Statistics 419 Survey of Multivariate Analysis

Week 3 Datasets Assignment Revisited

Connor StarrHurst
connor.starrhurst@wsu.edu
11596221

Instructor: Monte J. Shaffer

September 21, 2020

```
library(devtools); #for source_url()
library(tibble) #for add_column() under 4. PERSONALITY DATA
library(lubridate) #for year() and week() under 4. PERSONALITY DATA

my.source = 'github';

github.path = "https://raw.githubusercontent.com/youknowwwho/stats419/";
source_url( paste0(github.path,"master/Functions/libraries.R") );
source_url( paste0(github.path,"master/Functions/functions-imdb.R") );

#source_url("https://raw.githubusercontent.com/youknowwwho/stats419/master/Functions/libraries.R");
#source_url("https://raw.githubusercontent.com/youknowwwho/stats419/master/Functions/functions-imdb.R")

my.source = 'local';

local.path = "C:/Users/Connor/.ssh/stats419/";
local.data.path = "";
source( paste0(local.path,"Functions/libraries.R") );
```

1 Rotate Matrix

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

```
source( paste0(local.path,"Week 3/Week 3 Functions/functions rotate matrix.R") );

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

yourMatrix = matrix( c(
  1, 2, 3,
  4, 5, 6,
```

```
  7, 8, 9
), nrow=3, byrow=T);

myMatrix
```

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

```
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
```

```
multiply3x3Matrix(myMatrix, yourMatrix);
```

```
##      [,1] [,2] [,3]
## [1,]   15   18   21
## [2,]   12   15   18
## [3,]   39   48   57
```

2 IRIS Graphic

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

```
pairs(iris[1:4], #use IRIS list data to create a matrix of scatterplots
      main = "Iris Data (red=setosa,green=versicolor,blue=virginica)", #title
      pch = 21, #set plot characters to be filled circles
      bg = c("red", "green3", "blue")[unclass(iris$Species)]); #set the background/fill color of the pl
```

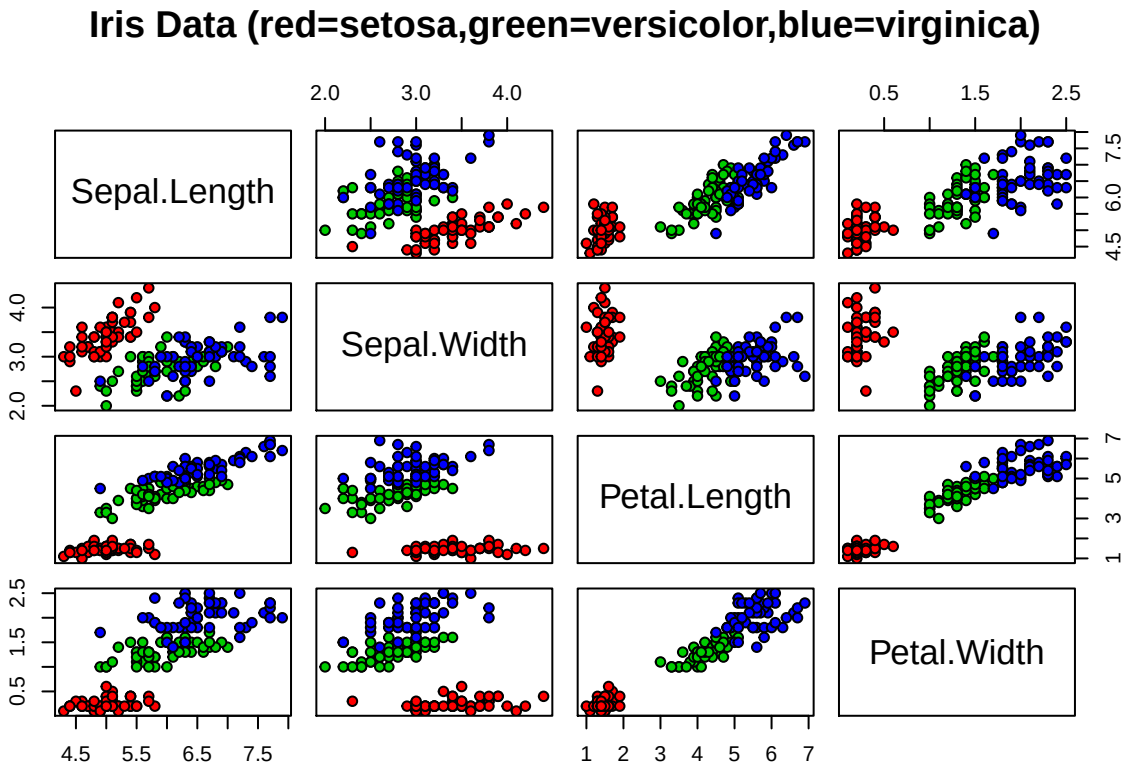


Figure 1: IRIS Data

3 IRIS Sentences

Write 2-3 sentences concisely defining the IRIS Data Set. Maybe search KAGGLE for a nice template. Be certain the final write up are your own sentences (make certain you modify what you find, make it your own, but also cite where you got your ideas from).

One of the first things I noticed when looking at the data is that the virginica species is overall a larger plant. It has the largest petals (in length and width) and longest sepals overall. Conversely, the setosa species has the smallest petals (in length and width) with the shortest sepal length.

4 Personality Data

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 task, report how many records your raw dataset had and how many records your clean dataset has.

```
data4 = read.delim("C:\\Users\\Connor\\Documents\\1) WSU 2018-\\Fall 2020\\Stat 419\\Week 2\\personality
data4 = data4[c(1:2,4:63)]; #remove "VOO" column
data4 = add_column(data4, year = NA, .after = 2); #add a year column
data4 = add_column(data4, week = NA, .after = 3); #add a week column

for(i in 1:nrow(data4))
{
  date = data4[i,2];
  date = strptime(date, format = "%m/%d/%Y %H:%M"); #convert date string to POSIXlt/POSIXt
  data4[i,3] = year(date); #get the year and put it in the year column
  data4[i,4] = week(date); #get the week and put it in the week column
}

data4 = data4[order(-data4$year, -data4$week),]; #sort the data descending by year first, then week
#dab = unique(data4, incomparables = F, MARGIN = 1, fromLast = F) #793 & 794 are duplicates (ece0c3bd12
#dab = data4 %>% distinct(data4£md5_email, .keep_all = T);
data4 = data4[!duplicated(data4$md5_email),]; #remove duplicates
write.table(data4, "personality-clean.txt", sep = "|", row.names = T); #save a cleaned version of the d

head(data4);
```

```
##                            md5_email       date_test year week V01 V02 V03 V04
## 838 b62c73cdaf59e0a13de495b84030734e  4/6/2020 12:57 2020   14 3.4 4.2 2.6 4.2
## 837 1358d38e6898b1a0e5940f8b99ba2325 12/1/2019 22:12 2019   48 3.4 3.4 3.4 4.2
## 828 bfd1c69406d322d17312e965752813c2  5/2/2019 10:26 2019   18 2.6 4.2 1.0 4.2
## 829 9cf05d7d516099c9533b98beb91993b9  5/2/2019 10:48 2019   18 5.0 5.0 1.8 5.0
## 830 a6544303c18e090ae3452aa266ecb2c0  5/2/2019 14:11 2019   18 3.4 3.4 2.6 2.6
## 831 ed03a1da7edce96ccd4f614d210a13e2  5/2/2019 17:45 2019   18 3.4 3.4 1.8 5.0
##     V05 V06 V07 V08 V09 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20 V21 V22 V23
## 838 2.6 2.6 4.2 2.6 3.4 4.2 4.2 3.4 3.4 4.2 5.0 3.4 5.0 3.4 1.8 2.6 2.6 2.6 4.2
## 837 4.2 4.2 5.0 3.4 4.2 3.4 2.6 3.4 3.4 4.2 4.2 4.2 4.2 4.2 3.4 2.6 3.4 4.2 4.2
## 828 4.2 2.6 3.4 1.8 4.2 4.2 1.8 2.6 3.4 5.0 4.2 4.2 5.0 4.2 3.4 1.8 1.0 3.4 4.2
## 829 5.0 4.2 1.0 5.0 5.0 5.0 1.0 5.0 5.0 5.0 5.0 5.0 5.0 3.4 3.4 3.4 4.2 5.0 5.0
## 830 5.0 3.4 3.4 3.4 5.0 2.6 1.8 3.4 2.6 2.6 4.2 4.2 4.2 4.2 2.6 2.6 1.0 3.4 4.2
## 831 4.2 4.2 5.0 5.0 5.0 5.0 1.0 5.0 5.0 3.4 5.0 5.0 5.0 5.0 5.0 1.0 1.0 5.0 5.0
##     V24 V25 V26 V27 V28 V29 V30 V31 V32 V33 V34 V35 V36 V37 V38 V39 V40 V41 V42
## 838 3.4 5.0 2.6 4.2 3.4 2.6 2.6 4.2 1.8 3.4 4.2 4.2 4.2 2.6 4.2 2.6 4.2 4.2 4.2
## 837 4.2 2.6 4.2 4.2 3.4 2.6 4.2 4.2 3.4 4.2 3.4 4.2 5.0 3.4 4.2 4.2 4.2 4.2 4.2
## 828 3.4 1.8 4.2 5.0 3.4 1.8 4.2 3.4 4.2 4.2 3.4 4.2 3.4 1.8 5.0 3.4 4.2 1.8 2.6
## 829 3.4 1.8 5.0 5.0 4.2 3.4 5.0 5.0 4.2 5.0 5.0 5.0 5.0 2.6 3.4 5.0 4.2 5.0 5.0
## 830 5.0 1.8 4.2 2.6 4.2 1.8 4.2 3.4 3.4 4.2 3.4 4.2 4.2 1.8 3.4 3.4 4.2 4.2 3.4
## 831 5.0 1.0 5.0 5.0 5.0 1.0 5.0 5.0 4.2 5.0 5.0 5.0 5.0 4.2 5.0 5.0 5.0 3.4 5.0
##     V43 V44 V45 V46 V47 V48 V49 V50 V51 V52 V53 V54 V55 V56 V57 V58 V59 V60
## 838 4.2 2.6 4.2 4.2 2.6 3.4 2.6 4.2 1.8 4.2 2.6 3.4 4.2 4.2 1.8 4.2 2.6 4.2
## 837 4.2 4.2 3.4 4.2 4.2 2.6 3.4 4.2 3.4 4.2 4.2 4.2 4.2 3.4 4.2 4.2 3.4 3.4
## 828 4.2 4.2 4.2 4.2 4.2 2.6 3.4 1.8 2.6 2.6 5.0 4.2 3.4 2.6 2.6 4.2 4.2 4.2
## 829 3.4 5.0 5.0 5.0 5.0 3.4 5.0 5.0 5.0 5.0 5.0 5.0 1.0 2.6 3.4 5.0 5.0 3.4
## 830 4.2 3.4 4.2 5.0 3.4 3.4 3.4 3.4 3.4 5.0 4.2 4.2 3.4 3.4 2.6 4.2 2.6 4.2
## 831 5.0 5.0 5.0 5.0 4.2 5.0 4.2 3.4 5.0 5.0 5.0 5.0 5.0 5.0 5.0 5.0 1.8 5.0
```

The raw dataset had 838 records and my clean dataset has 678 records.

5 Do, Variance, and Z-scores

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.

5.0.1 Do Functions

```
source( paste0(local.path,"Week 3/Week 3 Functions/functions do, variance, & z-scores.R") );

record = data4[1,]; #get the monte.shaffer@gmail.com record (b62c73cdaf59e0a13de495b84030734e 4/6/2020

doSummary(record);
```

```
## Length =  64
## Number of NAs =  0
## Mean =  3.48
## Median =  3.4
## Mode =  4.2
## Variance =  0.7528136
## Standard Deviation =  0.8676483
```

```
doMode(record);
```

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

5.0.2 Variance Functions

```
doSampleVariance(record, 1); #return sum, sumSquared, and variance
```

```
## [[1]]
##  [1] -0.08  0.72 -0.88  0.72 -0.88 -0.88  0.72 -0.88 -0.08  0.72  0.72 -0.08
## [13] -0.08  0.72  1.52 -0.08  1.52 -0.08 -1.68 -0.88 -0.88 -0.88  0.72 -0.08
## [25]  1.52 -0.88  0.72 -0.08 -0.88 -0.88  0.72 -1.68 -0.08  0.72  0.72  0.72
## [37] -0.88  0.72 -0.88  0.72  0.72  0.72  0.72 -0.88  0.72  0.72 -0.88 -0.08
## [49] -0.88  0.72 -1.68  0.72 -0.88 -0.08  0.72  0.72 -1.68  0.72 -0.88  0.72
##
## [[2]]
## [1] 44.416
##
## [[3]]
## [1] 0.7528136
```

```
doSampleVariance(record, "naive"); #return sum, sumSquared, and variance
```

```
## [[1]]
## [1] 12.85067
##
## [[2]]
## [1] 12.1104
##
## [[3]]
## [1] 0.7528136
```

5.0.3 Z-Scores and Plot

Sentences.

6 Will vs Denzel

```
source( paste0(local.path,"Functions/functions-imdb.R") );
```

6.0.1 Will Smith

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

#pairs(will$movies.50[c(1,6,8:10)]);
plot(will$movies.50[,c(1,6,8:10)]);
```

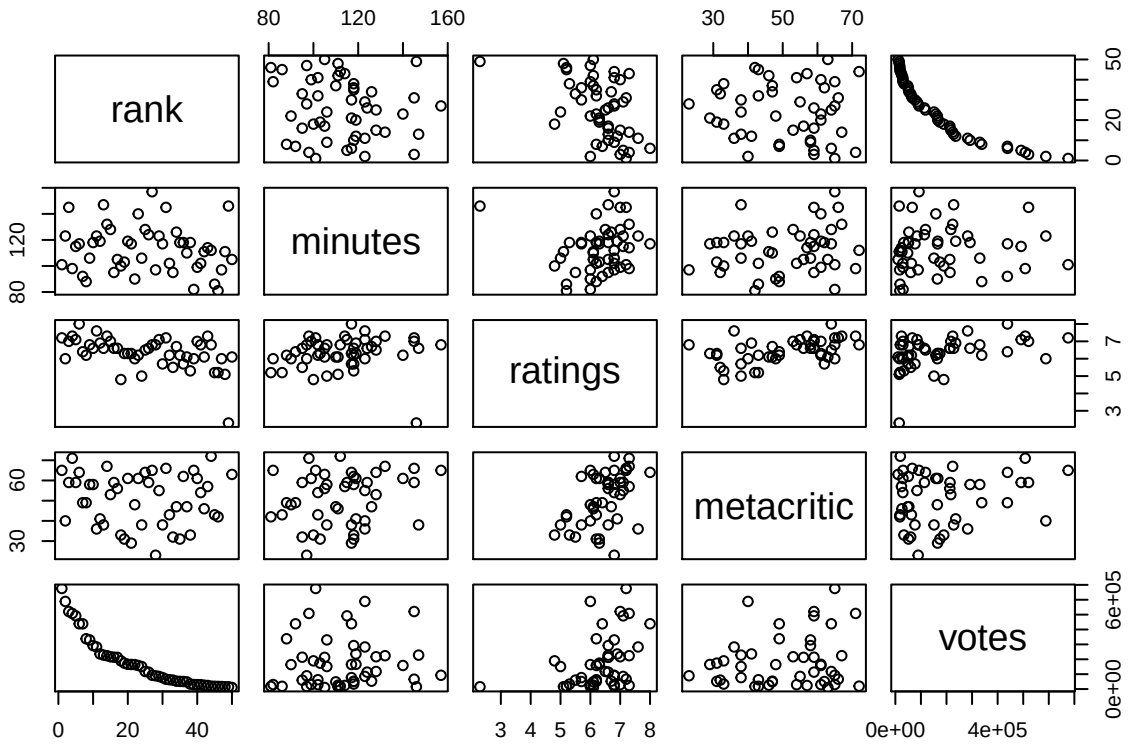


Figure 2: Will Smith Pairs Scatterplot: IMDB(2020)

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

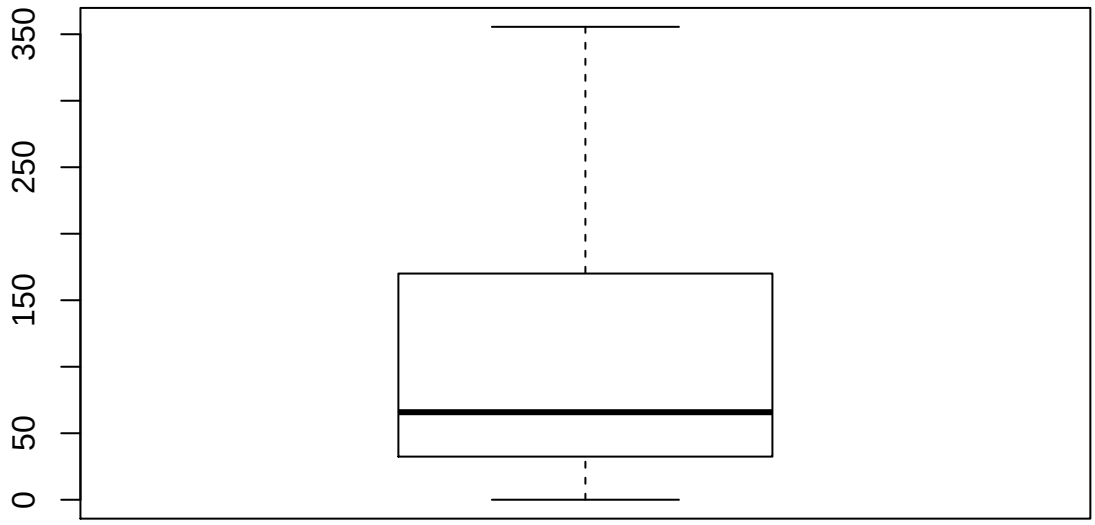


Figure 3: Will Smith Boxplot raw millions: IMDB(2020)

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

will$movies.50[widx,];
```

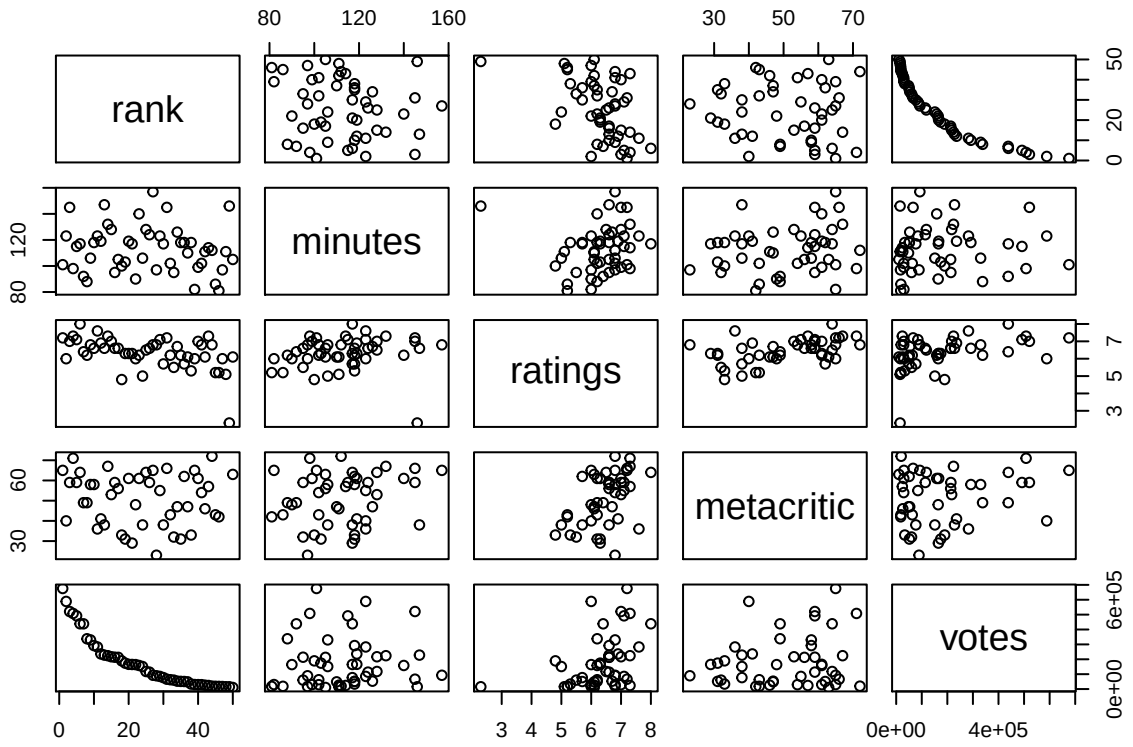
```
##    rank   title      ttid year rated minutes                       genre ratings
## 15   15 Aladdin tt6139732 2019    PG     128 Adventure, Family, Fantasy       7
##    metacritic  votes millions
## 15         53 216928   355.56
```

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

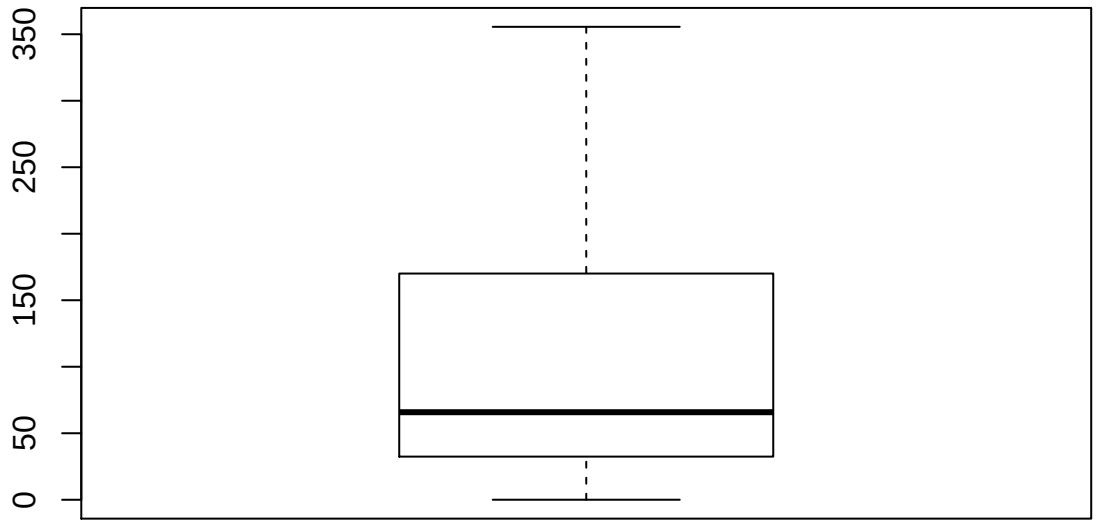
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1993    2001    2006    2007    2014    2020
```

6.0.2 Denzel Washington

```
denzel = grabFilmsForPerson(nmid);
    plot(denzel$movies.50[,c(1,6,8:10)]);
```




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




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

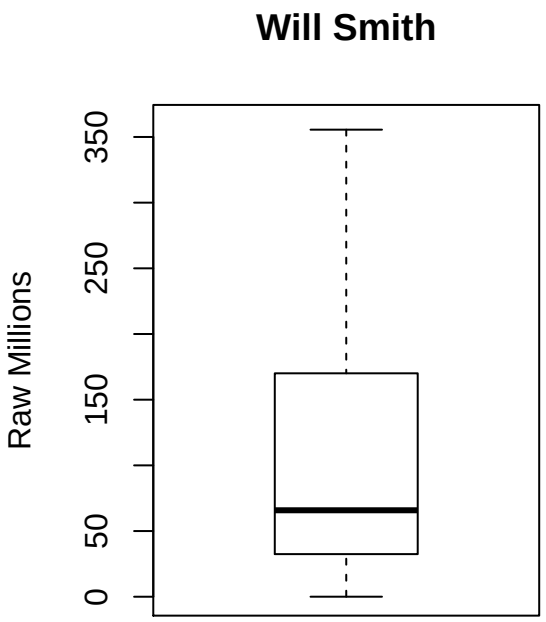
```
##    rank   title      ttid year rated minutes                       genre ratings
## 15   15 Aladdin tt6139732 2019    PG     128 Adventure, Family, Fantasy       7
##    metacritic  votes millions
## 15         53 216928   355.56
```

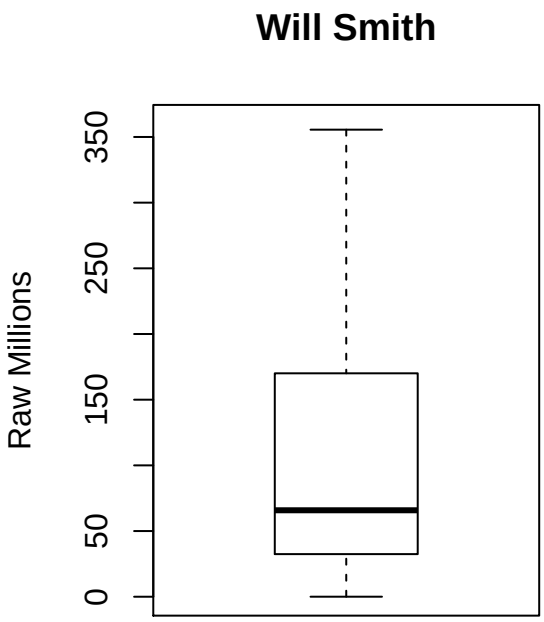
```
        summary(denzel$movies.50$year);
```

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

6.0.3 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" );
```






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

# https://www.in2013dollars.com/us/inflation/2000?endYear=1982&amount=100
# create variable £millions.2000 to convert all money to 2000 dollars ... based on year
```

7 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.

7.0.1 Adjusted Dollars (2000)

7.0.2 Totoal Votes (Divide by 1,000,000)

7.0.3 Average Ratings (Scale from 1-10)

7.0.4 Year & Minutes

7.0.5 Metacritic (NA Values)