STATS 419 Survey of Multivariate Analysis

03 datasets revisted

Shawn Petersen
(shawn.petersen@wsu.edu)

Instructor: Monte J. Shaffer

21 September 2020

1 Introduction

For homework week #2 we are applying our "R" knowledge for matrix math, data wrangling personality data, creating functions for basic data analysis, looking at IMDB data from Will Smith & Denzel Washington and box plots the results of our movie analysis.

2 Question 1

2.1 Rotation Matrix

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

Answer: I created one function and the user is able to pass in the original matrix and a rotational value in degrees. I support 90, 180, and 270 degrees rotations with this function

```
# Function to rotate a matrix based on user passed parameter on degrees
transposeMatrix <- function(myMatrix,rotate_degrees)
{
   transformation_mat <- matrix (c ( 0,  0,  1,
                                     0,  1,  0,
                                     1,  0,  0), nrow=3, byrow=T);

  if (rotate_degrees == 90){
    matrixrotate <- t(myMatrix %*% transformation_mat)
  }else if (rotate_degrees == 180){
    temprotate <- t(myMatrix %*% transformation_mat)
    matrixrotate <- t(temprotate %*% transformation_mat)
  } else if (rotate_degrees == 270){
    temprotate <- t(myMatrix %*% transformation_mat)
    temprotate <- t(temprotate %*% transformation_mat)
    matrixrotate <- t(temprotate %*% transformation_mat)
  }else {
    print ("Invalid rotation passed. Returning original, unchanged matrix")
    matrixrotate <- myMatrix
  }

  return(matrixrotate)
}
```

2.2 Origina Matrix

```
original_matrix <-  matrix (c ( 1, 0, 2,
                                0, 3, 0,
                                4, 0, 5), nrow=3, byrow=T);



original_matrix
```

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

2.3 Rotate Matrix 90 Degrees

```
rotateMatrix90 <- transposeMatrix(original_matrix,90)
rotateMatrix90
```

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

2.4 Rotate Matrix 180 Degrees

```
rotateMatrix180 <- transposeMatrix(original_matrix,180)
rotateMatrix180
```

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

2.5 Rotate Matrix 270 Degrees

```
rotateMatrix270 <- transposeMatrix(original_matrix,270)
rotateMatrix270
```

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

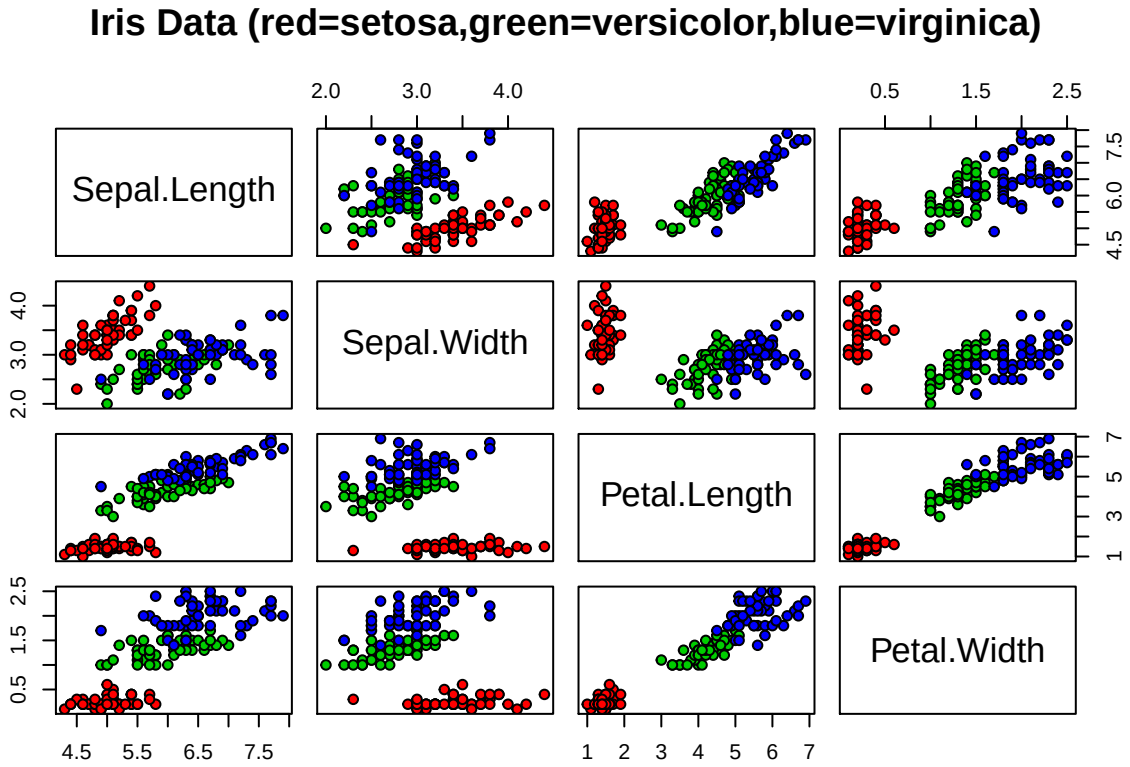
3 Question 2

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

Answer: I imported the built-in IRIS data set and grouped IRIS species and plotted the correlation between the variables

```
## [1] "data.frame"
```

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



4 Question 3

Right 2-3 sentences concisely defining the IRIS Data Set

Answer: The IRIS data set, sometimes called the Fishers or Anderson data set. Fisher was a British statistician and biologist and published the results of the IRIS flower report in 1936. The IRIS dataset measured the flowers sepal length, sepal width, petal length, and pedal length over a set of 50 flowers and 3 types of species (setosa, versicolor, and virginica). The data set is the rock child for beginners learning data science, hence it is built into R's core language engine.

5 Question 4

Import "personality-raw.txt" into R. Remove the V00 column. Create two new columns from the current column "date_test"

Below is the R code I developed to import the personality data and wrangle the data's date_test column into 2 additional columns, year and week. These columns were sorted and used to find unique md5 email strings. I then extracted the md5 email string for "monte.shaffer@gmail.com" and did a summary of the data set using custom functions seen below.

```
#source("MyFunctions.R")

personality_df <- read.table("personality-raw.txt", header=TRUE, sep =  "|")

#copy data frame
modified_df <- personality_df

#remove column £V00
modified_df$V00 <- NULL

#split the data_test column into year and week
year_time_split <- str_split_fixed(modified_df$date_test, "\\s+", 2)
year_string <-  as.Date(year_time_split[,1],'%m/%d/%Y')
year_numeric <- as.numeric(format(year_string,'%Y'))

modified_df["year"] <- year_numeric
modified_df["week"] <- as.numeric(strftime((year_string), format = "%V"))


# sort data frame by year then week
sorted_df <- arrange(modified_df,desc(modified_df$year), desc(modified_df$week))

#remove unique email address
unique_vec <- unique(modified_df$md5_email)
length(unique_vec)

unique_df <- sorted_df %>% distinct(md5_email, .keep_all = TRUE)

#write unique email dataframe to text file, pipe delimited
write.table(unique_df, file = "personality-clean.txt", sep = "|")
```

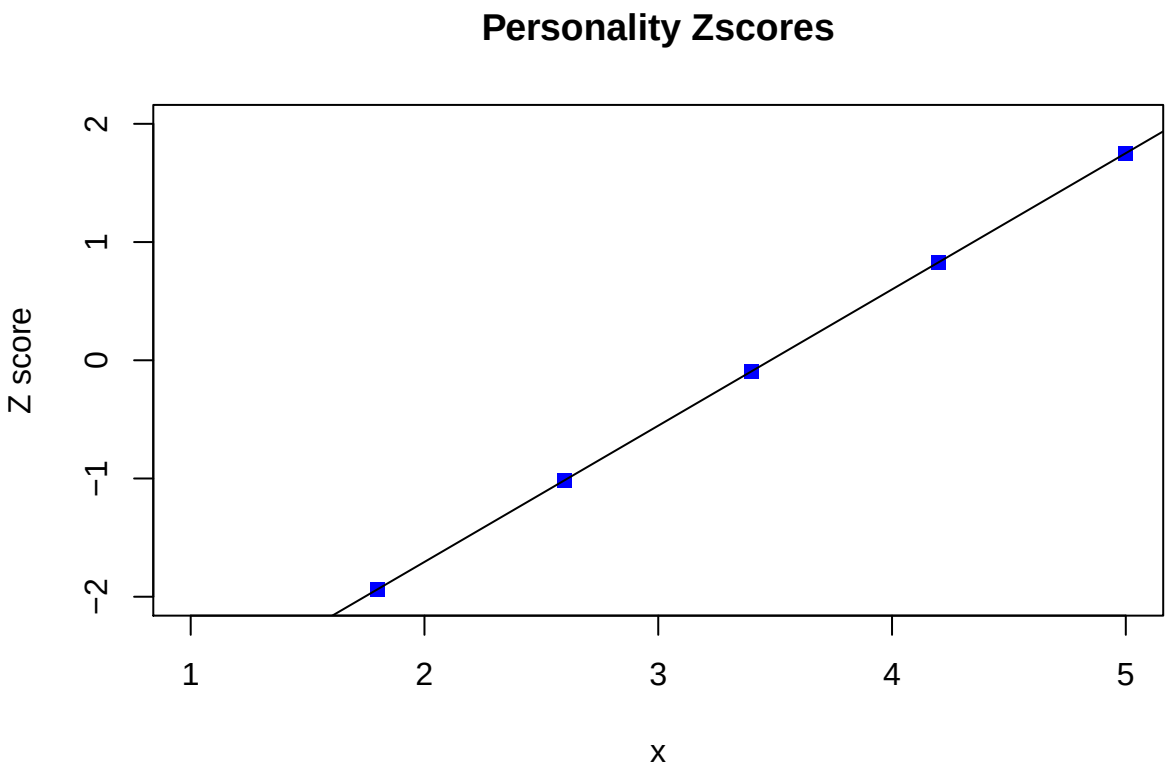
```
## [1] 678
```

```
#filter for Monte Shaffer email
monte_df <- filter(unique_df, unique_df$md5_email == "b62c73cdaf59e0a13de495b84030734e")

#remove all columns not starting with "V"
monte_df_clean <- monte_df %>% dplyr:: select(starts_with("V"))

#Transpose the vector from row to column format
monte_df_clean_t <- t(monte_df_clean)

#Summarize the data set
doSummary(monte_df_clean_t)
```




```
## Record lengeth= 60
## Number of NA's= 0
## Mean= 3.48
## Median= 3.4
## Mode results= 4.2
## Naive sum =  208.8
## Naive sum squard =   771.04
## Naive variance = 0.7528136
## 2-pass sum = 208.8
## 2-pass sum squard =  771.04
## 2-pass variance =    0.7528136
## Built-in Standard deviation function= 0.8676483
```

6 Question 5

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.

I created R functions, "doSummary", "sampleVariance" and "doMode"

Dataset findings: * original dataset had 838 records * clean dataset had 678 records * 2-pass and Naive algorithms produced the same results for variance

Zscores plot: * The Z scores plot. A Z score, also called the "Standard Score", measures how many standard deviations below or above the population mean a raw score is. A Z score of 0 ("zero") is exactly average. A Z score of 1 is 1 standard deviation above the mean. A Z score of -2 is -2 standard deviations below the mean.

The average for "monte.shaffer@gmail.com" was 3.4 and the Z score plot shows a Z score of 0 at 3.4.

7 Question 6

Compare Will Smith and Denzel Washington. 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.

I created a custom function seen below to convert the actors revenue and adjust for inflation. The inflation year is passed in by the user so its flexible by any given year from 1920-2020

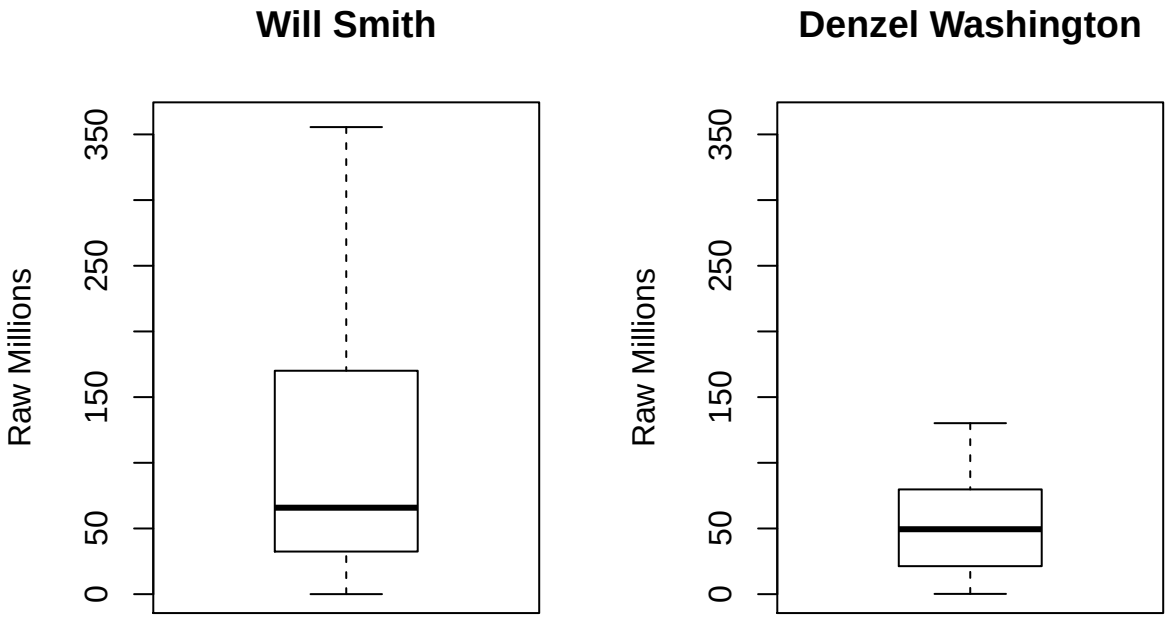
```
nmid = "nm0000226";
will = grabFilmsForPerson(nmid);
a <- will$movies.50

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


#Denzel Washington
nmid = "nm0000243";
  denzel = grabFilmsForPerson(nmid);

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

  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" )
```



```
#read inflation data
inflation_df <- read.table("inflation.txt", header=TRUE, sep =  "|")

#calculate inflation adjustment for Will Smith movie revenue
will$millions.2000 <- convert_inflation(will$movies.50,2000, inflation_df)
#will£millions.2000

#calculate inflation adjustment for Denzel Washington movie revenue
denzel$millions.2000 <- convert_inflation(denzel$movies.50,2000, inflation_df)
#denzel£millions.2000

#Plot new box plots with inflation adjusted revenues
par(mfrow=c(1,2))
boxplot(will$millions.2000, main=will$name, ylim=c(0,600), ylab="Inflation adjusted Millions" )
boxplot(denzel$millions.2000, main=denzel$name, ylim=c(0,600), ylab="Inflation adjusted Millions" )
```

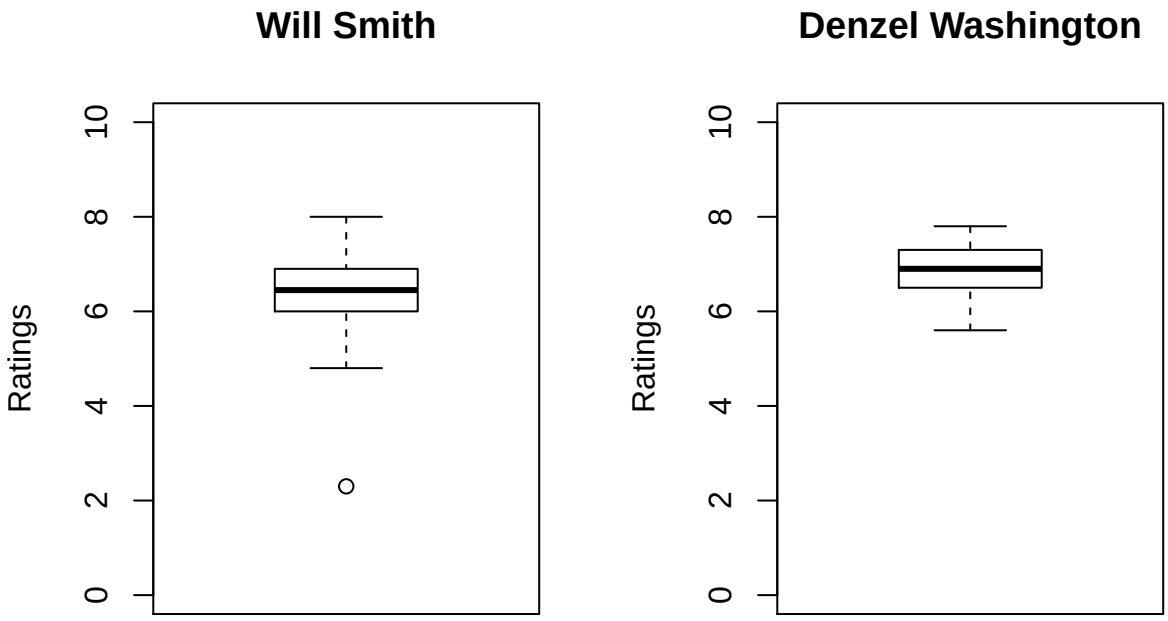

```
##    rank   title      ttid year rated minutes                       genre ratings
## 15   15 Aladdin tt6139732 2019    PG     128 Adventure, Family, Fantasy       7
##    metacritic  votes millions
## 15         53 216913   355.56
##   rank             title      ttid year rated minutes                   genre
## 1    1 American Gangster tt0765429 2007     R     157 Biography, Crime, Drama
##   ratings metacritic  votes millions
## 1     7.8         76 384283   130.16
```

8 Question 7

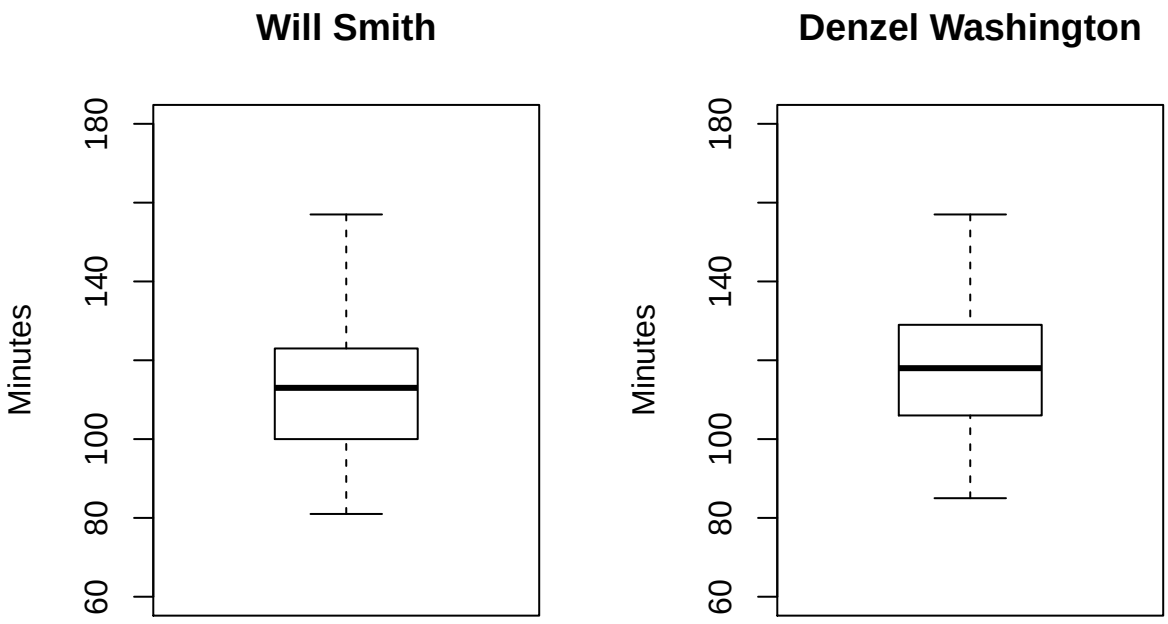
Build side-by-side box plots on several of the variables (including #6) to compare the two movie stars. After each box plot, write 2+ sentence describing what you are seeing, and what conclusions you can logically make. You will need to review what the box plot is showing with the box portion, the divider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

I created the following box plots, Will Smith vs. Denzel Washington: * Box plot for ratings The results of the ratings showed Denzel had a better rating over his career than Will S. Will S. does have one outlier rating of 2.3(Student of the Year 2). Denzel has a tight box plot, which shows his movies ranking have been consistent over his career * Box plot for movie length(minutes) The results of the movie minutes showed Denzel tends to make longer movies than Will S. Will S. box plot shows a taller whiskers which indicates he has several movies he made that were outside his normal movie length. These movies where "Ali" and "Bad Boys II"

```
#Plot movie rankings
  par(mfrow=c(1,2))
  boxplot(will$movies.50$ratings, main=will$name, ylim=c(0,10), ylab="Ratings" )
  boxplot(denzel$movies.50$ratings, main=denzel$name, ylim=c(0,10), ylab="Ratings" )
```




```
  #plot movie minutes
  par(mfrow=c(1,2))
  boxplot(will$movies.50$minutes, main=will$name, ylim=c(60,180), ylab="Minutes" )
  boxplot(denzel$movies.50$minutes, main=denzel$name, ylim=c(60,180), ylab="Minutes" )
```


Will Smith
Minutes
60
80
100
140
180
Denzel Washington
Minutes
60
80
100
140
180