Week 03 Assignment/ Week 04 Resubmission

Jessica Smith

```
library(devtools);
my.source = 'local';
local.path = "C:\\Users\\jsmit\\Desktop\\WSU\\DataAnalytics\\STAT419\\WSU_STATS419_FALL2020";
local.data.path = "R:/WSU_STATS419_FALL2020/";
#setwd(local.path)
knitr::opts_knit$set(root.dir = local.path)
source( paste0(local.path,"\\functions\\libraries.R"), local=T );

#install_github("MonteShaffer/humanVerseWSU/humanVerseWSU");
library(humanVerseWSU); # if your functions have the same name as the humanVerseWSU functions, there ma
```

Matrix

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

```
source( paste0(local.path,"/functions/functions-matrix.R"), local=T );
#install_github("/jsmith0434/WSU_STATS419_FALL2020/functions")

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

# dput(myMatrix); # useful
```

```
humanVerseWSU::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);
```

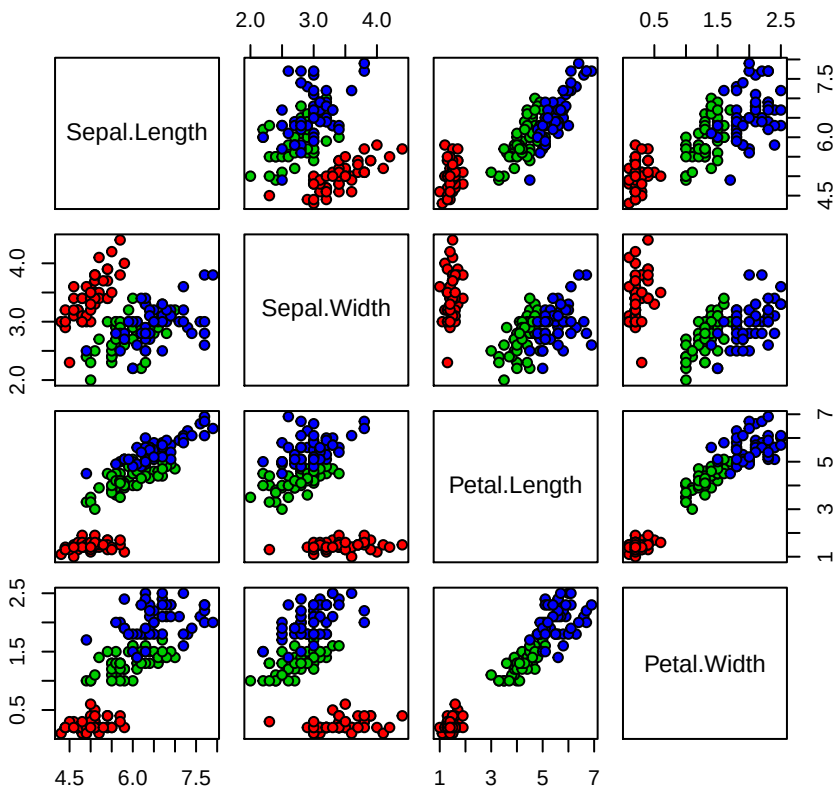
```
##      [,1] [,2] [,3]
## [1,]    2    0    5
## [2,]    0    3    0
## [3,]    1    0    4
```

```
# rotateMatrix(mat,a) ### one function using a switch statement ...
```

IRIS

Recreate the graphic for the IRIS Data Set using R. Same titles, same scales, same colors.

```
data(iris)
pairs(iris[1:4], main = "Iris Data (red=setosa, green=versicolor,blue-virginica)",
      pch = 21, bg = c("red", "green3", "blue")[unclass(iris$Species)])
```



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.]

The well known iris data set contains 50 measurements, each from 3 species of iris flower. The metrics include petal length, petal width, sepal length, and sepal width. The dataset is commonly used to teach clustering analysis and to demonstrate basic programmatic functionality. (Kaggle, 2020)

Personality

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.

```
# working directory is good-to-go

myFile = paste0(local.path,"/datasets/personality/personality-raw.txt");

#read in the data
personality = read.table(myFile, header = TRUE, sep = "|", dec = ".")

#remove unwanted column
personality = subset(personality, select = -c(V00))

#create two columns from date_test, one year and one week
temp = strsplit(as.character(personality$date_test), " ")
personality$date = matrix(unlist(temp), ncol=2, byrow=TRUE)[,1]
personality$year = format(as.Date(personality$date, "%m/%d/%Y"), format="%Y")
personality$week = format(as.Date(personality$date, "%m/%d/%Y"), format= "%W")
personality = subset(personality, select = -c(date, date_test))
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.3
```

```
personality = personality  %>% select(md5_email, year, week, everything())

#Sort the new data frame by YEAR, WEEK so the newest tests are at the top of the df
personality = personality[order(-(as.numeric(personality$year)), -(as.numeric(personality$week))), ]

#remove duplicates using the unique function based on the column "md5_email"
unique = unique(personality$md5_email)
rows = match(unique, personality$md5_email)

unique_personalities = personality[rows,]

#Save the data frame in the "pipe-delimited format" as "personality-clean.txt"
write.table(unique_personalities,"personality-clean.txt",sep="|", row.names=FALSE)

cat("The raw dataset contains ",nrow(personality),
" records, and the cleaned dataset has ", nrow(unique_personalities),".", sep = "")
```

```
## The raw dataset contains 838 records, and the cleaned dataset has 678.
```

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.

```
x.norm = rnorm(100,0,1);
s.norm = doStatsSummary ( x.norm );
str(s.norm);  # mode is pretty meaningless on this data
```

```
## List of 32
##  $ length           : int 100
##  $ length.na        : int 0
##  $ length.good      : int 100
##  $ mean             : num 0.0129
##  $ mean.trim.05     : num 0.013
##  $ mean.trim.20     : num 0.0247
##  $ median           : num 0.0287
##  $ MAD              : num 0.913
##  $ IQR              : num 1.21
##  $ quartiles        : Named num [1:3] -0.5829 0.0287 0.6305
##   ..- attr(*, "names")= chr [1:3] "25%" "50%" "75%"
##  $ deciles          : Named num [1:9] -1.0545 -0.6607 -0.4095 -0.2694 0.0287 ...
##   ..- attr(*, "names")= chr [1:9] "10%" "20%" "30%" "40%" ...
##  $ centiles         : Named num [1:99] -2.38 -1.83 -1.77 -1.54 -1.47 ...
##   ..- attr(*, "names")= chr [1:99] "1%" "2%" "3%" "4%" ...
##  $ median.weighted  : num 0.913
##  $ MAD.weighted     : num 0.0287
##  $ max              : num 2.93
##  $ min              : num -3.16
##  $ range            : num 6.09
##  $ xlim             : num [1:2] -3.16 2.93
##  $ max.idx          : num 93
##  $ min.idx          : num 37
##  $ freq.max         : num [1:100] -3.16 -2.38 -1.82 -1.77 -1.53 ...
##  $ mode             : num [1:100] -3.16 -2.38 -1.82 -1.77 -1.53 ...
##  $ which.min.freq   : num [1:100] -3.16 -2.38 -1.82 -1.77 -1.53 ...
##  $ ylim             : int [1:2] 1 1
##  $ sd               : num 0.969
##  $ var              : num 0.938
##  $ var.naive        :List of 3
##   ..$ x.bar: num 0.0129
##   ..$ s.var: num 0.938
##   ..$ s.sd : num 0.969
##  $ var.2step        :List of 3
##   ..$ x.bar: num 0.0129
##   ..$ s.var: num 0.938
##   ..$ s.sd : num 0.969
##  $ shapiro          :List of 4
##   ..$ statistic: Named num 0.986
##   .. ..- attr(*, "names")= chr "W"
##   ..$ p.value  : num 0.362
##   ..$ method   : chr "Shapiro-Wilk normality test"
##   ..$ data.name: chr "xx"
##   ..- attr(*, "class")= chr "htest"
##  $ shapiro.is.normal:List of 3
##   ..$ 0.10: logi TRUE
##   ..$ 0.05: logi TRUE
##   ..$ 0.01: logi TRUE
##  $ outliers.z       :'data.frame':   2 obs. of  2 variables:
##   ..$ value    : Factor w/ 2 levels "-3.15517166462159",..: 1 2
##   ..$ direction: Factor w/ 2 levels "lower","upper": 1 2
##  $ outliers.IQR     :'data.frame':   3 obs. of  3 variables:
##   ..$ value    : Factor w/ 3 levels "-3.15517166462159",..: 1 3 2
```

```
##   ..$ fence    : Factor w/ 1 level "inner": 1 1 1
##   ..$ direction: Factor w/ 2 levels "lower","upper": 1 2 2
```

```
x.unif = runif(100,0,1);
s.unif = doStatsSummary ( x.unif );
str(s.unif);  # mode is pretty meaningless on this data
```

```
## List of 32
##  $ length           : int 100
##  $ length.na        : int 0
##  $ length.good      : int 100
##  $ mean             : num 0.476
##  $ mean.trim.05     : num 0.471
##  $ mean.trim.20     : num 0.448
##  $ median           : num 0.411
##  $ MAD              : num 0.346
##  $ IQR              : num 0.473
##  $ quartiles        : Named num [1:3] 0.241 0.411 0.714
##   ..- attr(*, "names")= chr [1:3] "25%" "50%" "75%"
##  $ deciles          : Named num [1:9] 0.138 0.201 0.272 0.34 0.411 ...
##   ..- attr(*, "names")= chr [1:9] "10%" "20%" "30%" "40%" ...
##  $ centiles         : Named num [1:99] 0.0477 0.0675 0.0752 0.0771 0.0804 ...
##   ..- attr(*, "names")= chr [1:99] "1%" "2%" "3%" "4%" ...
##  $ median.weighted  : num 0.346
##  $ MAD.weighted     : num 0.411
##  $ max              : num 0.992
##  $ min              : num 0.0449
##  $ range            : num 0.947
##  $ xlim             : num [1:2] 0.0449 0.992
##  $ max.idx          : num 2
##  $ min.idx          : num 92
##  $ freq.max         : num [1:100] 0.0449 0.0477 0.0679 0.0755 0.0772 ...
##  $ mode             : num [1:100] 0.0449 0.0477 0.0679 0.0755 0.0772 ...
##  $ which.min.freq   : num [1:100] 0.0449 0.0477 0.0679 0.0755 0.0772 ...
##  $ ylim             : int [1:2] 1 1
##  $ sd               : num 0.288
##  $ var              : num 0.0829
##  $ var.naive        :List of 3
##   ..$ x.bar: num 0.476
##   ..$ s.var: num 0.0829
##   ..$ s.sd : num 0.288
##  $ var.2step        :List of 3
##   ..$ x.bar: num 0.476
##   ..$ s.var: num 0.0829
##   ..$ s.sd : num 0.288
##  $ shapiro          :List of 4
##   ..$ statistic: Named num 0.932
##   .. ..- attr(*, "names")= chr "W"
##   ..$ p.value  : num 6.23e-05
##   ..$ method   : chr "Shapiro-Wilk normality test"
##   ..$ data.name: chr "xx"
##   ..- attr(*, "class")= chr "htest"
##  $ shapiro.is.normal:List of 3
##   ..$ 0.10: logi FALSE
```

```
##   ..$ 0.05: logi FALSE
##   ..$ 0.01: logi FALSE
##  $ outliers.z       :'data.frame':   0 obs. of  2 variables:
##   ..$ value    : Factor w/ 0 levels:
##   ..$ direction: Factor w/ 0 levels:
##  $ outliers.IQR     :'data.frame':   0 obs. of  3 variables:
##   ..$ value    : Factor w/ 0 levels:
##   ..$ fence    : Factor w/ 0 levels:
##   ..$ direction: Factor w/ 0 levels:
```

```
v2.norm = doSampleVariance(x.norm, "two-pass");
v2b.norm = doSampleVariance(x.norm);  # default value is "two-pass" in the function
v2c.norm = doSampleVariance(x.norm, "garblideljd=-gook"); # if logic defaults to "two-pass"

unlist(v2.norm);
```

```
##          sumXs sumDiffSquared       variance
##      1.2912690     92.8883443      0.9382661
```

```
unlist(v2b.norm);
```

```
##          sumXs sumDiffSquared       variance
##      1.2912690     92.8883443      0.9382661
```

```
unlist(v2c.norm);
```

```
##          sumXs sumDiffSquared       variance
##      1.2912690     92.8883443      0.9382661
```

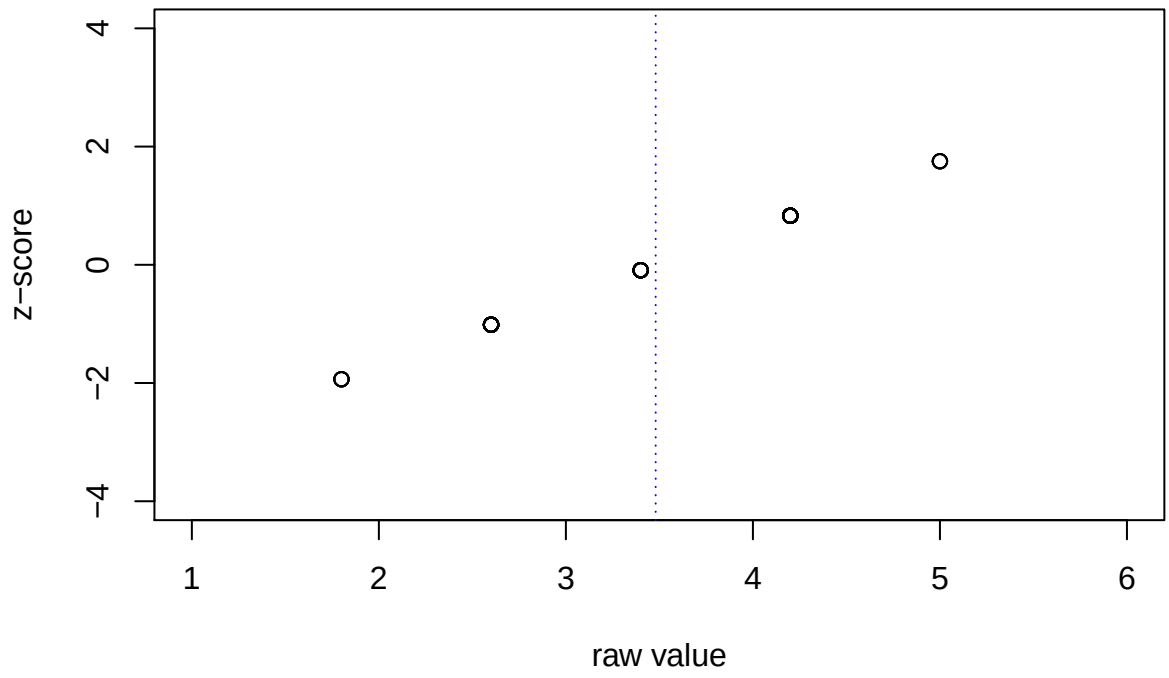
Z-Scores

Application of z-score

```
#library(digest);
#md5_monte = digest("monte.shaffer@gmail.com", algo="md5");  # no workee???
md5_monte = "b62c73cdaf59e0a13de495b84030734e";
#get the Monte row from the clean dataset
monte = unique_personalities[unique_personalities$md5_email == md5_monte, ]
monte = monte[c(-1, -2, -3)]

data = zScores(monte)
plot(unlist(data["value", ]), unlist(data["z-score", ]), ylab = "z-score",
     xlab = "raw value", main = "monte.shaffer@gmail.com", xlim = (c(1,6)),
     ylim = (c(-4, 4)))
abline(v = rowMeans(data["value", ]), lty = 3, col = "blue")
```

monte.shaffer@gmail.com




```
writeLines("The zscore is a measure of how far from the mean a data point is, to phrase it
informally.
           \n The plot of the monte sample shows that as the raw value gets higher, the z-score
           gets lower. The \n mean of the sample is show by the blue dashed line at 3.84, so
           the z-scores closest to zero should \n be associated with raw values near the mean.")
```

```
## The zscore is a measure of how far from the mean a data point is, to phrase it
## informally.
##
##  The plot of the monte sample shows that as the raw value gets higher, the z-score
##            gets lower. The
##  mean of the sample is show by the blue dashed line at 3.84, so
##            the z-scores closest to zero should
##  be associated with raw values near the mean.
```

Will vs Denzel

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

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

nmid = "nm0000243";
```

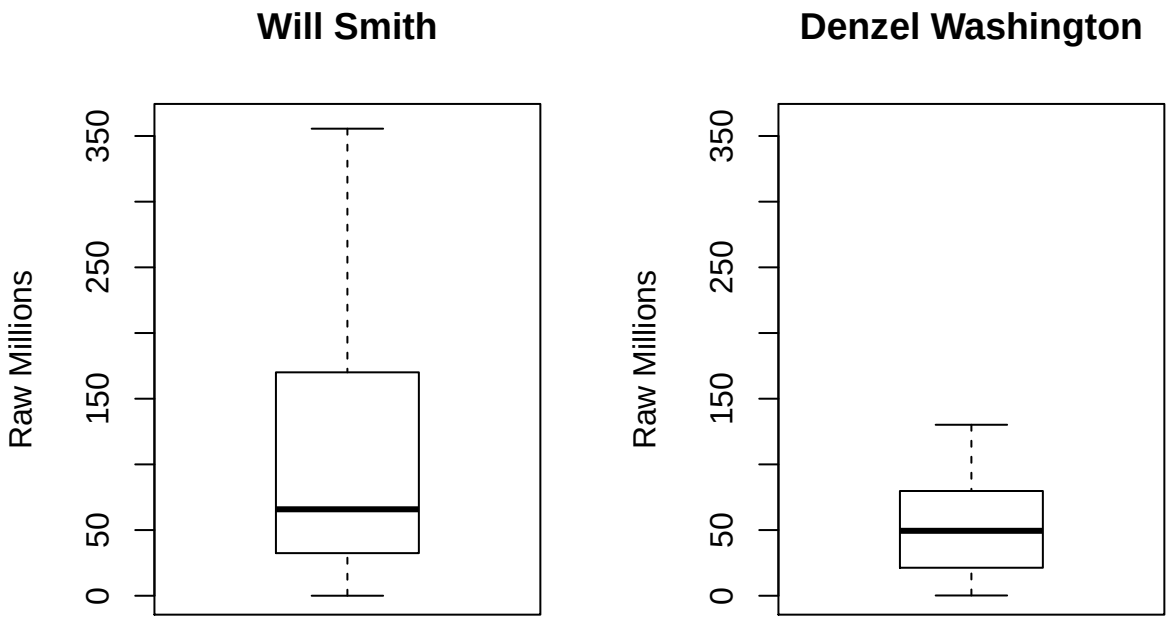
```
denzel = grabFilmsForPerson(nmid);

#
```

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. 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.

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));
```

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.

Adjusted Dollars (2000)

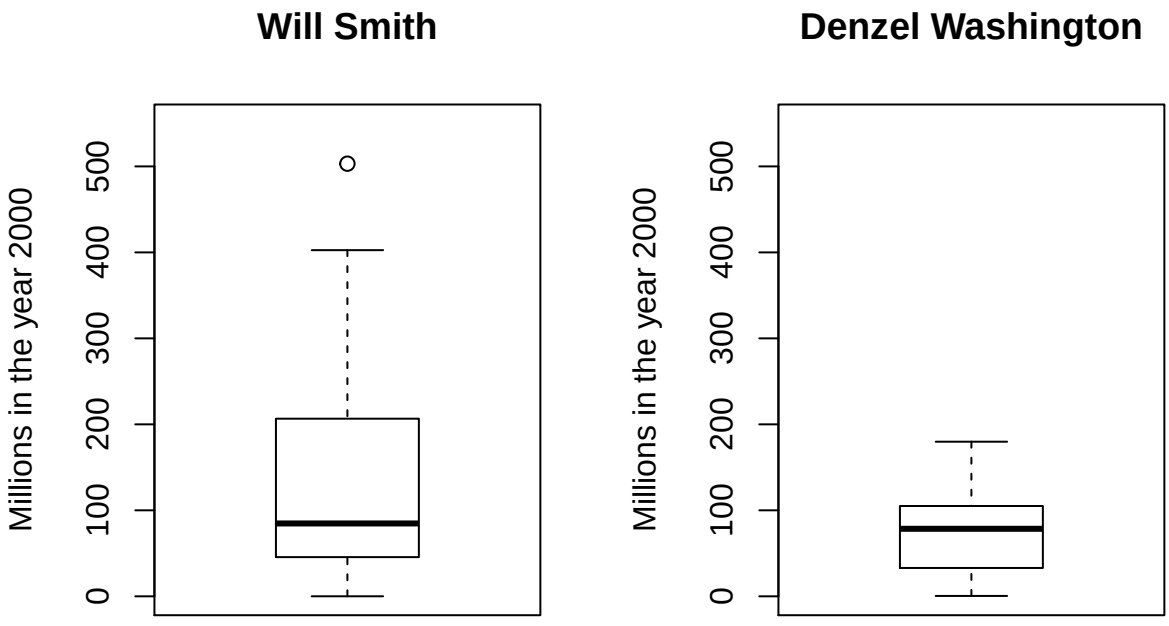
```
#inflation: https://inflationdata.com/inflation/Inflation_Articles/CalculateInflation.asp
#CPI: https://inflationdata.com/Inflation/Consumer_Price_Index/HistoricalCPI.aspx?reloaded=true#Table
cpi = read.table("C:\\Users\\jsmit\\Desktop\\WSU\\DataAnalytics\\STAT419\\WSU_STATS419_FALL2020\\datase

mean = rowMeans(cpi[ , -1], na.rm= TRUE)
CPI = as.data.frame(cbind(year = cpi$AR ,cpi = mean))
cpi2000 = CPI[1,2]

library(dplyr)
D = denzel$movies.50 %>% inner_join(CPI, by = "year")
W = will$movies.50 %>% inner_join(CPI, by = "year")

D$millions_2000 = (D$millions*cpi2000)/D$cpi
W$millions_2000 = (W$millions*cpi2000)/W$cpi
```

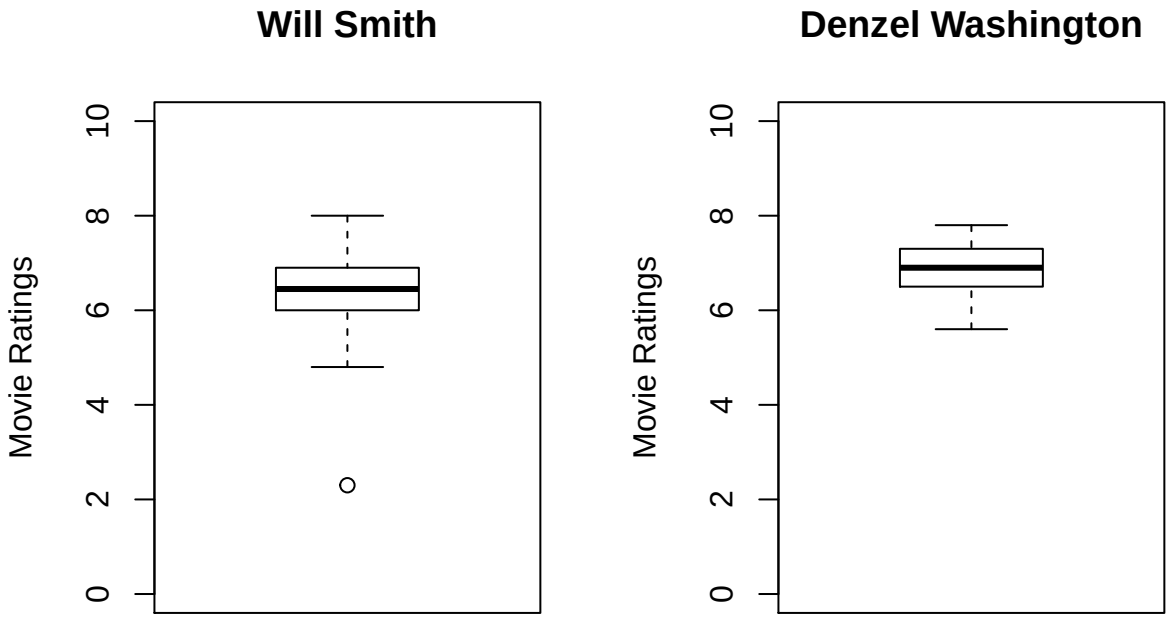
```
par(mfrow=c(1,2));
boxplot(W$millions_2000, main=will$name, ylim=c(0,550), ylab="Millions in the year 2000")
boxplot(D$millions_2000, main=denzel$name, ylim=c(0,550), ylab="Millions in the year 2000")
```




```
writeLines("The boxplots of box office sales show that the median for the two stars is about
the same.\n There is greater variation in the amount that Will Smith movies earn as shown by
the longer \n whiskers and larger interquartile range, while Denzel's movies appear to be
more consistent.")
```

```
## The boxplots of box office sales show that the median for the two stars is about
## the same.
##  There is greater variation in the amount that Will Smith movies earn as shown by
## the longer
##  whiskers and larger interquartile range, while Denzel's movies appear to be
## more consistent.
```

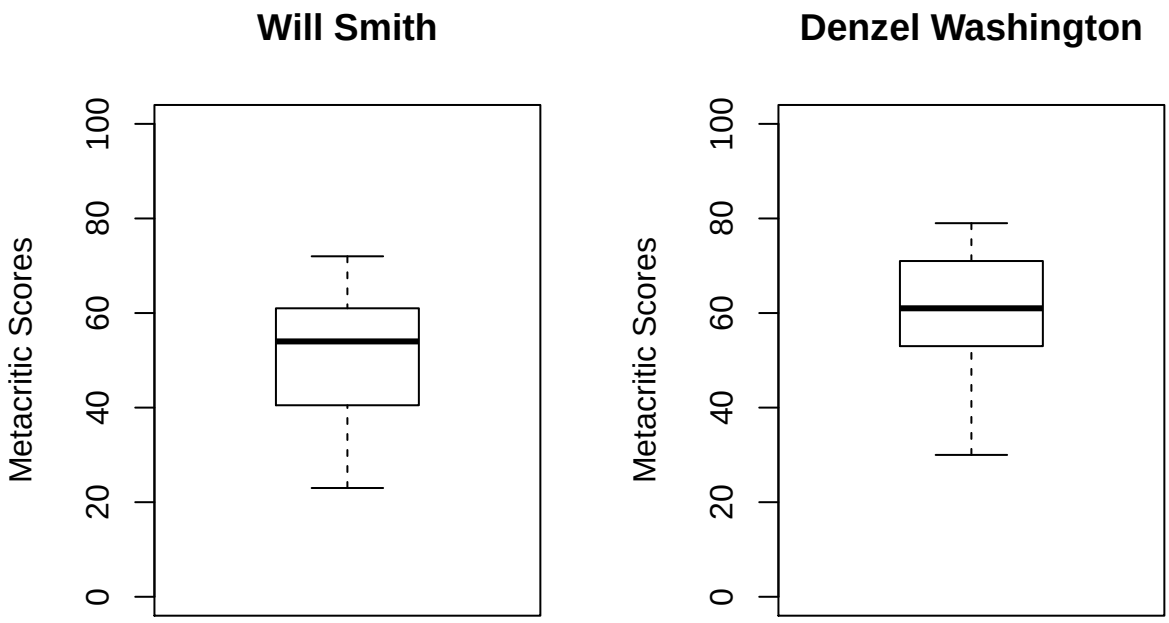
```
par(mfrow=c(1,2));
boxplot(W$ratings, main=will$name, ylim=c(0,10), ylab="Movie Ratings")
boxplot(D$ratings, main=denzel$name, ylim=c(0,10), ylab="Movie Ratings")
```




```
writeLines("The boxplots of movie ratings show that the median for the two stars is again
very close,\n but Denzel's score is a little higher.  There is a larger range of ratings
for Will Smith's movies,\n shown by the longer whiskers. The interquartile ranges have
similar sizes for both actors")
```

```
## The boxplots of movie ratings show that the median for the two stars is again
## very close,
##  but Denzel's score is a little higher.  There is a larger range of ratings
## for Will Smith's movies,
##  shown by the longer whiskers. The interquartile ranges have
## similar sizes for both actors
```

```
par(mfrow=c(1,2));
boxplot(W$metacritic, main=will$name, ylim=c(0,100), ylab="Metacritic Scores")
boxplot(D$metacritic, main=denzel$name, ylim=c(0,100), ylab="Metacritic Scores")
```




```
writeLines("The metacritic scores for the two actors show the mean score for Denzel to
be slightly higher.\n The whisker legth is similar, as is the size of the interquartile
range.")
```

```
## The metacritic scores for the two actors show the mean score for Denzel to
## be slightly higher.
##  The whisker legth is similar, as is the size of the interquartile
## range.
```