Modeling and prediction for movies

Setup

Load packages

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
library(statsr)
```

```
## Warning: package 'BayesFactor' was built under R version 4.1.2
```

```
## Warning: package 'coda' was built under R version 4.1.2
```

```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.1.3
```

```
library(Metrics)
```

```
## Warning: package 'Metrics' was built under R version 4.1.3
```

Load data

Make sure your data and R Markdown files are in the same directory. When loaded your data file will be called `movies` . Delete this note when before you submit your work.

```
load("movies.Rdata")

movies %>% head(5)
```

```
## # A tibble: 5 x 32
##   title title_type genre runtime mpaa_rating studio thtr_rel_year thtr_rel_month
##   <chr> <fct>      <fct>   <dbl> <fct>       <fct>          <dbl>          <dbl>
## 1 Fill~ Feature F~ Drama      80 R           Indom~          2013              4
## 2 The ~ Feature F~ Drama     101 PG-13       Warne~          2001              3
## 3 Wait~ Feature F~ Come~      84 R           Sony ~          1996              8
## 4 The ~ Feature F~ Drama     139 PG          Colum~          1993             10
## 5 Male~ Feature F~ Horr~      90 R           Ancho~          2004              9
## # ... with 24 more variables: thtr_rel_day <dbl>, dvd_rel_year <dbl>,
## #   dvd_rel_month <dbl>, dvd_rel_day <dbl>, imdb_rating <dbl>,
## #   imdb_num_votes <int>, critics_rating <fct>, critics_score <dbl>,
## #   audience_rating <fct>, audience_score <dbl>, best_pic_nom <fct>,
## #   best_pic_win <fct>, best_actor_win <fct>, best_actress_win <fct>,
## #   best_dir_win <fct>, top200_box <fct>, director <chr>, actor1 <chr>,
## #   actor2 <chr>, actor3 <chr>, actor4 <chr>, actor5 <chr>, imdb_url <chr>, ...
```

Part 1: Data

As per the dataset documentation, "*The data set is comprised of 651 randomly sampled movies produced and released before 2016*". Therefore, as we have a random sample, we can generalize the data to other movies. But being an observational dataset, there's no causality, only association possible.

One potential bias source is that the movies not present in the IMDB cannot be selected.

```
# Dataset shape (rows, columns)
print( paste('This dataset has', dim(movies)[1], 'rows and', dim(movies)[2], 'columns.') )
```

```
## [1] "This dataset has 651 rows and 32 columns."
```

```
# Types of each variable
str(movies)
```

```
## tibble [651 x 32] (S3: tbl_df/tbl/data.frame)
##  $ title           : chr [1:651] "Filly Brown" "The Dish" "Waiting for Guffman" "The Age of Innocence" ...
##  $ title_type      : Factor w/ 3 levels "Documentary",..: 2 2 2 2 2 1 2 2 1 2 ...
##  $ genre           : Factor w/ 11 levels "Action & Adventure",..: 6 6 4 6 7 5 6 6 5 6 ...
##  $ runtime         : num [1:651] 80 101 84 139 90 78 142 93 88 119 ...
##  $ mpaa_rating     : Factor w/ 6 levels "G","NC-17","PG",..: 5 4 5 3 5 6 4 5 6 6 ...
##  $ studio          : Factor w/ 211 levels "20th Century Fox",..: 91 202 167 34 13 163 147 118 88 84 ...
##  $ thtr_rel_year   : num [1:651] 2013 2001 1996 1993 2004 ...
##  $ thtr_rel_month  : num [1:651] 4 3 8 10 9 1 1 11 9 3 ...
##  $ thtr_rel_day    : num [1:651] 19 14 21 1 10 15 1 8 7 2 ...
##  $ dvd_rel_year    : num [1:651] 2013 2001 2001 2001 2005 ...
##  $ dvd_rel_month   : num [1:651] 7 8 8 11 4 4 2 3 1 8 ...
##  $ dvd_rel_day     : num [1:651] 30 28 21 6 19 20 18 2 21 14 ...
##  $ imdb_rating     : num [1:651] 5.5 7.3 7.6 7.2 5.1 7.8 7.2 5.5 7.5 6.6 ...
##  $ imdb_num_votes  : int [1:651] 899 12285 22381 35096 2386 333 5016 2272 880 12496 ...
##  $ critics_rating  : Factor w/ 3 levels "Certified Fresh",..: 3 1 1 1 3 2 3 3 2 1 ...
##  $ critics_score   : num [1:651] 45 96 91 80 33 91 57 17 90 83 ...
##  $ audience_rating : Factor w/ 2 levels "Spilled","Upright": 2 2 2 2 1 2 2 1 2 2 ...
##  $ audience_score  : num [1:651] 73 81 91 76 27 86 76 47 89 66 ...
##  $ best_pic_nom    : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ best_pic_win    : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ best_actor_win  : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 2 1 1 ...
##  $ best_actress_win: Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ best_dir_win    : Factor w/ 2 levels "no","yes": 1 1 1 2 1 1 1 1 1 1 ...
##  $ top200_box      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ director        : chr [1:651] "Michael D. Olmos" "Rob Sitch" "Christopher Guest" "Martin Scorsese" ...
##  $ actor1          : chr [1:651] "Gina Rodriguez" "Sam Neill" "Christopher Guest" "Daniel Day-Lewis" ...
##  $ actor2          : chr [1:651] "Jenni Rivera" "Kevin Harrington" "Catherine O'Hara" "Michelle Pfeiffer" ...
##  $ actor3          : chr [1:651] "Lou Diamond Phillips" "Patrick Warburton" "Parker Posey" "Winona Ryder" ...
##  $ actor4          : chr [1:651] "Emilio Rivera" "Tom Long" "Eugene Levy" "Richard E. Grant" ...
##  $ actor5          : chr [1:651] "Joseph Julian Soria" "Genevieve Mooy" "Bob Balaban" "Alec McCowen" ...
##  $ imdb_url        : chr [1:651] "http://www.imdb.com/title/tt1869425/" "http://www.imdb.com/title/tt0205873/" "http://ww
w.imdb.com/title/tt0118111/" "http://www.imdb.com/title/tt0106226/" ...
##  $ rt_url          : chr [1:651] "//www.rottentomatoes.com/m/filly_brown_2012/" "//www.rottentomatoes.com/m/dish/" "//ww
w.rottentomatoes.com/m/waiting_for_guffman/" "//www.rottentomatoes.com/m/age_of_innocence/" ...
```

While I am sure we could explore this dataset in many ways, for the purpose of this project that is a Linear Regression Analysis, I will initially drop some columns that will not matter for the study. The variables dropped are: * actor1, actor2, actor3, actor4, actor5 * imdb_url and rt_url * studio * director

```
# Drop variables and assign to a new dataset name to preserve the original data
movies2 <- movies %>%
  select( !c(actor1, actor2, actor3, actor4, actor5, director, imdb_url, rt_url, studio) )
```

Part 2: Research question

The present study will perform a **Multiple Linear Regression Analysis**, seeking answer to the question:

What variables from the dataset can better explain the variance of the IMDB ratings?

Part 3: Exploratory data analysis

The first steps for a good exploratory analysis are to check the distribution of the data and the descriptive statistics.

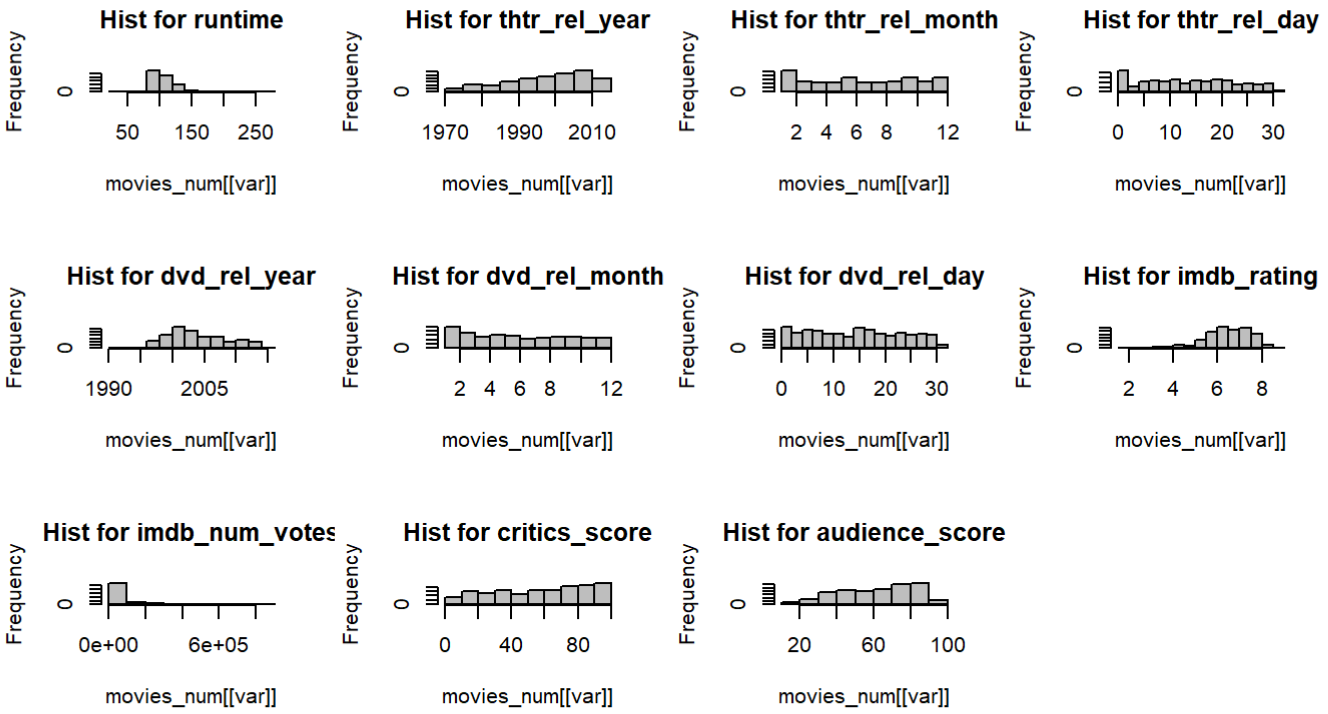
```
# Descriptive Statistics
summary(movies2)
```

```
##     title              title_type                 genre        runtime
##  Length:651         Documentary : 55   Drama             :305   Min.   : 39.0
##  Class :character   Feature Film:591   Comedy            : 87   1st Qu.: 92.0
##  Mode  :character   TV Movie    :  5   Action & Adventure: 65   Median :103.0
##                                        Mystery & Suspense: 59   Mean   :105.8
##                                        Documentary       : 52   3rd Qu.:115.8
##                                        Horror            : 23   Max.   :267.0
##                                        (Other)           : 60   NA's   :1
##   mpaa_rating  thtr_rel_year  thtr_rel_month   thtr_rel_day    dvd_rel_year
##  G      : 19   Min.   :1970   Min.   : 1.00   Min.   : 1.00   Min.   :1991
##  NC-17  :  2   1st Qu.:1990   1st Qu.: 4.00   1st Qu.: 7.00   1st Qu.:2001
##  PG     :118   Median :2000   Median : 7.00   Median :15.00   Median :2004
##  PG-13  :133   Mean   :1998   Mean   : 6.74   Mean   :14.42   Mean   :2004
##  R      :329   3rd Qu.:2007   3rd Qu.:10.00   3rd Qu.:21.00   3rd Qu.:2008
##  Unrated: 50   Max.   :2014   Max.   :12.00   Max.   :31.00   Max.   :2015
##                                                                NA's   :8
##  dvd_rel_month     dvd_rel_day     imdb_rating    imdb_num_votes
##  Min.   : 1.000   Min.   : 1.00   Min.   :1.900   Min.   :   180
##  1st Qu.: 3.000   1st Qu.: 7.00   1st Qu.:5.900   1st Qu.:  4546
##  Median : 6.000   Median :15.00   Median :6.600   Median : 15116
##  Mean   : 6.333   Mean   :15.01   Mean   :6.493   Mean   : 57533
##  3rd Qu.: 9.000   3rd Qu.:23.00   3rd Qu.:7.300   3rd Qu.: 58301
##  Max.   :12.000   Max.   :31.00   Max.   :9.000   Max.   :893008
##  NA's   :8        NA's   :8
##          critics_rating critics_score    audience_rating audience_score
##  Certified Fresh:135    Min.   :  1.00   Spilled:275     Min.   :11.00
##  Fresh          :209    1st Qu.: 33.00   Upright:376     1st Qu.:46.00
##  Rotten         :307    Median : 61.00                   Median :65.00
##                         Mean   : 57.69                   Mean   :62.36
##                         3rd Qu.: 83.00                   3rd Qu.:80.00
##                         Max.   :100.00                   Max.   :97.00
##
##  best_pic_nom best_pic_win best_actor_win best_actress_win best_dir_win
##  no :629      no :644      no :558        no :579          no :608
##  yes: 22      yes:  7      yes: 93        yes: 72          yes: 43
##
##
##
##
##
##  top200_box
##  no :636
##  yes: 15
##
##
##
##
##
```

The dataset looks balanced for many variables. Let's point some insights extracted from these stats: * The *genre* has a spike in Drama movies and the other ones more balanced. * The *year* of release looks more concentrated in recent years. * The *release month* looks balanced throughout the year. Interesting. * *runtime* brings mean and median around 100 minutes. * *dvd release year* is around 4 to 6 years more than the mean of the relase of the film. Maybe that has something to do with the time when DVDs became more popular and cheap, thus more titles started to be released. * The **target variable** *imdb_rating* goes from (lowest) 1-10 (highest) and the mean is around 6.5, what points out that there are slightly more good ratings than bad ones. * The *critics score* is the same thing, with a mean/ median around 6.

Let's see the distributions now.

```
# Selecting only numerical variables
movies_num <- select_if(movies2, is.numeric)

# Creating a figure for the plots
par(mfrow=c(3, 4))


# Plotting histograms
for (var in colnames(movies_num)){
  hist(movies_num[[var]], main=paste('Hist for', var), col='gray' )
}
```



After plotted all of the histograms, we see that most of the variables are skewed. There are no normally distributed variables.

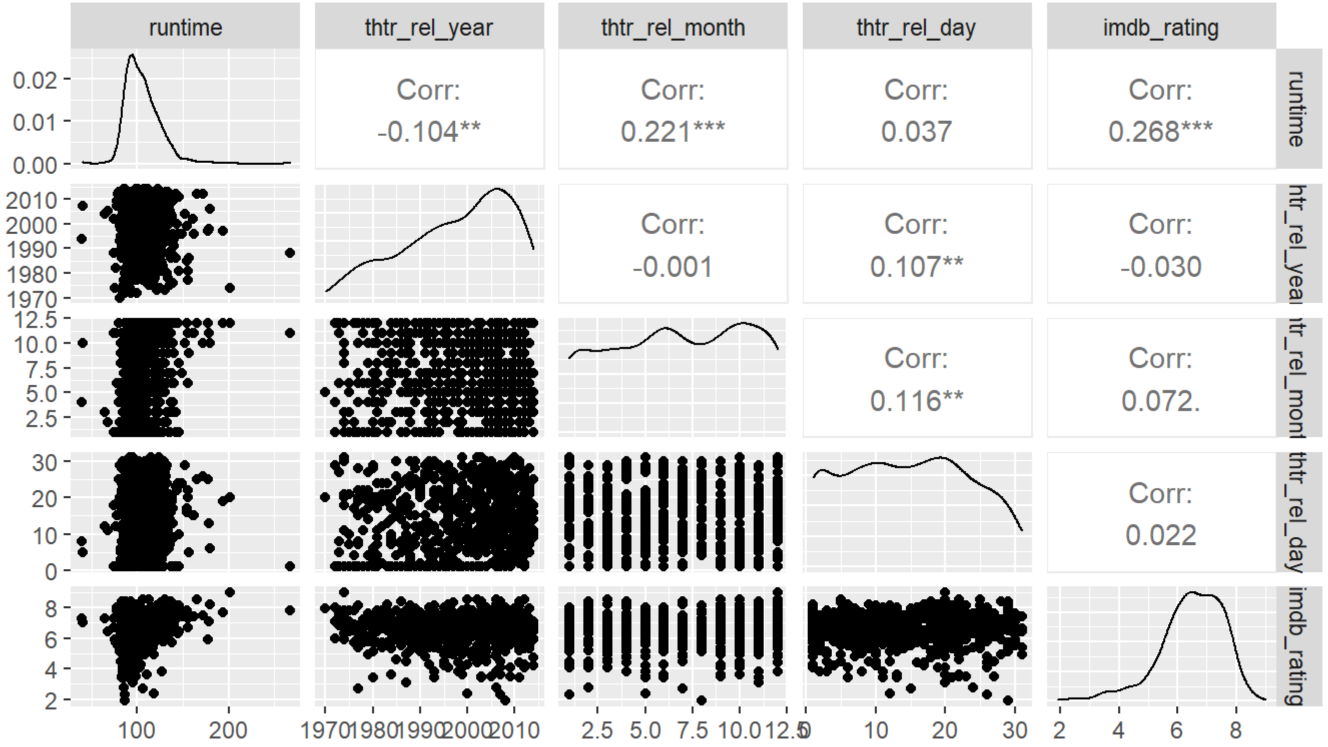
In order to start thinking about the modeling, it is needed to plot the scatterplots and check the relationships between the variables.

```
# Plot scatterplots
ggpairs(movies_num[,c('runtime', 'thtr_rel_year', 'thtr_rel_month', 'thtr_rel_day','imdb_rating')])
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
```

```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removing 1 row that contained a missing value

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removing 1 row that contained a missing value

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removing 1 row that contained a missing value

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removing 1 row that contained a missing value
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
## Removed 1 rows containing missing values (geom_point).
## Removed 1 rows containing missing values (geom_point).
## Removed 1 rows containing missing values (geom_point).
```



```
ggpairs(movies_num[,c('dvd_rel_year', 'dvd_rel_month', 'imdb_num_votes','imdb_rating')])
```

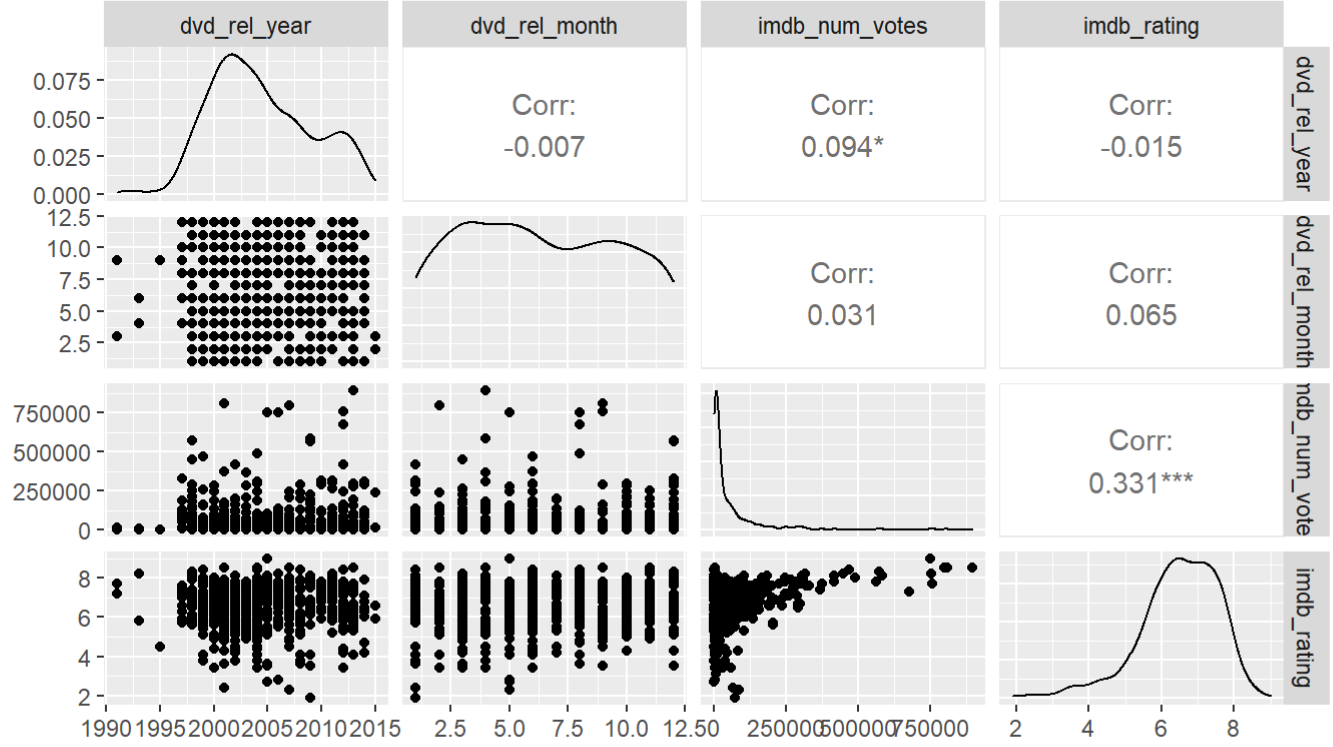
```
## Warning: Removed 8 rows containing non-finite values (stat_density).
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 8 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 8 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 8 rows containing missing values
```

```
## Warning: Removed 8 rows containing missing values (geom_point).
```

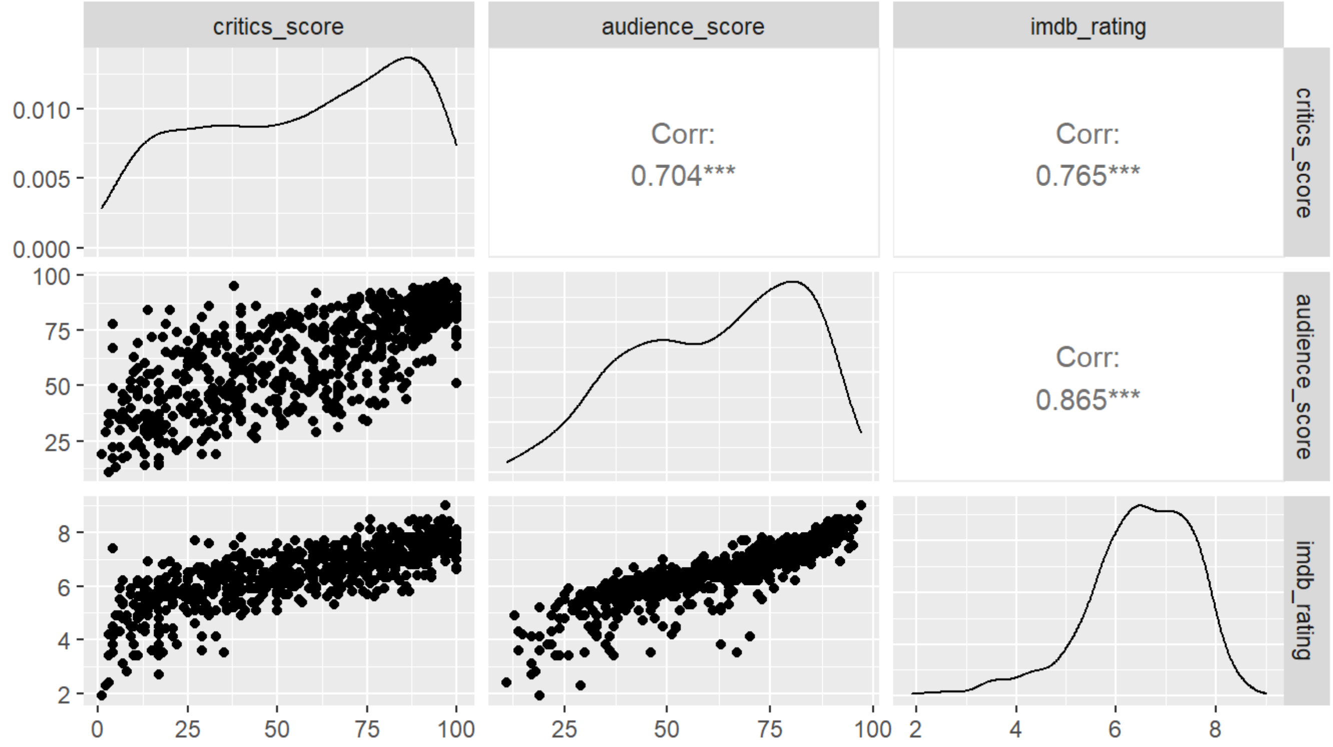
```
## Warning: Removed 8 rows containing non-finite values (stat_density).
```

```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 8 rows containing missing values

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 8 rows containing missing values
```

```
## Warning: Removed 8 rows containing missing values (geom_point).
## Removed 8 rows containing missing values (geom_point).
## Removed 8 rows containing missing values (geom_point).
## Removed 8 rows containing missing values (geom_point).
```



```
ggpairs(movies_num[,c('critics_score', 'audience_score', 'imdb_rating')])
```



With the scatterplots on screen, they present that the highest correlations with the response variable *imdb_ratings* are the scores from other sources, such as *audience_score and critics_score*.

Does the genre interfere on the ratings?

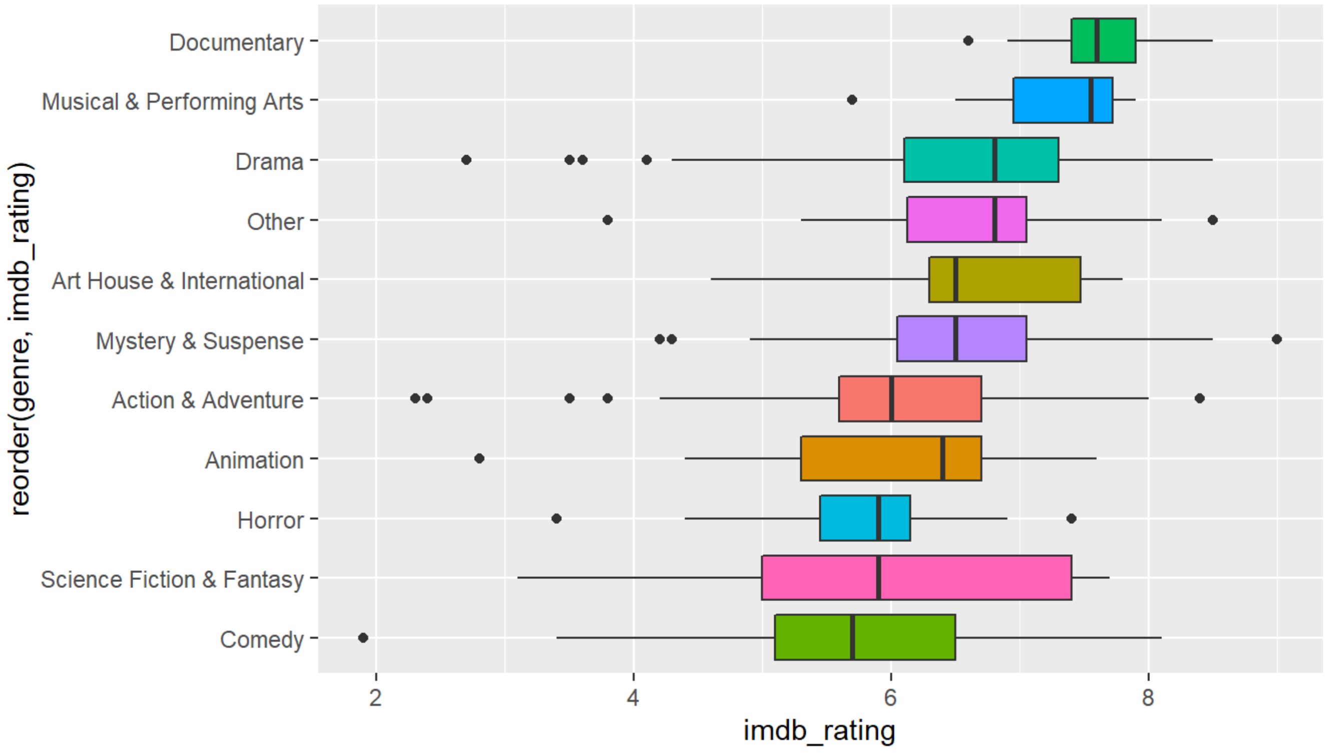
As the ratings are skewed, we will prefer to use the median

```
# Mean ratings by
movies2 %>%
  group_by(genre) %>%
  summarise(median_value = quantile(imdb_rating, 0.5),
            n = n(),
            mean = mean(imdb_rating)) %>%
  arrange(desc(median_value))
```

```
## # A tibble: 11 x 4
##    genre                      median_value     n  mean
##    <fct>                             <dbl> <int> <dbl>
##  1 Documentary                        7.6     52  7.65
##  2 Musical & Performing Arts          7.55    12  7.3
##  3 Drama                              6.8    305  6.67
##  4 Other                              6.8     16  6.63
##  5 Art House & International          6.5     14  6.61
##  6 Mystery & Suspense                 6.5     59  6.48
##  7 Animation                          6.4      9  5.9
##  8 Action & Adventure                 6       65  5.97
##  9 Horror                             5.9     23  5.76
## 10 Science Fiction & Fantasy          5.9      9  5.76
## 11 Comedy                             5.7     87  5.74
```

We can see that the medians are pretty close, there is no much difference. Documentaries, Musicals and Drama lead the ratings.

```
# Boxplot Ratings by genre
ggplot(data=movies2, aes(y=reorder(genre,imdb_rating), x=imdb_rating) ) +
  geom_boxplot(aes(fill=genre), show.legend = FALSE)
```



But, can we say that those averages are statistically different?

```
# ANOVA test for the genre means
anova_genre <- aov(imdb_rating ~ genre, data = movies2)
summary(anova_genre)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## genre        10  174.5  17.446   18.91 <2e-16 ***
## Residuals   640  590.4   0.922
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

With a P value below the significance level of 0.05, **we reject the null hypothesys that the means are equal for all of the groups** and confirm that there is significant difference by genre.

Does the movie being in the top200 influence the ratings?

```
# ANOVA test for the top 200 box
anova_top <- aov(imdb_rating ~ top200_box, data = movies2)
summary(anova_top)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## top200_box    1    6.4   6.425   5.499 0.0193 *
## Residuals   649  758.4   1.169
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

According to the P-Value, yes. Therefore, another variable to be tested in the model.

Does the movie winning the Best picture prize or Best Pic Nomination influence the ratings?

```
# ANOVA test for best picture
anova_bestpic <- aov(imdb_rating ~ best_pic_win, data = movies2)
summary(anova_bestpic)
```

```
##               Df Sum Sq Mean Sq F value   Pr(>F)
## best_pic_win   1   14.0  14.006   12.11 0.000536 ***
## Residuals    649  750.8   1.157
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# ANOVA test for best picture nomination
anova_bestnom <- aov(imdb_rating ~ best_pic_nom, data = movies2)
summary(anova_bestnom)
```

```
##               Df Sum Sq Mean Sq F value   Pr(>F)
## best_pic_nom   1   36.0   35.97   32.03 2.28e-08 ***
## Residuals    649  728.9    1.12
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Yes, it shows that those variables influence the ratings and should be added to the model for evaluation.

Does best actor or director influence the ratings?

```
# ANOVA test for best actor
anova_bestactor <- aov(imdb_rating ~ best_actor_win, data = movies2)
summary(anova_bestactor)
```

```
##                 Df Sum Sq Mean Sq F value Pr(>F)
## best_actor_win   1    3.2   3.189   2.717 0.0998 .
## Residuals      649  761.7   1.174
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# ANOVA test for best actrees
anova_bestactress <- aov(imdb_rating ~ best_actress_win, data = movies2)
summary(anova_bestactress)
```

```
##                   Df Sum Sq Mean Sq F value Pr(>F)
## best_actress_win   1    3.9   3.897   3.324 0.0687 .
## Residuals        649  760.9   1.172
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# ANOVA test for best picture
anova_bestdir <- aov(imdb_rating ~ best_dir_win, data = movies2)
summary(anova_bestdir)
```

```
##               Df Sum Sq Mean Sq F value   Pr(>F)
## best_dir_win   1   13.9  13.865   11.98 0.000572 ***
## Residuals    649  751.0   1.157
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# ANOVA test for Audience Rating
anova_audience <- aov(imdb_rating ~ audience_rating, data = movies2)
summary(anova_audience)
```

```
##                  Df Sum Sq Mean Sq F value Pr(>F)    
## audience_rating   1  369.6   369.6     607 <2e-16 ***
## Residuals       649  395.2     0.6                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

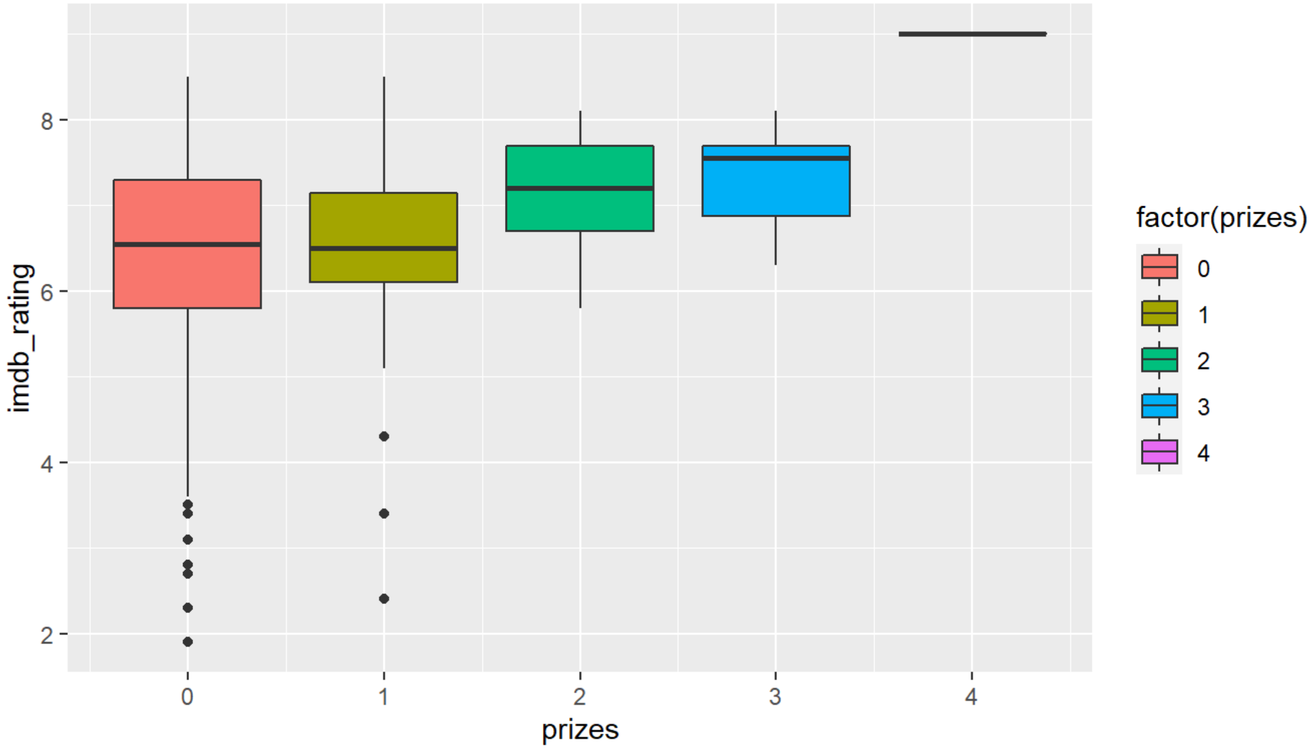
Best director influences. Best actor or actress, does not.

Feature Engineering: Creating a new Variable Prizes

```
# Variable prizes won
movies2$prizes <- as.integer(movies2$best_actor_win) + as.integer(movies2$best_actress_win) + as.integer(movies2$best_pic_wi
         n) + as.integer(movies2$best_dir_win) - 4

# ANOVA test for the new variable prizes
anova_prizes <- aov(imdb_rating ~ prizes, data = movies2)
summary(anova_prizes)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)    
## prizes        1   17.0  16.997   14.75 0.000135 ***
## Residuals   649  747.8   1.152                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Plot boxplots ratings vs prizes
ggplot(data=movies2, aes(x=prizes, y=imdb_rating, group=prizes)) +
  geom_boxplot(aes(fill=factor(prizes)) )
```



The ANOVA test shows that the variable created *prizes* influences the ratings. It will be tested in the model.

Before we go to the modeling, let's get rid of some missing values, identified in the stats.

```
# Check for total NAs
sum(is.na(movies2))
```

```
## [1] 25
```

As there are only 25, I will just go ahead and remove them. it is just 3% of the data

```
# Remove NAs
movies2 <- na.omit(movies2)
```

Part 4: Modeling

In this section, we will start to model the problem using linear regression.

We know that the best correlated variables are *audience_score, critics_score*. We also know that *genre* makes difference in the ratings, thus I will begin with those variables and build from them.

```
# Initial model
model1 <- lm(imdb_rating ~ audience_score + critics_score + genre, data=movies2)

summary(model1)
```

```
## 
## Call:
## lm(formula = imdb_rating ~ audience_score + critics_score + genre, 
##     data = movies2)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.41382 -0.21245  0.04164  0.28329  1.16637 
## 
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                    3.6964115  0.0819704  45.094  < 2e-16 ***
## audience_score                 0.0343525  0.0013458  25.525  < 2e-16 ***
## critics_score                  0.0105927  0.0009559  11.081  < 2e-16 ***
## genreAnimation                -0.4735190  0.1692520  -2.798  0.00530 ** 
## genreArt House & International  0.2252294  0.1450609   1.553  0.12101    
## genreComedy                   -0.1887325  0.0784810  -2.405  0.01647 *  
## genreDocumentary               0.1851475  0.0967460   1.914  0.05611 .  
## genreDrama                     0.0726158  0.0674299   1.077  0.28193    
## genreHorror                    0.0246026  0.1161995   0.212  0.83239    
## genreMusical & Performing Arts  0.0375621  0.1522032   0.247  0.80515    
## genreMystery & Suspense        0.2793802  0.0866169   3.225  0.00132 ** 
## genreOther                    -0.0432402  0.1338846  -0.323  0.74683    
## genreScience Fiction & Fantasy -0.0891595  0.1784214  -0.500  0.61745    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4743 on 629 degrees of freedom
## Multiple R-squared:  0.8088, Adjusted R-squared:  0.8051 
## F-statistic: 221.7 on 12 and 629 DF,  p-value: < 2.2e-16
```

That is a good start. The R^2 was 81% from start. Audience and critics scores are very important for the model and help to explain the variance. Notice that the genre brings some significant variables and some that are not.

Let’s use the forward technique and add some new variables, aiming to increase our R^2-Adjusted. Some other variables to be used are using the correlation criterium. The stronger, the better, so *imdb_num_votes, runtime, thtr_rel_month and thtr_rel_year*

```
# Added imdb_num_votes to the model
model2 <- lm(imdb_rating ~ audience_score + critics_score + genre + imdb_num_votes, data=movies2)
summary(model2)
```

```
## 
## Call:
## lm(formula = imdb_rating ~ audience_score + critics_score + genre + 
##     imdb_num_votes, data = movies2)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.48200 -0.17968  0.03647  0.26330  1.08675 
## 
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                    3.736e+00  8.027e-02  46.545  < 2e-16 ***
## audience_score                 3.225e-02  1.363e-03  23.654  < 2e-16 ***
## critics_score                  1.034e-02  9.336e-04  11.079  < 2e-16 ***
## genreAnimation                -4.277e-01  1.653e-01  -2.587  0.00990 ** 
## genreArt House & International  3.293e-01  1.427e-01   2.308  0.02133 *  
## genreComedy                   -1.535e-01  7.681e-02  -1.998  0.04612 *  
## genreDocumentary               3.353e-01  9.795e-02   3.423  0.00066 ***
## genreDrama                     1.215e-01  6.633e-02   1.832  0.06745 .  
## genreHorror                    6.455e-02  1.136e-01   0.568  0.57000    
## genreMusical & Performing Arts  1.598e-01  1.500e-01   1.065  0.28723    
## genreMystery & Suspense        2.879e-01  8.452e-02   3.406  0.00070 ***
## genreOther                    -5.165e-02  1.306e-01  -0.395  0.69270    
## genreScience Fiction & Fantasy -9.707e-02  1.741e-01  -0.558  0.57728    
## imdb_num_votes                 1.027e-06  1.791e-07   5.733 1.53e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4627 on 628 degrees of freedom
## Multiple R-squared:  0.8183, Adjusted R-squared:  0.8145 
## F-statistic: 217.5 on 13 and 628 DF,  p-value: < 2.2e-16
```

imdb_num_votes proved to be a good variable, increasing our R^2-Adj in 1%.

```
# Adding runtime
model3 <- lm(imdb_rating ~ audience_score + critics_score + genre + imdb_num_votes +
               runtime, data=movies2)
summary(model3)
```

```
## 
## Call:
## lm(formula = imdb_rating ~ audience_score + critics_score + genre + 
##     imdb_num_votes + runtime, data = movies2)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.41680 -0.19091  0.03382  0.25756  1.10483 
## 
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                     3.352e+00  1.296e-01  25.860  < 2e-16 ***
## audience_score                  3.216e-02  1.350e-03  23.828  < 2e-16 ***
## critics_score                   1.021e-02  9.247e-04  11.041  < 2e-16 ***
## genreAnimation                 -3.642e-01  1.645e-01  -2.214 0.027206 *  
## genreArt House & International  3.219e-01  1.412e-01   2.279 0.022994 *  
## genreComedy                    -1.330e-01  7.622e-02  -1.744 0.081562 .  
## genreDocumentary                3.600e-01  9.717e-02   3.705 0.000230 ***
## genreDrama                      9.465e-02  6.605e-02   1.433 0.152320    
## genreHorror                     1.003e-01  1.128e-01   0.889 0.374326    
## genreMusical & Performing Arts  1.173e-01  1.489e-01   0.787 0.431318    
## genreMystery & Suspense         2.658e-01  8.386e-02   3.170 0.001600 ** 
## genreOther                     -6.713e-02  1.294e-01  -0.519 0.603952    
## genreScience Fiction & Fantasy -8.300e-02  1.723e-01  -0.482 0.630225    
## imdb_num_votes                  8.229e-07  1.854e-07   4.439 1.07e-05 ***
## runtime                         3.948e-03  1.053e-03   3.749 0.000194 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.458 on 627 degrees of freedom
## Multiple R-squared:  0.8223, Adjusted R-squared:  0.8183 
## F-statistic: 207.2 on 14 and 627 DF,  p-value: < 2.2e-16
```

The addition of *runtime* was almost not percepted. That one won’t be kept.

```
# Adding thtr_rel_month
model4 <- lm(imdb_rating ~ audience_score + critics_score + genre + imdb_num_votes+
               thtr_rel_month, data=movies2)
summary(model4)
```

```
## 
## Call:
## lm(formula = imdb_rating ~ audience_score + critics_score + genre + 
##     imdb_num_votes + thtr_rel_month, data = movies2)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2.41462 -0.18380  0.03488  0.26683  1.07437 
## 
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                     3.659e+00  8.659e-02  42.259  < 2e-16 ***
## audience_score                  3.229e-02  1.359e-03  23.767  < 2e-16 ***
## critics_score                   1.029e-02  9.306e-04  11.060  < 2e-16 ***
## genreAnimation                 -4.366e-01  1.648e-01  -2.649 0.008271 ** 
## genreArt House & International  3.241e-01  1.422e-01   2.279 0.022983 *  
## genreComedy                    -1.581e-01  7.657e-02  -2.065 0.039371 *  
## genreDocumentary                3.366e-01  9.761e-02   3.448 0.000603 ***
## genreDrama                      1.189e-01  6.612e-02   1.798 0.072674 .  
## genreHorror                     6.558e-02  1.132e-01   0.579 0.562546    
## genreMusical & Performing Arts  1.470e-01  1.496e-01   0.983 0.326028    
## genreMystery & Suspense         2.935e-01  8.426e-02   3.484 0.000529 ***
## genreOther                     -4.117e-02  1.303e-01  -0.316 0.752026    
## genreScience Fiction & Fantasy -8.621e-02  1.735e-01  -0.497 0.619520    
## imdb_num_votes                  9.828e-07  1.795e-07   5.477 6.28e-08 ***
## thtr_rel_month                  1.196e-02  5.170e-03   2.314 0.020987 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4611 on 627 degrees of freedom
## Multiple R-squared:  0.8198, Adjusted R-squared:  0.8158 
## F-statistic: 203.8 on 14 and 627 DF,  p-value: < 2.2e-16
```

The variable *thtr_rel_month* is also not very meaningful. It will just make the model more complex without really adding value.

```
# Adding thtr_rel_year
model5 <- lm(imdb_rating ~ audience_score + critics_score + genre +
              imdb_num_votes+ thtr_rel_year, data=movies2)
summary(model5)
```

```
##
## Call:
## lm(formula = imdb_rating ~ audience_score + critics_score + genre +
##     imdb_num_votes + thtr_rel_year, data = movies2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.47926 -0.18402  0.03632  0.26347  1.09182
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     4.788e+00  3.599e+00   1.330 0.183883
## audience_score                  3.221e-02  1.372e-03  23.478  < 2e-16 ***
## critics_score                   1.031e-02  9.398e-04  10.975  < 2e-16 ***
## genreAnimation                 -4.236e-01  1.660e-01  -2.552 0.010960 *
## genreArt House & International  3.324e-01  1.432e-01   2.322 0.020573 *
## genreComedy                    -1.524e-01  7.696e-02  -1.980 0.048110 *
## genreDocumentary                3.422e-01  1.009e-01   3.393 0.000736 ***
## genreDrama                      1.230e-01  6.657e-02   1.847 0.065199 .
## genreHorror                     6.398e-02  1.137e-01   0.563 0.573776
## genreMusical & Performing Arts  1.632e-01  1.506e-01   1.084 0.278753
## genreMystery & Suspense         2.890e-01  8.466e-02   3.414 0.000682 ***
## genreOther                     -5.453e-02  1.311e-01  -0.416 0.677592
## genreScience Fiction & Fantasy -9.932e-02  1.744e-01  -0.570 0.569141
## imdb_num_votes                  1.041e-06  1.857e-07   5.606 3.11e-08 ***
## thtr_rel_year                  -5.255e-04  1.797e-03  -0.292 0.770106
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4631 on 627 degrees of freedom
## Multiple R-squared:  0.8183, Adjusted R-squared:  0.8142
## F-statistic: 201.7 on 14 and 627 DF,  p-value: < 2.2e-16
```

Once again, the same is valid. The variable *thtr_rel_year* is not relevant to the model.

Now I will add the binary variables *best_dir_win, best_pic_nom, best_pic_win and top200_box*. All of those showed influence over the ratings during ANOVA test.

```
# Adding best director, actor, actress, picture
model6 <- lm(imdb_rating ~ audience_score + critics_score + genre + imdb_num_votes+
              best_dir_win + best_pic_nom + best_pic_win + top200_box, data=movies2)
summary(model6)
```

```
##
## Call:
## lm(formula = imdb_rating ~ audience_score + critics_score + genre +
##     imdb_num_votes + best_dir_win + best_pic_nom + best_pic_win +
##     top200_box, data = movies2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4871 -0.1792  0.0357  0.2612  1.0881
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    3.738e+00  8.122e-02  46.026  < 2e-16 ***
## audience_score                 3.224e-02  1.369e-03  23.546  < 2e-16 ***
## critics_score                  1.030e-02  9.427e-04  10.927  < 2e-16 ***
## genreAnimation                -4.292e-01  1.659e-01  -2.588 0.009885 **
## genreArt House & International  3.298e-01  1.432e-01   2.304 0.021573 *
## genreComedy                   -1.571e-01  7.736e-02  -2.030 0.042747 *
## genreDocumentary               3.375e-01  9.884e-02   3.414 0.000681 ***
## genreDrama                     1.161e-01  6.705e-02   1.732 0.083759 .
## genreHorror                    5.943e-02  1.140e-01   0.521 0.602474
## genreMusical & Performing Arts  1.524e-01  1.505e-01   1.013 0.311545
## genreMystery & Suspense        2.792e-01  8.512e-02   3.280 0.001097 **
## genreOther                    -5.382e-02  1.314e-01  -0.409 0.682334
## genreScience Fiction & Fantasy -9.783e-02  1.744e-01  -0.561 0.574977
## imdb_num_votes                 1.066e-06  1.955e-07   5.455 7.09e-08 ***
## best_dir_winyes                1.003e-01  7.874e-02   1.274 0.203086
## best_pic_nomyes               -1.678e-03  1.192e-01  -0.014 0.988774
## best_pic_winyes               -1.036e-01  2.137e-01  -0.485 0.627943
## top200_boxyes                 -1.175e-01  1.280e-01  -0.919 0.358679
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4633 on 624 degrees of freedom
## Multiple R-squared:  0.819,  Adjusted R-squared:  0.8141
## F-statistic: 166.1 on 17 and 624 DF,  p-value: < 2.2e-16
```

The addition of those variables did not increase significantly the R². So, they should be kept out of the model.

Let’s see the variable prizes.

```
# Adding prizes
model7 <- lm(imdb_rating ~ audience_score + critics_score + genre +
               imdb_num_votes+ prizes, data=movies2)
summary(model7)
```

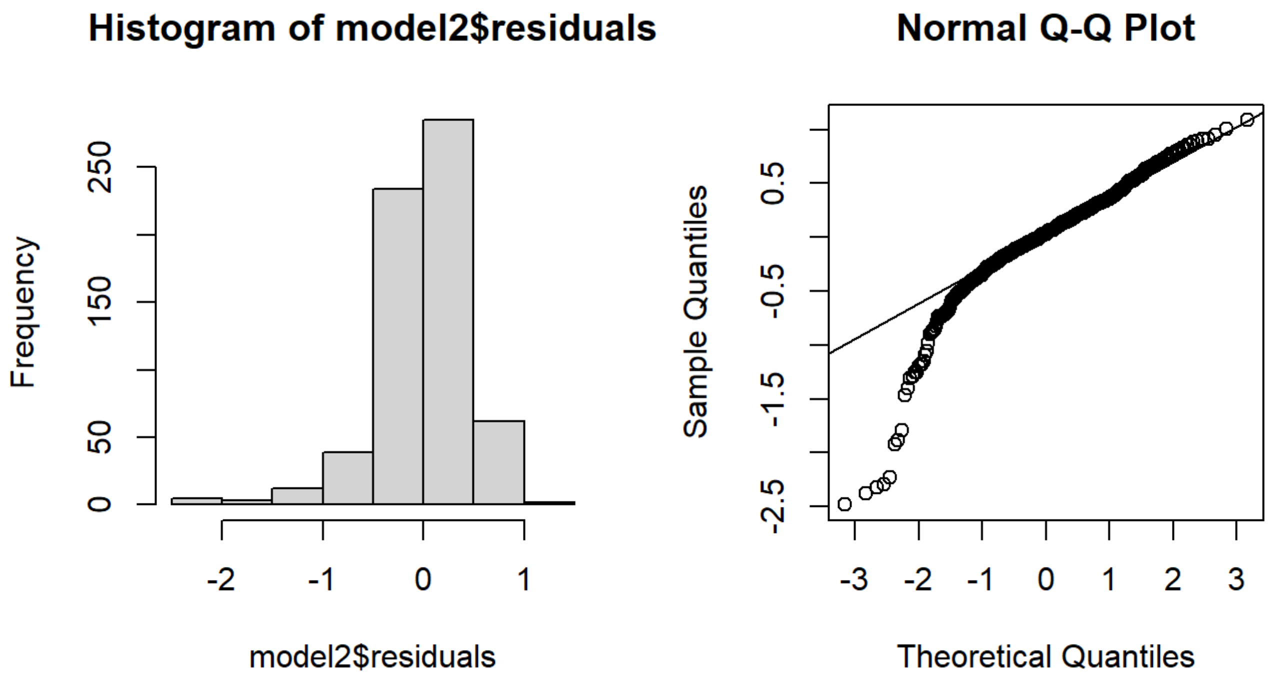
```
##
## Call:
## lm(formula = imdb_rating ~ audience_score + critics_score + genre +
##     imdb_num_votes + prizes, data = movies2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.47469 -0.18431  0.03972  0.26432  1.11468
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    3.732e+00  8.009e-02  46.595  < 2e-16 ***
## audience_score                 3.237e-02  1.361e-03  23.782  < 2e-16 ***
## critics_score                  1.016e-02  9.355e-04  10.860  < 2e-16 ***
## genreAnimation                -4.321e-01  1.649e-01  -2.620 0.009003 **
## genreArt House & International  3.309e-01  1.423e-01   2.325 0.020383 *
## genreComedy                   -1.612e-01  7.671e-02  -2.101 0.036012 *
## genreDocumentary               3.417e-01  9.775e-02   3.495 0.000507 ***
## genreDrama                     1.056e-01  6.662e-02   1.585 0.113508
## genreHorror                    7.038e-02  1.133e-01   0.621 0.534755
## genreMusical & Performing Arts  1.594e-01  1.496e-01   1.065 0.287254
## genreMystery & Suspense        2.654e-01  8.501e-02   3.122 0.001881 **
## genreOther                    -5.898e-02  1.303e-01  -0.453 0.651056
## genreScience Fiction & Fantasy -9.062e-02  1.737e-01  -0.522 0.601956
## imdb_num_votes                 9.553e-07  1.819e-07   5.250 2.08e-07 ***
## prizes                         6.395e-02  3.109e-02   2.057 0.040141 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4616 on 627 degrees of freedom
## Multiple R-squared:  0.8195, Adjusted R-squared:  0.8155
## F-statistic: 203.3 on 14 and 627 DF,  p-value: < 2.2e-16
```

Not much added as well.

Given that the model 2 is the most simple and keeps the same level of explanatory power, I will keep the model2 at ~81% of R²-Adjusted.

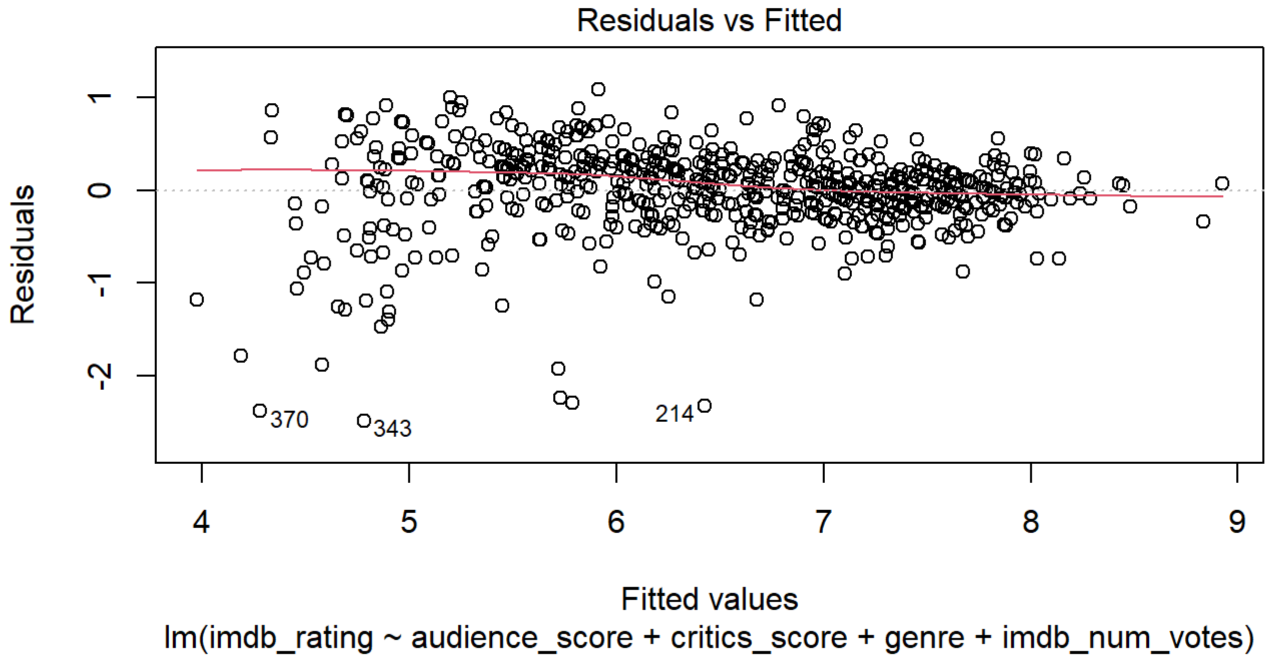
Let’s now look at the residuals and assess the final model.

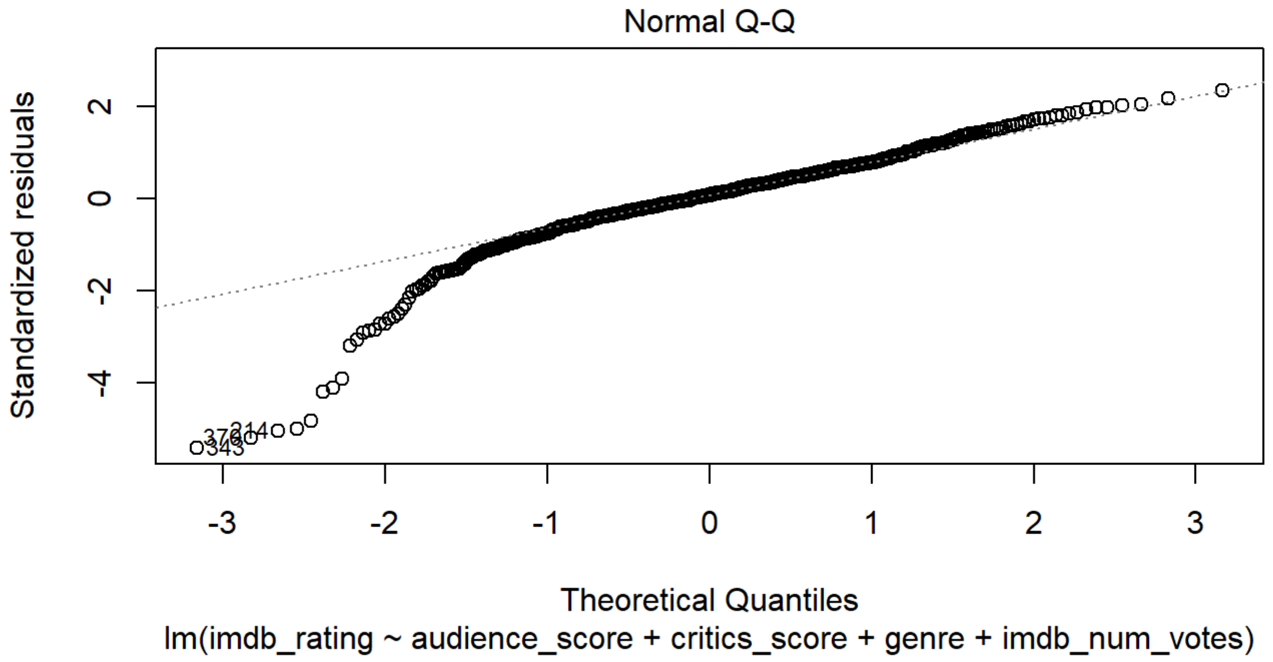
```
# Setup 2 graphics in one row
par(mfrow=c(1,2))

# Histogram of the residuals
hist(model2$residuals)

# qqplot
qqnorm(model2$residuals)
qqline(model2$residuals)
```

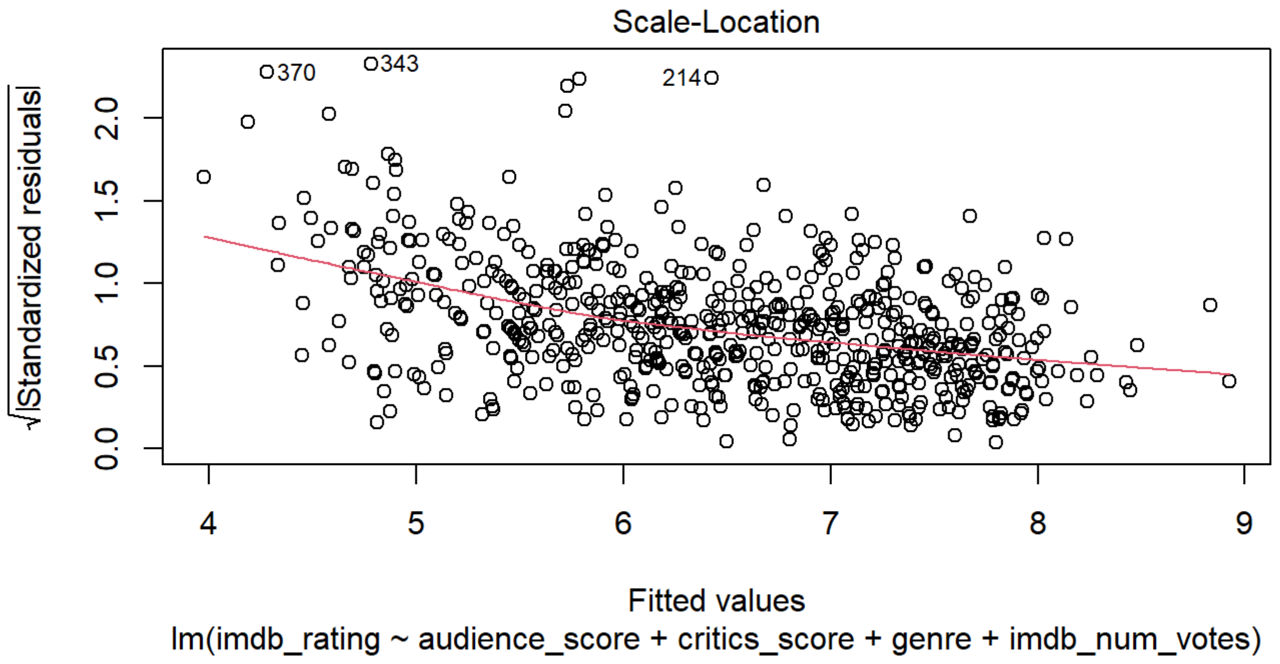


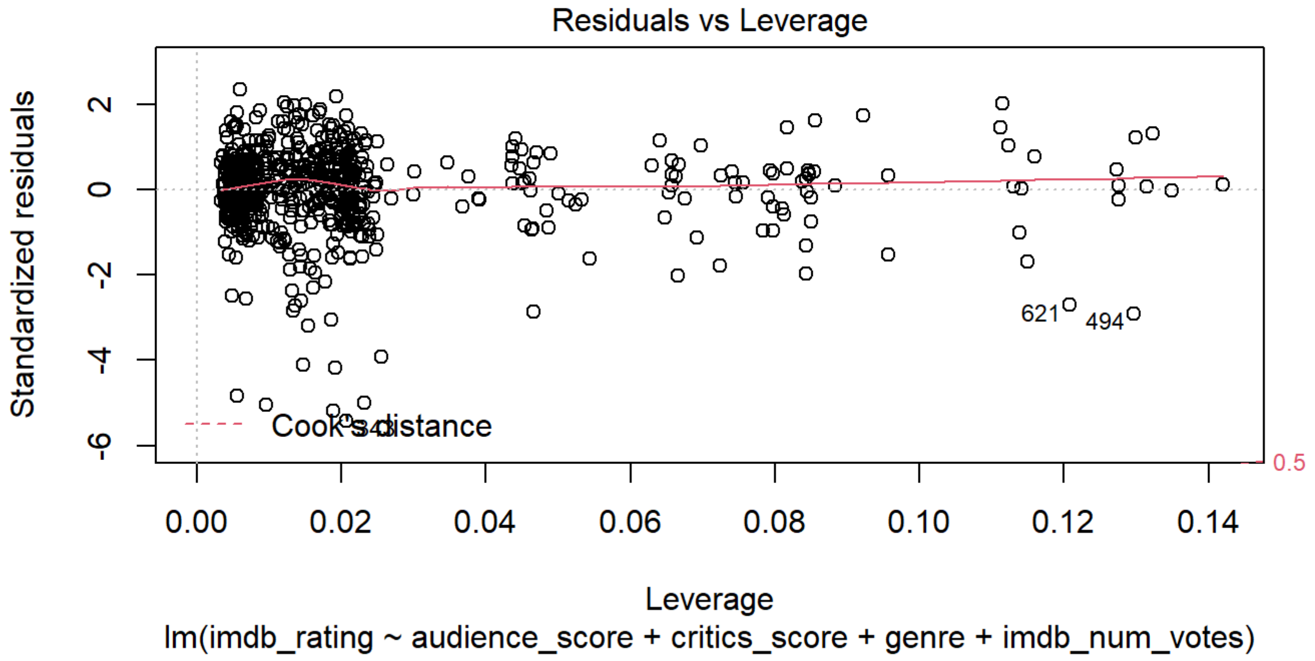
That’s approximately normal. There is some skewness on the left side, but we are ok with it.

```
plot(model2)
```


Residuals vs Fitted
Residuals
1
0
-1
-2
370
343
214
4
5
6
7
8
9
Fitted values
lm(imdb_rating ~ audience_score + critics_score + genre + imdb_num_votes)


Normal Q-Q
Standardized residuals
2
0
-2
-4
370
214
343
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(imdb_rating ~ audience_score + critics_score + genre + imdb_num_votes)


Scale-Location
√|Standardized residuals|
2.0
1.5
1.0
0.5
0.0
370
343
214
4
5
6
7
8
9
Fitted values
lm(imdb_rating ~ audience_score + critics_score + genre + imdb_num_votes)



The model is fairly good, it shows some skewness in the residuals, but it is still a good model. The residuals don't show any pattern, which is good.

Part 5: Prediction

Train Test Split

```
#Index to create a random sample for training
train_index <- sample(1:nrow(movies2), size= 0.8*nrow(movies2) )

# Train and test datasets
train <- movies2[train_index,]
test <- movies2[-train_index,]

print( paste('train:', dim(train)) )
```

```
## [1] "train: 513" "train: 24"
```

```
print( paste('test:', dim(test)) )
```

```
## [1] "test: 129" "test: 24"
```

```
# Model Traning
linear_model <- lm(imdb_rating ~ audience_score + critics_score + genre + imdb_num_votes,
                   data= train)

summary(linear_model)
```

```
##
## Call:
## lm(formula = imdb_rating ~ audience_score + critics_score + genre +
##     imdb_num_votes, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.44955 -0.18187  0.03919  0.24502  1.07898
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     3.689e+00  8.699e-02  42.407  < 2e-16 ***
## audience_score                  3.257e-02  1.484e-03  21.951  < 2e-16 ***
## critics_score                   9.892e-03  1.029e-03   9.617  < 2e-16 ***
## genreAnimation                 -4.295e-01  1.957e-01  -2.195 0.028635 *
## genreArt House & International  3.798e-01  1.429e-01   2.658 0.008122 **
## genreComedy                    -3.694e-02  8.483e-02  -0.435 0.663445
## genreDocumentary                3.722e-01  1.077e-01   3.457 0.000593 ***
## genreDrama                      1.857e-01  7.369e-02   2.519 0.012070 *
## genreHorror                     1.362e-01  1.188e-01   1.147 0.252043
## genreMusical & Performing Arts  2.009e-01  1.620e-01   1.240 0.215450
## genreMystery & Suspense         3.647e-01  9.237e-02   3.948 9.02e-05 ***
## genreOther                      3.467e-02  1.428e-01   0.243 0.808249
## genreScience Fiction & Fantasy -3.438e-02  1.727e-01  -0.199 0.842251
## imdb_num_votes                  9.379e-07  1.968e-07   4.766 2.47e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4513 on 499 degrees of freedom
## Multiple R-squared:  0.8264, Adjusted R-squared:  0.8218
## F-statistic: 182.7 on 13 and 499 DF,  p-value: < 2.2e-16
```

```
# Predictions
y_hat <- predict(linear_model, test)

performance = data.frame(title = test$title,
                         rating= test$imdb_rating,
                         prediction = y_hat,
                         y_error = test$imdb_rating - y_hat)

# RMSE
sqrt( mean(performance$y_error^2) )
```

```
## [1] 0.5105454
```

```
# MAE
mae(performance$rating, performance$prediction)
```

```
## [1] 0.3431509
```

```
# Get random predictions
preds_sample <- sample(1:nrow(performance), 15)

# View some predictions
performance[preds_sample,]
```

```
##                                       title rating prediction      y_error
## 45                        Strictly Business    5.3   5.798861 -0.498861455
## 126                       Cocoon: The Return    5.2   5.381772 -0.181772205
## 124                 In the Name of the Father    8.1   7.995831  0.104168709
## 82        Russian Dolls (Les Poupees Russes)    7.0   6.991796  0.008203717
## 85                                Guess Who    5.9   5.704479  0.195521089
## 109                                 Flipped    7.7   7.012744  0.687256499
## 33                         A Boy and His Dog    6.6   6.680098 -0.080097773
## 67                             Aspen Extreme    5.8   6.244208 -0.444207889
## 38                               Topsy-Turvy    7.4   7.337531  0.062468694
## 39                        Night and the City    5.8   5.550661  0.249338845
## 111                              Snake Eyes    5.9   5.469285  0.430715379
## 30                                 Godzilla    6.5   7.207295 -0.707295365
## 8                          Driving Miss Daisy    7.4   7.379306  0.020693843
## 57                             Run Lola Run    7.8   7.874034 -0.074033623
## 94  Dumb and Dumberer: When Harry Met Lloyd    3.4   4.528407 -1.128407311
```

Part 6: Conclusion

Movie ratings are something subjective. What one likes may not be the same as what other like. But having a lot of ratings from many people can show us some patterns that can be explained by the variables we have.

In a Linear Regression, the objective is to explain the variability of the response variable - IMDB Ratings in this case - using other variables that are related to it. Using some statistical tests such as correlation (that measures the strength of a linear relationship between two variables) and ANOVA (that can test if the means of different groups are statistically equal or not), we are able to drive the choice of the best features to compose the model.

From a dataset with 32 variables, we got to a final model with 4 variables, explaining more than 81% of the *imdb_ratings* variance. The model presented - on the test set - a 0.5 points error on average for each prediction and 0.36 of Mean Absolute error.