STATS419 Survey of Multivariate Analysis

Week 03 Assignment 02_datasets

Minju Lee
(minju.lee@wsu.edu)
[11638071]

Instructor: Monte J. Shaffer

21 September 2020

```
library(devtools);
my.source = 'github'
github.path = "https://raw.githubusercontent.com/minju-lee92/WSU_STATS419_FALL2020/";
source_url(paste0(github.path,"master/functions/libraries.R"));
source_url(paste0(github.path,"master/functions/functions-imdb.R"));
source_url(paste0(github.path,"master/functions/functions-inflation.R"));
```

1 Matrix

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

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


transposeMatrix = function(mat)
  {
  t(mat);
  }

transposeMatrix(myMatrix)
```

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

```
#I need to revere the rows befor applying transpose

rotateMatrix90 = function(mat)
  {
  t(mat[nrow(myMatrix):1,])
  }

rotateMatrix90(myMatrix)
```

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

```
rotateMatrix180 = function(mat)
  {
  t(rotateMatrix90(mat)[nrow(rotateMatrix90(mat)):1,])
  }

rotateMatrix180(myMatrix)
```

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

```
rotateMatrix270 = function(mat)
  {
  t(rotateMatrix180(mat)[nrow(rotateMatrix180(mat)):1,])
  }

rotateMatrix270(myMatrix)
```

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

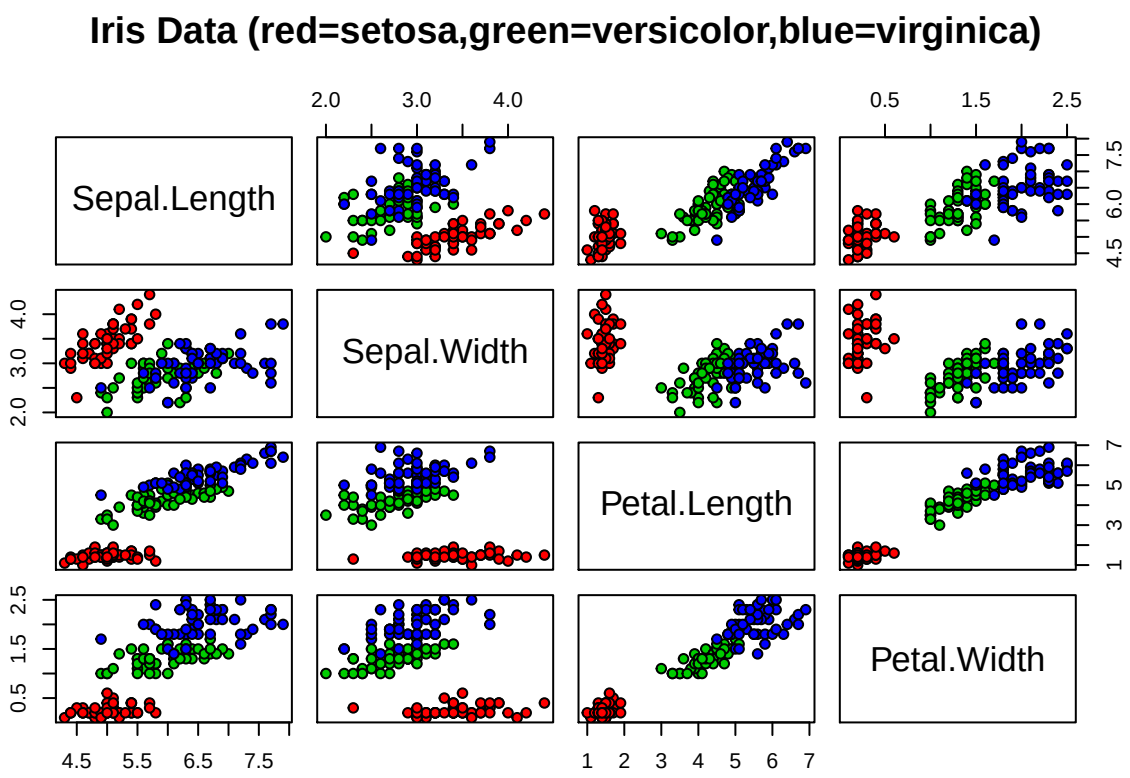
2 IRIS

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

```
#getting the data and investigate the data
data(iris)
#dim(iris)
#attributes(iris)
#summary(iris)

#create the graph
# This code can be found at https://commons.wikimedia.org/wiki/File:Iris_dataset_scatterplot.svg

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



3 Personality

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.

The raw dataset had 838 rows. The clean dataset has 678 rows.

```
# read in the raw data
pers_raw <- read.table("../datasets/personality/personality-raw.txt", sep="|", header=TRUE)
#pers_raw

# remove V00 column
pers_new <- subset(pers_raw, select = -V00)
#pers_new

# create two new columns for year and week
library(dplyr)
library(lubridate)

pers_new$date_test <- mdy_hm(pers_new$date_test)
```

```
pers_new$year <- year(pers_new$date_test)
pers_new$week <- format(pers_new$date_test, "%m/%d")

# sort the data
pers_sort <- arrange(pers_new, desc(year), desc(week))

# remove duplicates
length(unique(pers_sort$md5_email))
```

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

```
pers_clean <- pers_sort %>% distinct(pers_sort$md5_email, .keep_all=TRUE)
#pers_clean

ms <- filter(pers_clean, md5_email =='b62c73cdaf59e0a13de495b84030734e')

ms <- as.matrix(ms)
ms
```

```
##      md5_email                          date_test             V01   V02   V03
## [1,] "b62c73cdaf59e0a13de495b84030734e" "2020-04-06 12:57:00" "3.4" "4.2" "2.6"
##      V04   V05   V06   V07   V08   V09   V10   V11   V12   V13   V14   V15
## [1,] "4.2" "2.6" "2.6" "4.2" "2.6" "3.4" "4.2" "4.2" "3.4" "3.4" "4.2" "5"
##      V16   V17 V18   V19   V20   V21   V22   V23   V24   V25 V26   V27   V28
## [1,] "3.4" "5" "3.4" "1.8" "2.6" "2.6" "2.6" "4.2" "3.4" "5" "2.6" "4.2" "3.4"
##      V29   V30   V31   V32   V33   V34   V35   V36   V37   V38   V39   V40
## [1,] "2.6" "2.6" "4.2" "1.8" "3.4" "4.2" "4.2" "4.2" "2.6" "4.2" "2.6" "4.2"
##      V41   V42   V43   V44   V45   V46   V47   V48   V49   V50   V51   V52
## [1,] "4.2" "4.2" "4.2" "2.6" "4.2" "4.2" "2.6" "3.4" "2.6" "4.2" "1.8" "4.2"
##      V53   V54   V55   V56   V57   V58   V59   V60   year   week
## [1,] "2.6" "3.4" "4.2" "4.2" "1.8" "4.2" "2.6" "4.2" "2020" "04/06"
##      pers_sort$md5_email
## [1,] "b62c73cdaf59e0a13de495b84030734e"
```

```
# save cleaned data

# write.table(pers_clean, "../WEEK-03/output/personality-clean.txt", sep="|",row.names = FALSE)
```

4 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.

4.1 Variance

4.1.1 Naive

howww

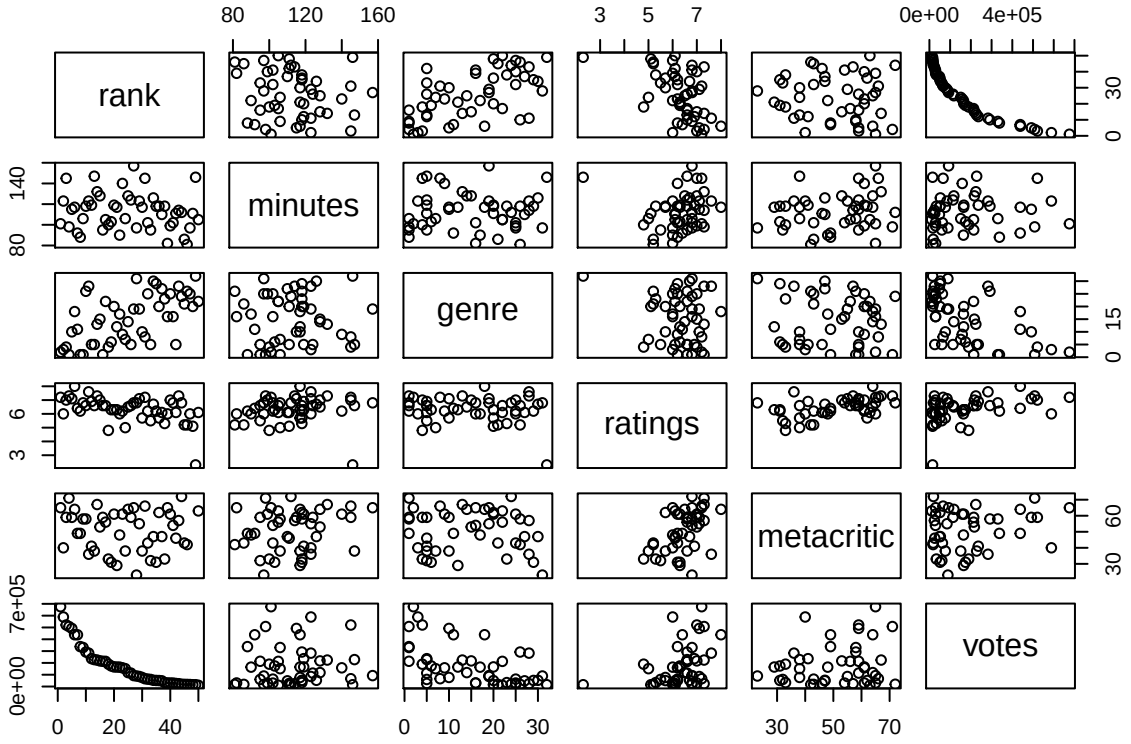


Figure 1: Will Smith scatterplot: IMDB(2020)

4.1.2 Traditional Two Pass

4.2 Z-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//imc 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.

5.1 Will Smith

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

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

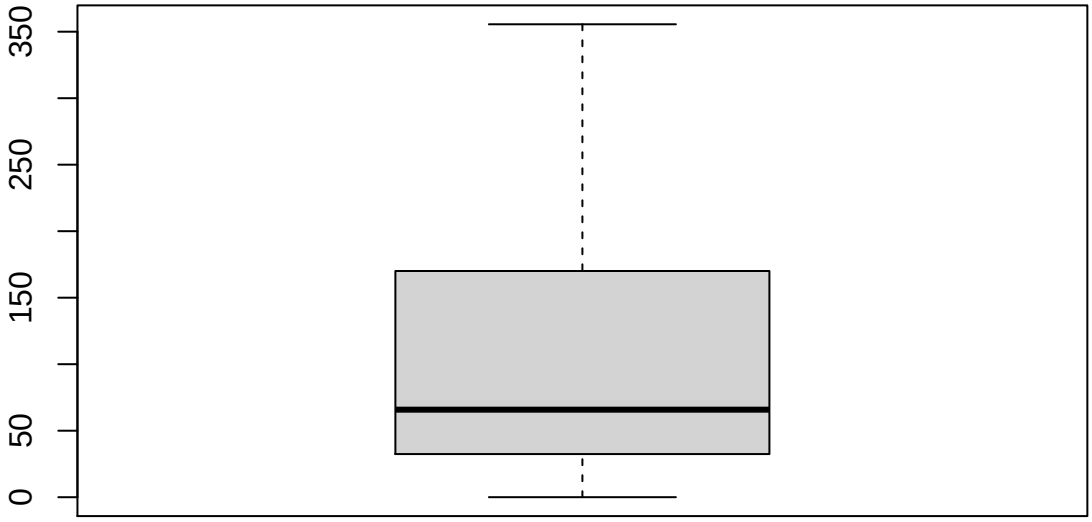


Figure 2: Will Smoth boxplot raw millions: IMDB(2020)

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

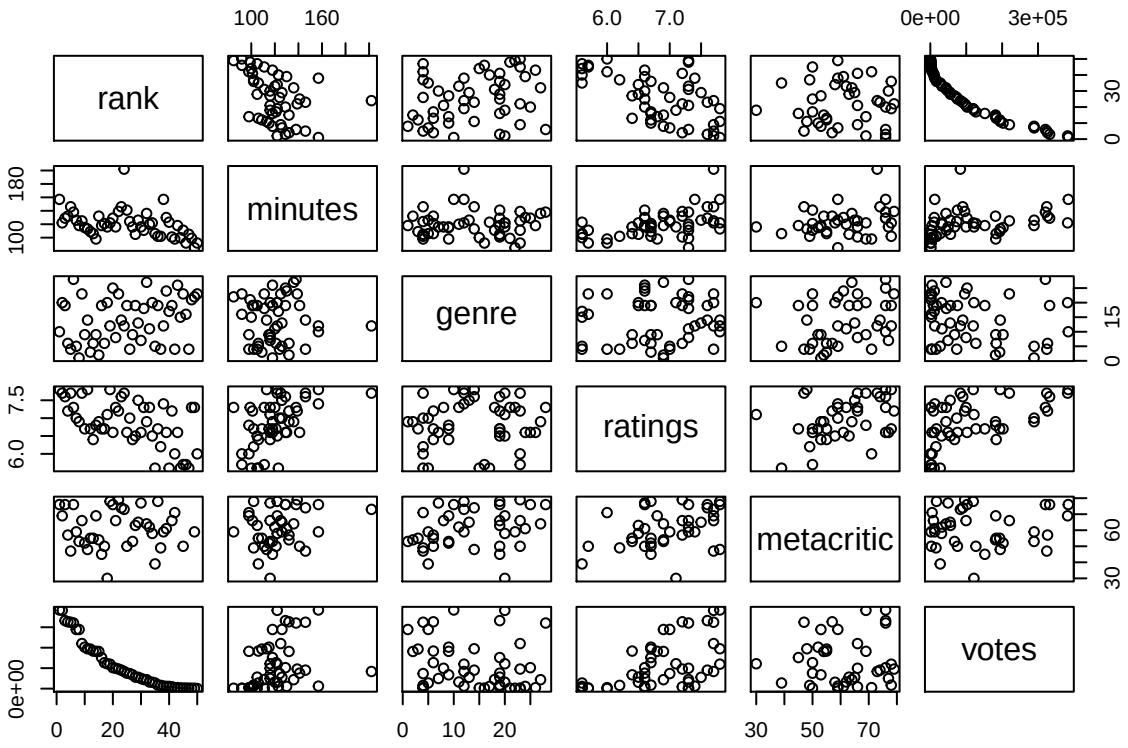
```
##    rank   title      ttid year rated minutes                       genre ratings
## 15   15 Aladdin tt6139732 2019    PG     128 Adventure, Family, Fantasy       7
##    metacritic  votes millions
## 15         53 216928   355.56
```

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

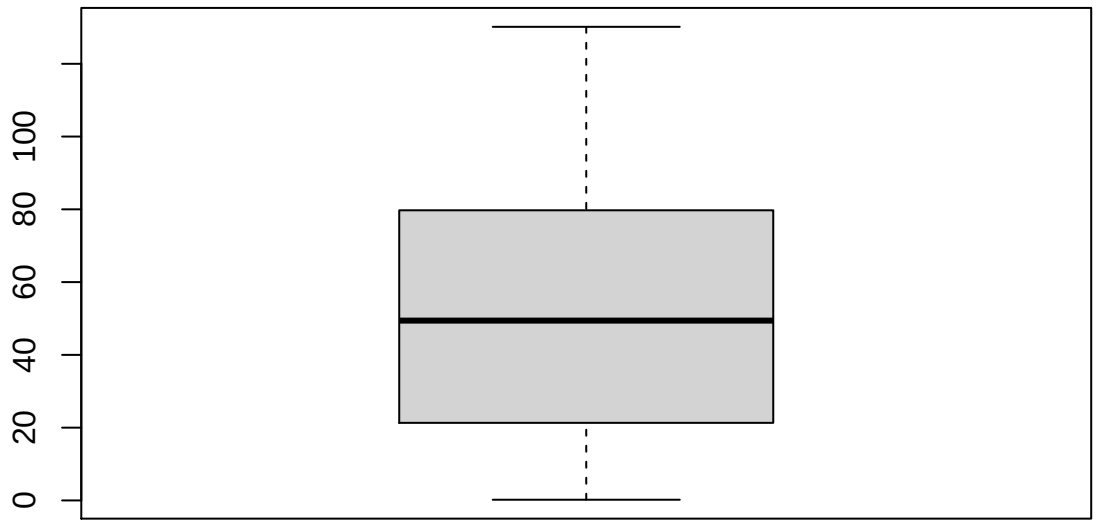
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1993    2001    2006    2007    2014    2020
```

5.2 Denzel Washington

```
   nmid = "nm0000243";
    denzel = grabFilmsForPerson(nmid);
    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 384289   130.16
```

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

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

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" );
```


```
    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
```

5.4 Side-by-Side Comparisons

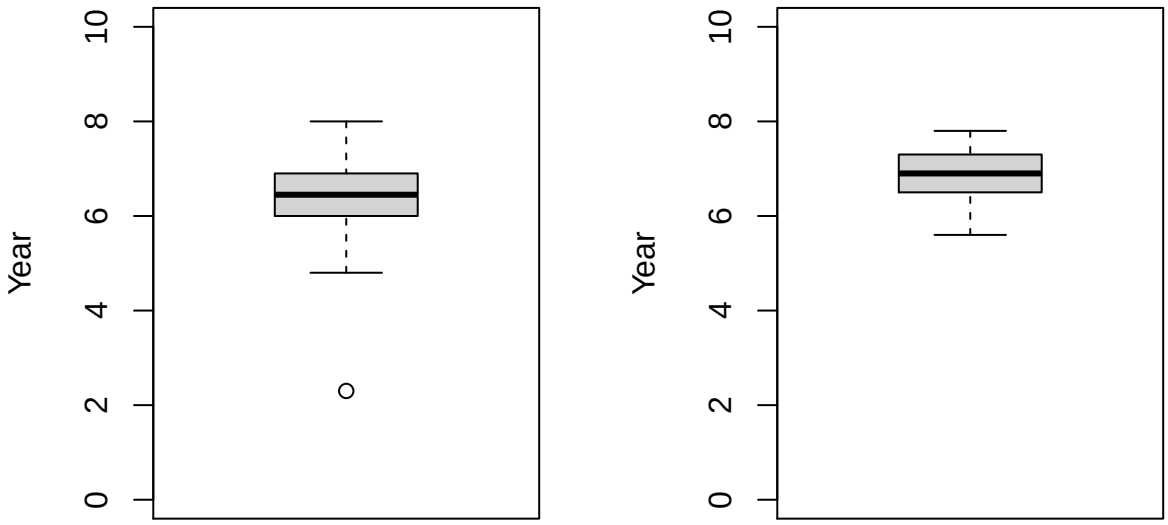
Build side-by-side box plots on several of the variables (including #6) to compare the two movie stars. After each box plot, write 2+ sentence describing what you are seeing, and what conclusions you can logically make. You will need to review what the box plot is showing with the box portion, the divider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

5.4.1 Adjusted Dollars (2000)

5.4.2 Total Votes (Divided by 1,000,000)

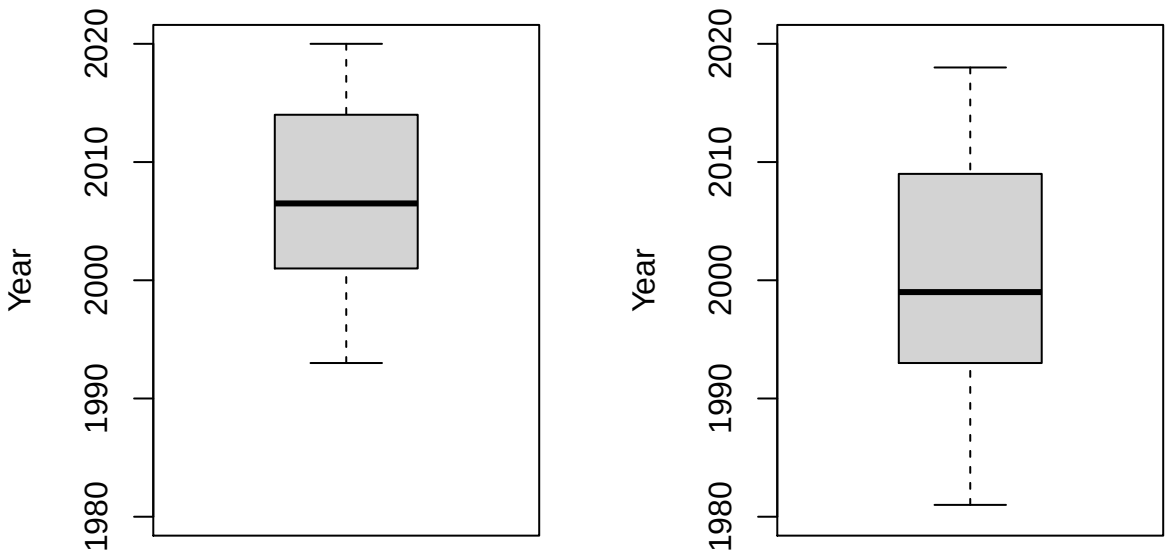
5.4.3 Average Ratings

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



Year? Minutes?

```
par(mfrow=c(1,2))
boxplot(will$movies.50$year, main=will$movies.50$name, ylab = "Year" ,ylim=c(1980,2020))
boxplot(denzel$movies.50$year, main=denzel$movie.50$name, ylab = "Year" ,ylim=c(1980,2020))
```



Metacritic (NA values)