
```

### 5.3 BoxPlot of 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" );
```

### 5.4 Side-by-Side Comparisons

Build side-by-side box plots on several of the variables (including #6) to compare the two movie stars. After each box plot, write 2+ sentence describing what you are seeing, and what conclusions you can logically make. You will need to review what the box plot is showing with the box portion, the divider in the box, and the whiskers.

#### 5.4.1 Adjusted Dollars (2000)

```
source_url("https://raw.githubusercontent.com/MonteShaffer/humanVerseWSU/master/humanVerseWSU/R/functio
```

```
## SHA-1 hash of file is 51bc90666b6b997894ed179c8f46019bb2517bad
```

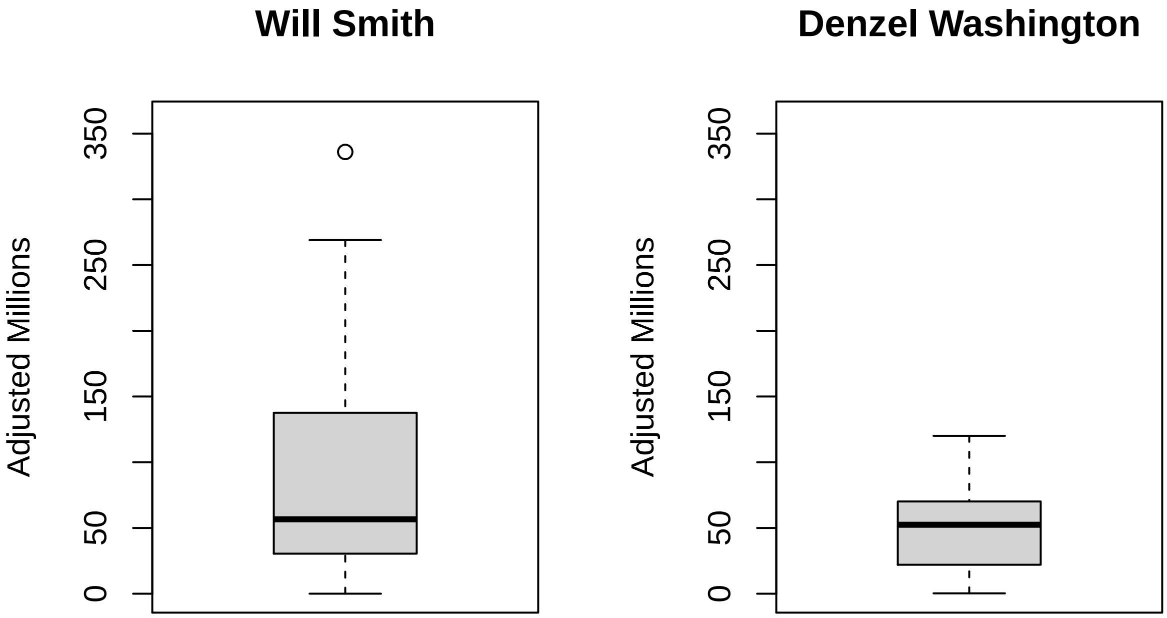
```
inflation.df <- read.table(paste0(github.path, "master/datasets/inflation.txt"), header = T, sep = "|")

denzel$movies.50 = standardizeDollarsInDataFrame(denzel$movies.50, 2000, "millions", "year", "millionsA

will$movies.50 = standardizeDollarsInDataFrame(will$movies.50, 2000, "millions", "year", "millionsAdj")

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

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



The boxplot above shows a side-by-side comparison of Will Smith and Denzel Washington movie earnings (adjusted for inflation) over each actors top 50 movies respectively. Will Smith has similar success in the average of film earnings with a high degree of variability for movies earning more than his average. The film earnings of Denzel Washington demonstrate much less variability across his film earnings.

### 5.4.2 Film Lengths (Minutes)

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

# will
```

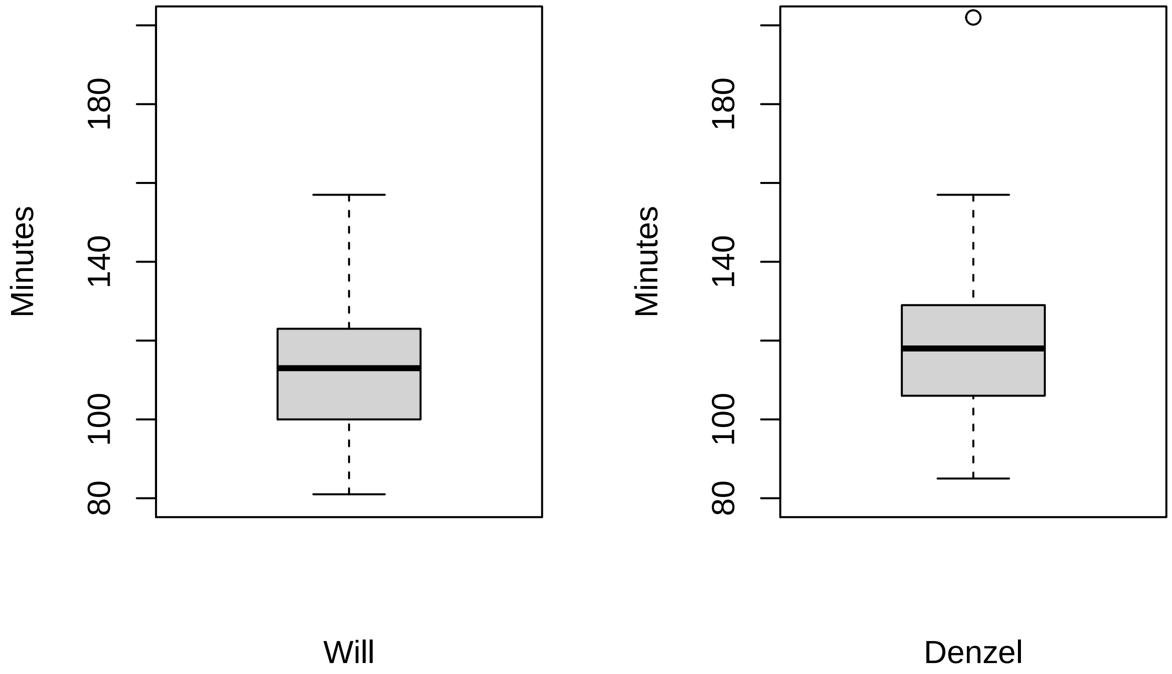


Figure 6: Will Smith and Denzel Washington Film Lengths Boxplot:IMDB(2020)

```
boxplot(will$movies.50$minutes,  xlab = "Will", ylab = "Minutes", ylim = c(80,200));

# denzel
boxplot(denzel$movies.50$minutes, xlab = "Denzel",ylab = "Minutes", ylim = c(80,200))
```

The above boxplot illustrates the distribution of film times for each respective actor. The distribution of Denzel Washington film lengths indicate fairly uniform distribution around 120 minutes with the exception of an outlier from his film Malcolm X. The boxplot of Will Smith film lengths show some variability in film lengths that are shorter than his median length film. Denzel Washington appears to have marginally longer film lengths when acting in movies.

### 5.4.3 Metacritic (NA values)

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

# will
boxplot(na.omit(will$movies.50$metacritic),  xlab = "Will", ylab = "Metacritic", ylim = c(0,100));

# denzel
boxplot(na.omit(denzel$movies.50$metacritic), xlab = "Denzel",ylab = "Metacritic", ylim = c(0,100))
```

The above bloxplots illustrates the distribution of metacritic scores for each respective actor with NA values omitted from each data set. Denzel Washington has some positive skenwess of critic score distribution

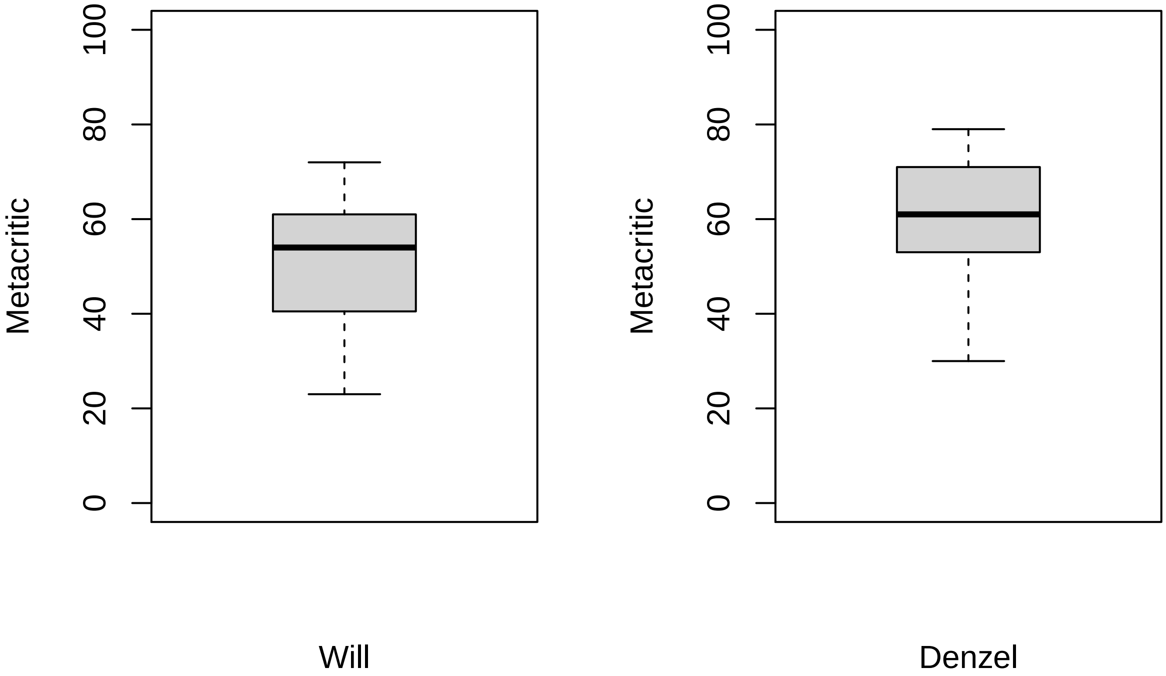


Figure 7: Will Smith and Denzel Washington Metacritic Scores Boxplot:IMDB(2020)

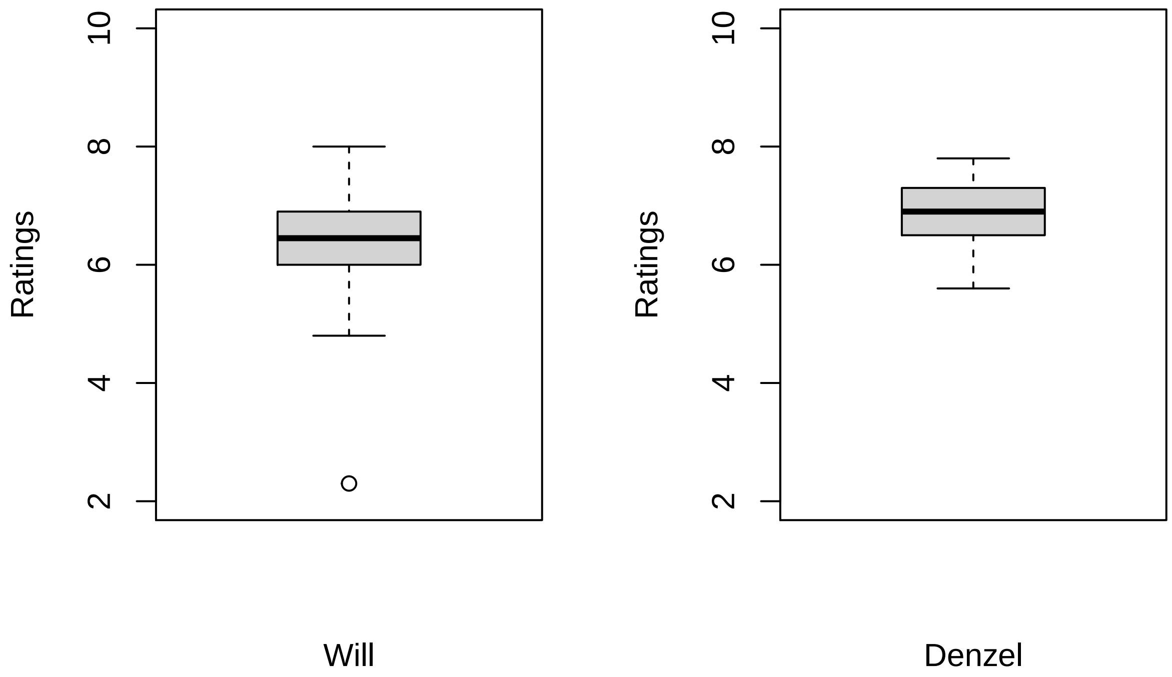


Figure 8: Will Smith and Denzel Washington Average Ratings of Top 50 Films Boxplot:IMDB(2020)

for his film contributions. Will Smith appears to have substantial negative skewness in critic score ratings. Denzel Washington appears to be more liked in overall film contributions.

#### 5.4.4 Average Ratings

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

# will
boxplot(will$movies.50$ratings,  xlab = "Will", ylab = "Ratings", ylim = c(2,10));

# denzel
boxplot(denzel$movies.50$ratings, xlab = "Denzel",ylab = "Ratings", ylim = c(2,10))
```

The boxplot above illustrates the average ratings for the top 50 films of each actor. The distribution of either actor appears to have uniform distribution of average ratings across a small range of potential scores. Will Smith has one outlier in which shows a low score for the film Virtuosity. From this illustration, it is difficult to assume any real advantage one actor has over another.

#### 5.4.5 Year

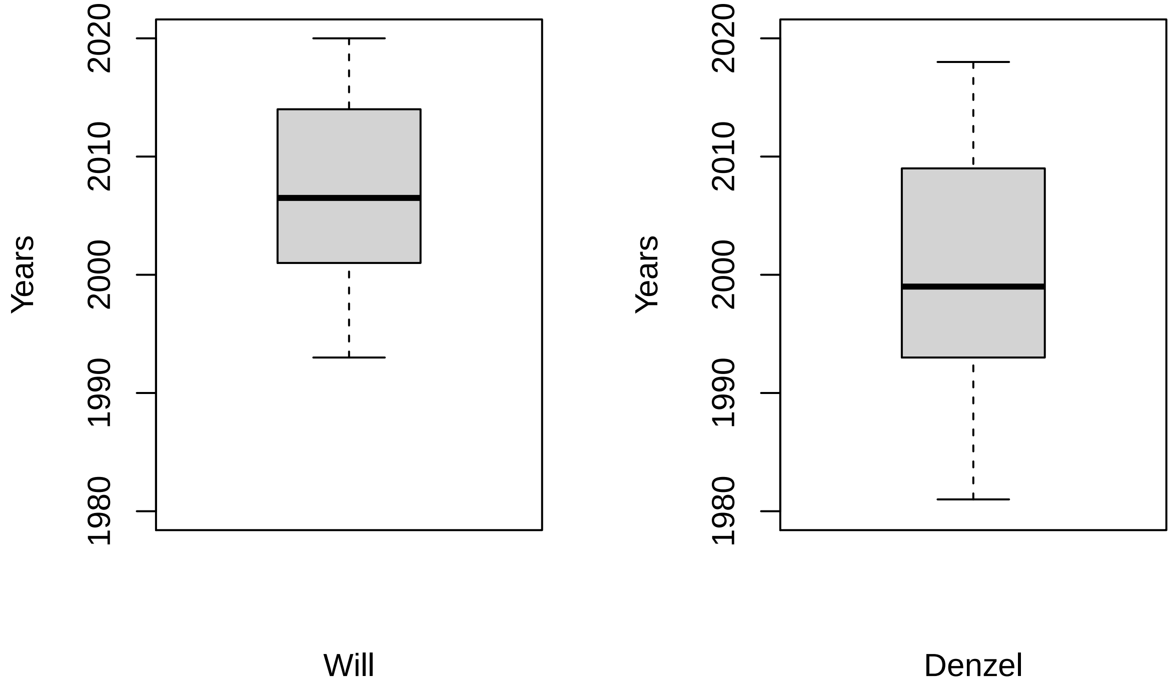


Figure 9: Will Smith and Denzel Washington Year Range of Top 50 Films Boxplot:IMDB(2020)

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

# will
boxplot(will$movies.50$year,  xlab = "Will", ylab = "Years", ylim = c(1980, 2020));

# denzel
boxplot(denzel$movies.50$year, xlab = "Denzel",ylab = "Years", ylim = c(1980, 2020))
```

The boxplot above illustrates the year of release for each film occupying the top 50 list of Will Smith and Denzel Washington, respectively. The distribution of Denzel movie releases indicate a long history of movies considered worthy of the top 50 list. In comparison, a large proportion of Will Smith top 50 movies have been released since the year 2000. One assumption that can be made from this illustration is that Will Smith has gained in popularity more recently and as a result has a larger proportion of movies occupying his top 50 list after the year 2000.