Film Analysis

James Crowley

I have an Excel spreadsheet that contains all of the films that I have seen over the last ~2 years, as well as my ratings of all these films. Ratings reflect how much I enjoyed a film, and are given on a 0 to 5 star scale with half-stars (i.e., 0/5 stars, 0.5/5 stars, 1/5 stars, 1.5/5 stars, etc). The spreadsheet also contains other information about the films as well. Let's read the file into R.

```
library(readxl) # fast excel reader
data.raw <- read_excel("JamesCrowleyMasterCopyMovieRatingsDatabase.xlsx",

Films <- as.data.frame(data.raw)

head(Films)
```

```
##                Movie      Rating Year Released          Director
## 1       Out of Sight   2/5 stars          1998 Steven Soderbergh
## 2    The Dark Knight   5/5 stars          2008 Christopher Nolan
## 3       The Departed   5/5 stars          2006   Martin Scorsese
## 4 The King of Comedy   5/5 stars          1982   Martin Scorsese
## 5       The Prestige 4.5/5 stars          2006 Christopher Nolan
## 6            Arrival   5/5 stars          2016  Denis Villeneuve
##   Director 2 Director 3 Runtime_In_Minutes RT Critic Score
## 1        N/A        N/A                123            0.93
## 2        N/A        N/A                152            0.94
## 3        N/A        N/A                151             0.9
## 4        N/A        N/A                109             0.9
## 5        N/A        N/A                130            0.75
## 6        N/A        N/A                116            0.94
##   RT Critic Reviews Counted RT Audience Score
## 1                        88              0.74
## 2                       326              0.94
## 3                       272              0.94
## 4                        50               0.9
## 5                       191              0.92
## 6                       346              0.82
##   RT Number of Audience Reviews
```

```
## 1                          59617
## 2                        1829253
## 3                         736383
## 4                          26635
## 5                         549931
## 6                          80489
```

We can see that there are 11 columns in this data frame, including the film, the rating I gave it, its director, its runtime, and some information on the film from the popular movie website Rotten Tomatoes.

Now let's restructure some of the columns in the data frame such that they're easier to reference.

```
colnames(Films)[3] <- "Year_Released"
colnames(Films)[5] <- "Director_2"
colnames(Films)[6] <- "Director_3"
colnames(Films)[8] <- "RT_Critic_Score"
colnames(Films)[9] <- "RT_Critic_Reviews_Counted"
colnames(Films)[10] <- "RT_Audience_Score"
colnames(Films)[11] <- "RT_Number_of_Audience_Reviews"
```

```
str(Films)
```

```
## 'data.frame':    351 obs. of  11 variables:
##  $ Movie                        : chr  "Out of Sight" "The Dark Knight
##  $ Rating                       : chr  "2/5 stars" "5/5 stars" "5/5 st
##  $ Year_Released                : num  1998 2008 2006 1982 2006 ...
##  $ Director                     : chr  "Steven Soderbergh" "Christophe
##  $ Director_2                   : chr  "N/A" "N/A" "N/A" "N/A" ...
##  $ Director_3                   : chr  "N/A" "N/A" "N/A" "N/A" ...
##  $ Runtime_In_Minutes           : num  123 152 151 109 130 116 143 124
##  $ RT_Critic_Score              : chr  "0.93" "0.94" "0.9" "0.9" ...
##  $ RT_Critic_Reviews_Counted    : num  88 326 272 50 191 346 212 223 2
##  $ RT_Audience_Score            : chr  "0.74" "0.94" "0.94" "0.9" ...
##  $ RT_Number_of_Audience_Reviews: chr  "59617" "1829253" "736383" "266
```

We'll also convert the Rotten Tomatoes Critic Score, Audience Score, and Audience Number of Reviews to numeric.

```
Films$RT_Critic_Score <- as.numeric(Films$RT_Critic_Score)
```

```
## Warning: NAs introduced by coercion
```

```
Films$RT_Audience_Score <- as.numeric(Films$RT_Audience_Score)
```

```
## Warning: NAs introduced by coercion
```

```
Films$RT_Number_of_Audience_Reviews <- as.numeric(Films$RT_Number_of_Audi
```

```
## Warning: NAs introduced by coercion
```

```
str(Films)
```

```
## 'data.frame':    351 obs. of  11 variables:
##  $ Movie                       : chr  "Out of Sight" "The Dark Knight
##  $ Rating                      : chr  "2/5 stars" "5/5 stars" "5/5 st
##  $ Year_Released               : num  1998 2008 2006 1982 2006 ...
##  $ Director                    : chr  "Steven Soderbergh" "Christophe
##  $ Director_2                  : chr  "N/A" "N/A" "N/A" "N/A" ...
##  $ Director_3                  : chr  "N/A" "N/A" "N/A" "N/A" ...
##  $ Runtime_In_Minutes          : num  123 152 151 109 130 116 143 124
##  $ RT_Critic_Score             : num  0.93 0.94 0.9 0.9 0.75 0.94 0.6
##  $ RT_Critic_Reviews_Counted   : num  88 326 272 50 191 346 212 223 2
##  $ RT_Audience_Score           : num  0.74 0.94 0.94 0.9 0.92 0.82 0.
##  $ RT_Number_of_Audience_Reviews: num  59617 1829253 736383 26635 5499
```

This all looks good, but the biggest issue is that the 'Rating' column contains the string 'X/X stars' in it. If we want to do some analysis on the ratings I give each film, we will have to clean this column up such that it contains only numeric values.

```
#Correct Rating column
Rating <- Films$Rating
```

```
Rating <- gsub( "/5 stars", "", as.character(Rating))
Rating <- as.numeric(Rating)

Films$Rating <- Rating
str(Films)
```

```
## 'data.frame':    351 obs. of  11 variables:
##  $ Movie                       : chr  "Out of Sight" "The Dark Knight
##  $ Rating                      : num  2 5 5 5 4.5 5 4.5 3.5 4 5 ...
##  $ Year_Released               : num  1998 2008 2006 1982 2006 ...
##  $ Director                    : chr  "Steven Soderbergh" "Christophe
##  $ Director_2                  : chr  "N/A" "N/A" "N/A" "N/A" ...
##  $ Director_3                  : chr  "N/A" "N/A" "N/A" "N/A" ...
##  $ Runtime_In_Minutes          : num  123 152 151 109 130 116 143 124
##  $ RT_Critic_Score             : num  0.93 0.94 0.9 0.9 0.75 0.94 0.6
##  $ RT_Critic_Reviews_Counted   : num  88 326 272 50 191 346 212 223 2
##  $ RT_Audience_Score           : num  0.74 0.94 0.94 0.9 0.92 0.82 0.
##  $ RT_Number_of_Audience_Reviews: num  59617 1829253 736383 26635 5499
```

Looks good. The last thing we need to do to clean the dataset up involves values that are 'N/A'... NA is a special value in R, but R does not automatically read 'N/A' as NA- in fact it reads 'N/A' as a character data type. Let's change all the 'N/A' values so that R can read them correctly.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(naniar)
```

```
#Replace all occurrences of N/A in Films dataset
Films <- Films %>%
  replace_with_na_all(condition = ~.x == "N/A")

Films <- as.data.frame(Films)

#Quick look at Films data frame
head(Films)
```

```
##                Movie Rating Year_Released          Director Director_2
## 1       Out of Sight    2.0          1998 Steven Soderbergh       <NA>
## 2    The Dark Knight    5.0          2008 Christopher Nolan       <NA>
## 3       The Departed    5.0          2006   Martin Scorsese       <NA>
## 4 The King of Comedy    5.0          1982   Martin Scorsese       <NA>
## 5       The Prestige    4.5          2006 Christopher Nolan       <NA>
## 6            Arrival    5.0          2016  Denis Villeneuve       <NA>
##   Director_3 Runtime_In_Minutes RT_Critic_Score RT_Critic_Reviews_Coun
## 1       <NA>                123            0.93
## 2       <NA>                152            0.94
## 3       <NA>                151            0.90
## 4       <NA>                109            0.90
## 5       <NA>                130            0.75
## 6       <NA>                116            0.94
##   RT_Audience_Score RT_Number_of_Audience_Reviews
## 1              0.74                         59617
## 2              0.94                       1829253
## 3              0.94                        736383
## 4              0.90                         26635
## 5              0.92                        549931
## 6              0.82                         80489
```

Let's move on to some analysis. Say we are interested in the correlation coefficients of each numeric variable in relation to one another. For example, when 'Year_Released' increases, does 'Rating' also increase with it? Let's take a look.

```
my_data <- Films[, c(2,3,7,8,9,10,11)]
library("Hmisc")
```

```
## Loading required package: lattice

## Loading required package: survival
```

```
## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
#Correlation matrix
res2 <- rcorr(as.matrix(my_data))
res2
```

```
##                              Rating Year_Released Runtime_In_Minutes
## Rating                         1.00          0.17                0.05
## Year_Released                  0.17          1.00                0.03
## Runtime_In_Minutes             0.05          0.03                1.00
## RT_Critic_Score                0.08         -0.36               -0.03
## RT_Critic_Reviews_Counted      0.12          0.73                0.11
## RT_Audience_Score              0.16         -0.33                0.09
## RT_Number_of_Audience_Reviews  0.09          0.04                0.06
##                              RT_Critic_Score RT_Critic_Reviews_Counte
## Rating                                  0.08                      0.1
## Year_Released                          -0.36                      0.7
## Runtime_In_Minutes                     -0.03                      0.1
## RT_Critic_Score                         1.00                     -0.1
## RT_Critic_Reviews_Counted              -0.17                      1.0
## RT_Audience_Score                       0.62                     -0.2
## RT_Number_of_Audience_Reviews          -0.04                      0.0
##                              RT_Audience_Score
## Rating                                    0.16
## Year_Released                            -0.33
## Runtime_In_Minutes                        0.09
## RT_Critic_Score                           0.62
## RT_Critic_Reviews_Counted                -0.22
## RT_Audience_Score                         1.00
```

```
## RT_Number_of_Audience_Reviews             -0.09
##                                RT_Number_of_Audience_Reviews
## Rating                                                  0.09
## Year_Released                                           0.04
## Runtime_In_Minutes                                      0.06
## RT_Critic_Score                                        -0.04
## RT_Critic_Reviews_Counted                               0.08
## RT_Audience_Score                                      -0.09
## RT_Number_of_Audience_Reviews                           1.00
##
## n
##                                Rating Year_Released Runtime_In_Minutes
## Rating                            351           351                351
## Year_Released                     351           351                351
## Runtime_In_Minutes                351           351                351
## RT_Critic_Score                   349           349                349
## RT_Critic_Reviews_Counted         351           351                351
## RT_Audience_Score                 349           349                349
## RT_Number_of_Audience_Reviews     349           349                349
##                                RT_Critic_Score RT_Critic_Reviews_Counte
## Rating                                     349                      35
## Year_Released                              349                      35
## Runtime_In_Minutes                         349                      35
## RT_Critic_Score                            349                      34
## RT_Critic_Reviews_Counted                  349                      35
## RT_Audience_Score                          347                      34
## RT_Number_of_Audience_Reviews              347                      34
##                                RT_Audience_Score
## Rating                                       349
## Year_Released                                349
## Runtime_In_Minutes                           349
## RT_Critic_Score                              347
## RT_Critic_Reviews_Counted                    349
## RT_Audience_Score                            349
## RT_Number_of_Audience_Reviews                349
##                                RT_Number_of_Audience_Reviews
## Rating                                                   349
## Year_Released                                            349
## Runtime_In_Minutes                                       349
## RT_Critic_Score                                          347
## RT_Critic_Reviews_Counted                                349
## RT_Audience_Score                                        349
## RT_Number_of_Audience_Reviews                            349
##
## P
```

```
##                                Rating Year_Released Runtime_In_Minutes
## Rating                                       0.0012             0.3368
## Year_Released                  0.0012                           0.6133
## Runtime_In_Minutes             0.3368 0.6133
## RT_Critic_Score                0.1163 0.0000                     0.5617
## RT_Critic_Reviews_Counted      0.0246 0.0000                     0.0315
## RT_Audience_Score              0.0034 0.0000                     0.0944
## RT_Number_of_Audience_Reviews  0.1046 0.4294                     0.2856
##                                RT_Critic_Score RT_Critic_Reviews_Counte
## Rating                         0.1163          0.0246
## Year_Released                  0.0000          0.0000
## Runtime_In_Minutes             0.5617          0.0315
## RT_Critic_Score                                0.0018
## RT_Critic_Reviews_Counted      0.0018
## RT_Audience_Score              0.0000          0.0000
## RT_Number_of_Audience_Reviews  0.4479          0.1459
##                                RT_Audience_Score
## Rating                         0.0034
## Year_Released                  0.0000
## Runtime_In_Minutes             0.0944
## RT_Critic_Score                0.0000
## RT_Critic_Reviews_Counted      0.0000
## RT_Audience_Score
## RT_Number_of_Audience_Reviews 0.0875
##                                RT_Number_of_Audience_Reviews
## Rating                         0.1046
## Year_Released                  0.4294
## Runtime_In_Minutes             0.2856
## RT_Critic_Score                0.4479
## RT_Critic_Reviews_Counted      0.1459
## RT_Audience_Score              0.0875
## RT_Number_of_Audience_Reviews
```

We can see that the strongest correlations are 'Year_Released' and 'RT_Critic_Reviews_Counted' (0.73 r-squared), 'RT_Critic_Score and RT_Audience_Score' (0.62 r-squared), 'RT_Critic_Score' and 'Year_Released' (-0.36 r-squared), and 'RT_Audience_Score' and 'Year_Released' (-0.33 r-squared).

In other words, as the release year of a film increases, there is generally a higher amount of critics that have posted reviews of the film, a lower percentage score on the Rotten Tomatoes Critic Tomatometer, and a lower percentage score on the Rotten Tomatoes Audience Tomatometer. This largely makes intuitive sense. The Critic Score and Audience Score on Rotten Tomatoes also move in moderate conjunction with one

another.

It's interesting that the 'Rating' column is not linearlly related with any other variable.

The correlation output in R was kind of messy. Let's make it look more presentable.

```
#format the correlation matrix into a table with 4 columns containing :
#Column 1 : row names (variable 1 for the correlation test)
#Column 2 : column names (variable 2 for the correlation test)
#Column 3 : the correlation coefficients
#Column 4 : the p-values of the correlations
flattenCorrMatrix <- function(cormat, pmat) {
  ut <- upper.tri(cormat)
  data.frame(
    row = rownames(cormat)[row(cormat)[ut]],
    column = rownames(cormat)[col(cormat)[ut]],
    cor  =(cormat)[ut],
    p = pmat[ut]
  )
}

flattenCorrMatrix(res2$r, res2$P)
```

```
##                          row                          column         cor
## 1                     Rating                   Year_Released  0.17276793
## 2                     Rating              Runtime_In_Minutes  0.05141244
## 3              Year_Released              Runtime_In_Minutes  0.02706590
## 4                     Rating                 RT_Critic_Score  0.08422249
## 5              Year_Released                 RT_Critic_Score -0.35578149
## 6         Runtime_In_Minutes                 RT_Critic_Score -0.03116914
## 7                     Rating     RT_Critic_Reviews_Counted  0.11997054
## 8              Year_Released     RT_Critic_Reviews_Counted  0.73406316
## 9         Runtime_In_Minutes     RT_Critic_Reviews_Counted  0.11481020
## 10           RT_Critic_Score     RT_Critic_Reviews_Counted -0.16644316
## 11                    Rating               RT_Audience_Score  0.15629236
## 12             Year_Released               RT_Audience_Score -0.33232195
## 13        Runtime_In_Minutes               RT_Audience_Score  0.08968130
## 14           RT_Critic_Score               RT_Audience_Score  0.62312727
## 15 RT_Critic_Reviews_Counted               RT_Audience_Score -0.22459958
## 16                    Rating RT_Number_of_Audience_Reviews  0.08702213
## 17             Year_Released RT_Number_of_Audience_Reviews  0.04243450
## 18        Runtime_In_Minutes RT_Number_of_Audience_Reviews  0.05732338
## 19           RT_Critic_Score RT_Number_of_Audience_Reviews -0.04086828
```

```
## 20 RT_Critic_Reviews_Counted RT_Number_of_Audience_Reviews  0.07800123
## 21         RT_Audience_Score RT_Number_of_Audience_Reviews -0.09160704
##                p
## 1  1.155143e-03
## 2  3.368483e-01
## 3  6.133038e-01
## 4  1.162865e-01
## 5  7.486012e-12
## 6  5.616856e-01
## 7  2.459189e-02
## 8  0.000000e+00
## 9  3.152271e-02
## 10 1.808197e-03
## 11 3.419000e-03
## 12 1.916387e-10
## 13 9.437723e-02
## 14 0.000000e+00
## 15 2.285076e-05
## 16 1.045989e-01
## 17 4.293795e-01
## 18 2.855541e-01
## 19 4.479352e-01
## 20 1.458987e-01
## 21 8.748506e-02
```

We can see that all of the variable pairs that were correlated are statistically significant (p-value < 0.05).

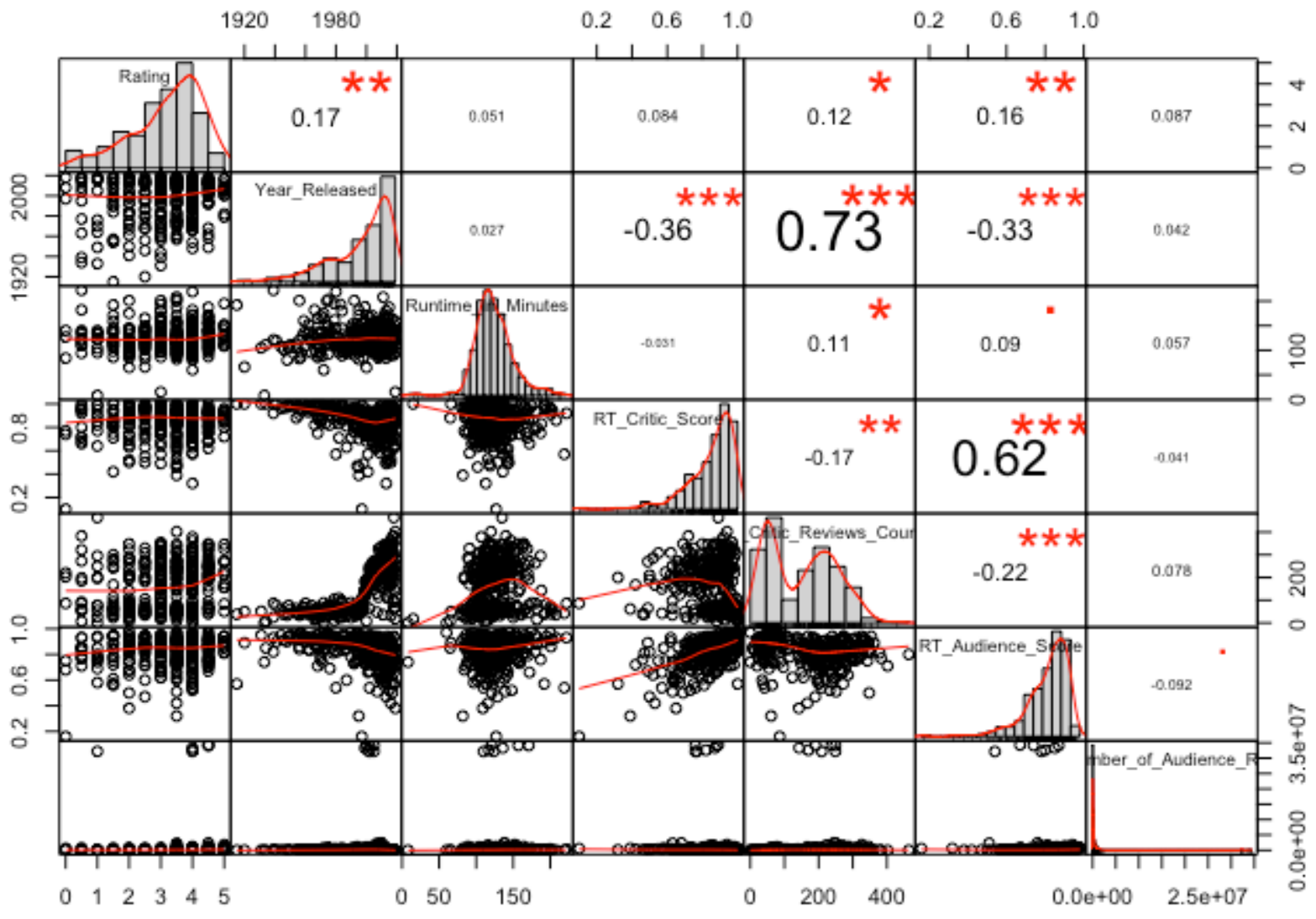
Here is a plot of the correlation matrix. In the plot, the distribution of each variable is shown on the diagonal. On the bottom of the diagonal is the bivariate scatter plots with a fitted line. On the top of the diagonal, the value of the correlation is shown, plus the significance level as stars. Each significance level is associated with a symbol (the greater the amount of stars, the lower the p-value).

```
library("PerformanceAnalytics")
```

```
## Loading required package: xts

## Loading required package: zoo

##
```

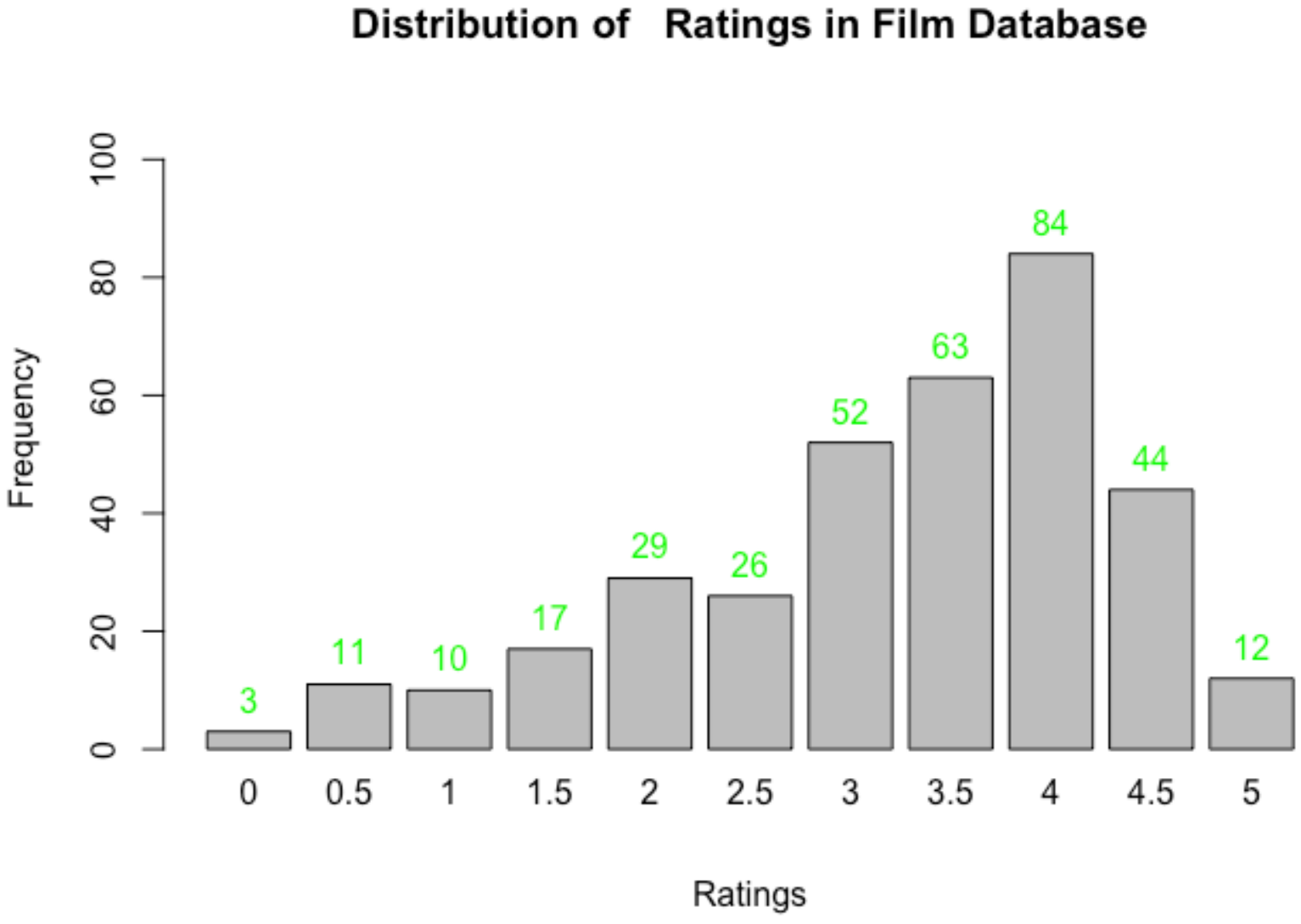
```
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

##
## Attaching package: 'xts'

## The following objects are masked from 'package:dplyr':
##
##     first, last

##
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':
##
##     legend
```

```r
#Chart of the correlation matrix
chart.Correlation(my_data, histogram=TRUE, pch=19)
```



Let's move on to an analysis of the ratings I gave each movie.

```
#Create some vectors that will be used in barplot
sorted <- sort(unique(Rating))
Rating <- factor(Films$Rating, levels = c(0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5,


freq <- summary(Rating)


freq <- as.numeric(as.character(freq))

#Create a barplot with the distribution of ratings in my film database
ylim <- c(0, 1.3*max(freq))
xx <- barplot(freq, xaxt = 'n', xlab = 'Ratings', width = 1, ylim = ylim,

text(x = xx, y = freq, label = freq, pos = 3, cex = 1, col = "green", xla
## Add x-axis labels
axis(1, at = xx, labels = sorted, tick=FALSE, las=1, line=-0.5, cex.axis=
```



We can see that most of the films that I've watched have been given a positive review (Rating >= 3/5 stars), as the barplot is skewed heavily to the right. I've only given 12 films a perfect rating of 5/5 stars, and only 3 films a rating of 0/5 stars.

How do films match up by decade?

```
#Average ratings of films by decade
w <- Films %>%
  mutate(Decade = cut(Year_Released,
                      breaks = c(1909, 1919, 1929, 1939, 1949, 1959, 1969
                                 2009, 2019),
                      labels=c("1910-1919", "1920-1929", "1930-1939", "19
                               "1960-1969", "1970-1979", "1980-1989", "19
                               "2010-2019"))) %>%
  group_by(Decade) %>%
  summarise(AvgRating = mean(Rating), NumberOfFilms=n()) %>%
  arrange(desc(AvgRating)) %>%
  print(n = Inf)
```

```
## # A tibble: 11 x 3
##    Decade    AvgRating NumberOfFilms
##    <fct>         <dbl>         <int>
##  1 2000-2009      3.52            72
##  2 2010-2019      3.27           136
##  3 1990-1999      3.22            49
##  4 1980-1989      3.20            22
##  5 1970-1979      3.2             30
##  6 1960-1969      2.92            18
##  7 1940-1949      2.83             6
##  8 1950-1959      2.77            11
##  9 1920-1929      2.5              1
## 10 1930-1939      1.6              5
## 11 1910-1919      1.5              1
```

```
w <- as.data.frame(w)
w$AvgRating <- round(w$AvgRating, digits = 2)
w
```

```
##       Decade AvgRating NumberOfFilms
## 1  2000-2009      3.52            72
## 2  2010-2019      3.27           136
## 3  1990-1999      3.22            49
## 4  1980-1989      3.20            22
## 5  1970-1979      3.20            30
## 6  1960-1969      2.92            18
## 7  1940-1949      2.83             6
## 8  1950-1959      2.77            11
## 9  1920-1929      2.50             1
## 10 1930-1939      1.60             5
## 11 1910-1919      1.50             1
```

We can see that most of the films that I've watched have come in the current millenium, and that the highest rated decade is 2000-2009. The '80s and '70s decades have the exact same rating (3.20/5 stars), despite the fact that I've seen 8 more movies in the '70s (30 films) than in the '80s (22 films).

Let's look at the ratings I've given of films by different directors. Only directors whom I've seen 4 or more of their films will be counted in this analysis.

```r
#Average rating of films by a director
s <- Films %>%
  group_by(Director) %>%
  summarise(AvgRating = mean(Rating), NumberOfFilms = n()) %>%
  filter(NumberOfFilms >= 4) %>%
  arrange(desc(AvgRating)) %>%
  print(n = Inf)
```

```
## # A tibble: 16 x 3
##    Director             AvgRating NumberOfFilms
##    <chr>                    <dbl>         <int>
##  1 Christopher Nolan         4.39             9
##  2 Denis Villeneuve          4                4
##  3 Francis Ford Coppola      3.62             4
##  4 Steven Soderbergh         3.62             4
##  5 Quentin Tarantino         3.57             7
##  6 Martin Scorsese           3.53            15
##  7 Ridley Scott              3.5              5
##  8 Terry Gilliam             3.36             7
##  9 Jim Jarmusch              3.3              5
## 10 Clint Eastwood            3.25             4
## 11 Joel Coen                 3.23            11
## 12 Paul Thomas Anderson      3.17             6
## 13 Darren Aronofsky          3                4
## 14 Stanley Kubrick           2.81             8
## 15 Alfred Hitchcock          2.72             9
## 16 Steven Spielberg          2.08             6
```

```r
s <- as.data.frame(s)
s$AvgRating <- round(s$AvgRating, digits = 2)
s
```

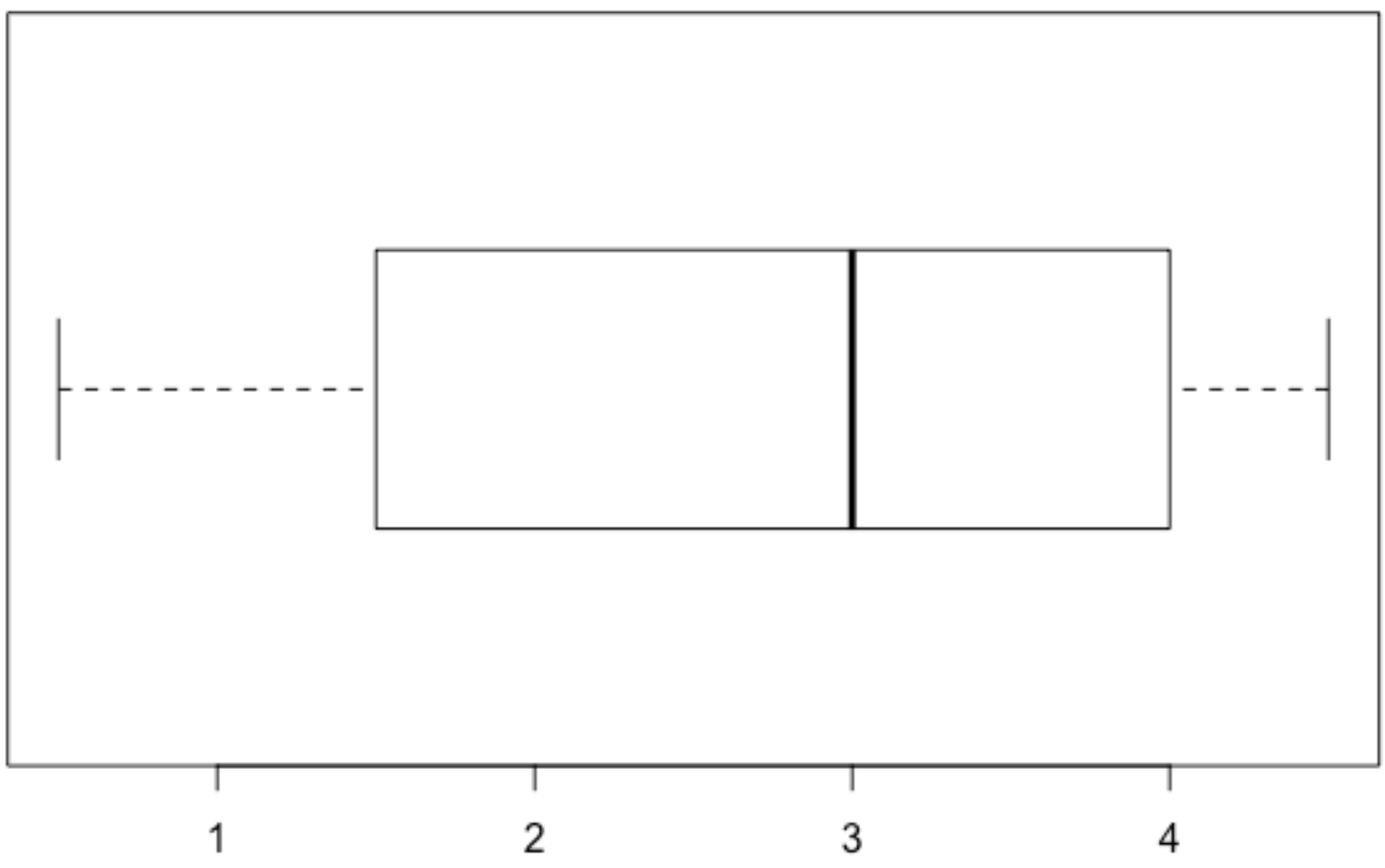
```
##                Director AvgRating NumberOfFilms
## 1     Christopher Nolan      4.39             9
## 2      Denis Villeneuve      4.00             4
## 3  Francis Ford Coppola      3.62             4
## 4     Steven Soderbergh      3.62             4
## 5     Quentin Tarantino      3.57             7
## 6       Martin Scorsese      3.53            15
## 7          Ridley Scott      3.50             5
## 8        Terry Gilliam      3.36             7
```

```
## 9          Jim Jarmusch      3.30             5
## 10       Clint Eastwood      3.25             4
## 11            Joel Coen      3.23            11
## 12 Paul Thomas Anderson      3.17             6
## 13     Darren Aronofsky      3.00             4
## 14      Stanley Kubrick      2.81             8
## 15     Alfred Hitchcock      2.72             9
## 16     Steven Spielberg      2.08             6
```

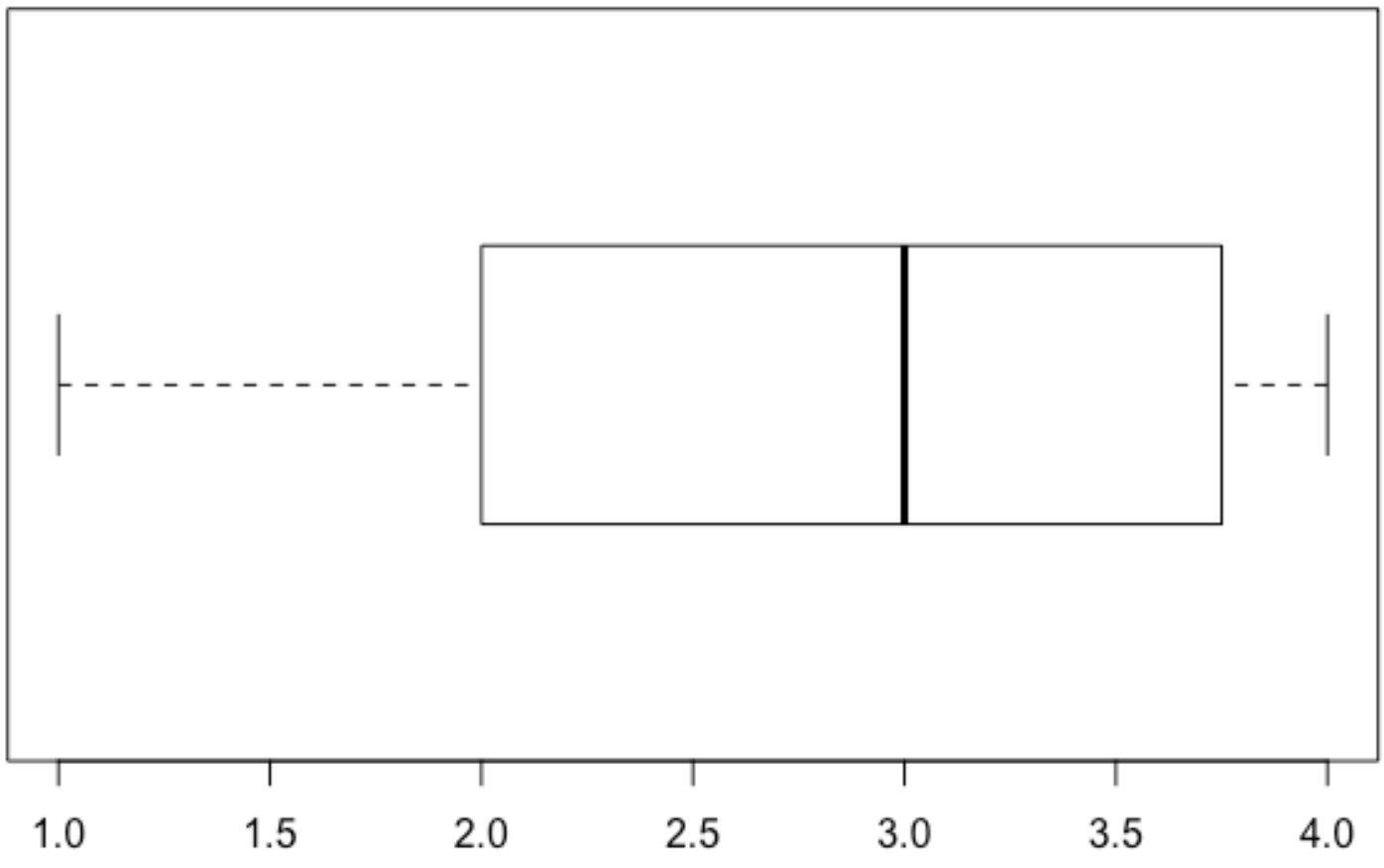
Some great directors on this list, and while I wouldn't say that Christopher Nolan is my favorite film director, he's definitely up there.

Two of my favorite film directors appear towards the bottom of the list in Stanley Kubrick and Alfred Hitchcock (I do genuinely hate Steven Spielberg). Why are Kubrick and Hitchcock so low? Let's take a look at the distribution of ratings for the two directors.

```
#Looking at distribution of Kubrick and Hitchcock films
Kubrick <- subset(Films, Director == "Stanley Kubrick")

Hitchcock <- subset(Films, Director == "Alfred Hitchcock")

boxplot(Hitchcock$Rating, horizontal=TRUE, main="Alfred Hitchcock Film Ra
```




```
boxplot(Kubrick$Rating, horizontal=TRUE, main="Stanley Kubrick Film Ratin
```



We can see that, despite having different average ratings, both Kubrick and Hitchcock have the exact same median film rating of 3.00/5 stars.

Let's use the raster package to look at the coefficient of variation (CV) of each director.

```
library("raster")
```

```
## Loading required package: sp

##
## Attaching package: 'raster'

## The following objects are masked from 'package:Hmisc':
##
##     mask, zoom

## The following object is masked from 'package:dplyr':
##
##     select
```

```
f <- Films %>%
  group_by(Director) %>%
  summarise(CoefficientOfVariation = cv(Rating),
            NumberOfFilms = n()) %>%
  filter(NumberOfFilms >= 4) %>%
  arrange(desc(CoefficientOfVariation)) %>%
  print(n = Inf)
```

```
## # A tibble: 16 x 3
##    Director             CoefficientOfVariation NumberOfFilms
##    <chr>                                 <dbl>         <int>
##  1 Steven Spielberg                       71.9             6
##  2 Alfred Hitchcock                       49.6             9
##  3 Quentin Tarantino                      44.6             7
##  4 Clint Eastwood                         40.7             4
##  5 Stanley Kubrick                        40.2             8
##  6 Darren Aronofsky                       38.5             4
##  7 Francis Ford Coppola                   34.5             4
##  8 Steven Soderbergh                      34.5             4
##  9 Martin Scorsese                        32.3            15
## 10 Jim Jarmusch                           31.4             5
## 11 Paul Thomas Anderson                   29.4             6
## 12 Ridley Scott                           26.7             5
## 13 Joel Coen                              26.3            11
## 14 Denis Villeneuve                       22.8             4
## 15 Terry Gilliam                          22.3             7
## 16 Christopher Nolan                       9.49            9
```

Kubrick and Hitchcock both appear in the top 5 with CV's of 40.2% and 49.6%, respectively. There is a lot of variation in the quality of their films (according to me), thus bringing their average rating down. Both have had plenty of great films that I have rated highly, but both have had at least a couple of films I've rated lowly.

Lastly, let's see if there is anything interesting in relation to the runtimes of films and my ratings of them.

```
#Grouping by runtime


r <- Films %>%
```

```
  mutate(Runtime = cut(Runtime_In_Minutes,
                       breaks = c(0, 89, 119, 150, 250),
                       labels = c("0-89 minutes", "90-119 minutes", "120-1
           group_by(Runtime) %>%
           summarise(AvgRating = mean(Rating), NumberOfFilms=n()) %>%
           arrange(desc(AvgRating)) %>%
           print(n = Inf)
```

```
## # A tibble: 4 x 3
##   Runtime          AvgRating NumberOfFilms
##   <fct>                <dbl>         <int>
## 1 Beyond 2.5 hours      3.49            50
## 2 90-119 minutes        3.29           144
## 3 120-150 minutes       3.14           140
## 4 0-89 minutes          2.76            17
```

Apparently, I enjoy long films, with films that were longer than 2.5 hours being given an average rating 3.49/5 stars.