Modeling and prediction for movies

Setup

Load packages

```
library(ggplot2)
library(dplyr)
library(statsr)
library(corrplot)
```

Load data

Load the dataset movies.Rdata.

```
load("movies.Rdata")
```

Part 1: Data

- The data set is comprised of 651 randomly sampled movies produced and released before 2016. We can consider it is generalizable.
- Since this dataset is not an experiment with random assignment, there is no causality invovled.

Part 2: Research question

What attributes make a movie popular? And what's new in the dataset we have collected.

This is an interesting question as most of us watching several kinds of movies, but we don't want to waste 2 hours for a bad one. Before we decide which movie should we watch, it's better to check others' comments, is it good for most of people? So, what are the attributes that can make a movie popular?

This is also an interesting question behind the recommended system which requires machine learning algorithm. It's cutting edge technology that we are all intereted in.

How to define if the movie is popular? We will look into **Rotten Tomatoes** or **IMDB** rating/score.

Part 3: Exploratory data analysis

First of all, I would like to find some relationships between those variables. Some of the variations will be omitted. Some of them are dependent. Let's use some plots and analysis to find out. After that, we can decide which variables are needed in our model.

1. Take a look at the variables:

```
names(movies)
```

```
##  [1] "title"           "title_type"       "genre"
##  [4] "runtime"         "mpaa_rating"      "studio"
##  [7] "thtr_rel_year"   "thtr_rel_month"   "thtr_rel_day"
## [10] "dvd_rel_year"    "dvd_rel_month"    "dvd_rel_day"
## [13] "imdb_rating"     "imdb_num_votes"   "critics_rating"
## [16] "critics_score"   "audience_rating"  "audience_score"
## [19] "best_pic_nom"    "best_pic_win"     "best_actor_win"
## [22] "best_actress_win" "best_dir_win"    "top200_box"
## [25] "director"        "actor1"           "actor2"
## [28] "actor3"          "actor4"           "actor5"
## [31] "imdb_url"        "rt_url"
```

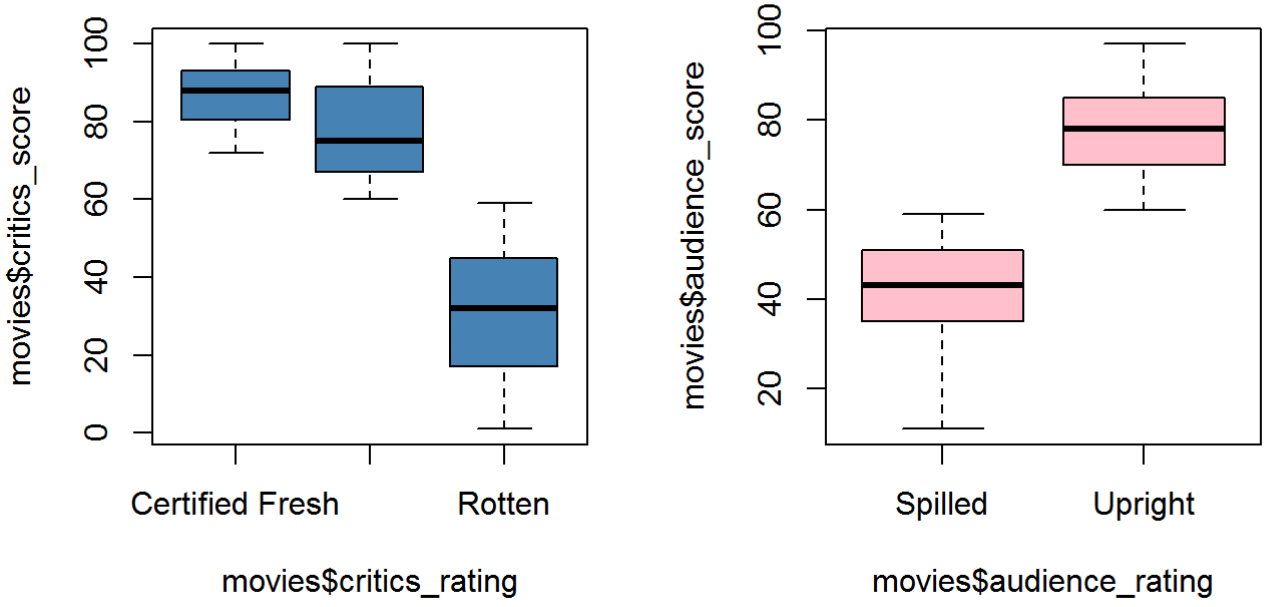
It seems that some of them are not necesary for the question:

- title, runtime, they should not affect if a movie is popular.
- dvd release year, month and day, seems not interested variations.
- for actor1 through 5, imdburl and rturl will not be discussed in the project.

All other variables are numeric or tranfered to numeric.

For those who are not familiar with Rotten tomatoes rating, it depends on its score:

```
par(mfrow = c(1,2))
plot(movies$critics_score~movies$critics_rating, col = 'steelblue')
plot(movies$audience_score~movies$audience_rating, col = 'pink')
```



For critics, certified fresh is 75 and above, Fresh is 60 and above, Rotten is lower than 60. For audience, upright is 60 and above, spilled is 60 and below.

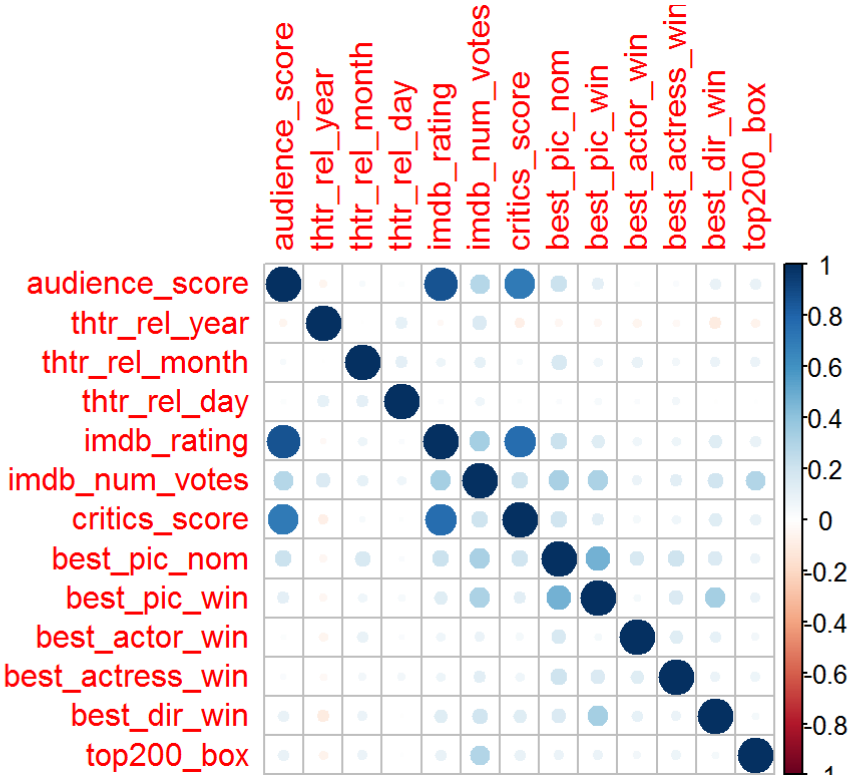
To simplify, I will only use the two scores variables. Since the question is regarding 'popular', I will use **'audience_score'** as outcome.

Here is the code to get necessary variables.

```
mov_data_factor <- select(movies,
                  best_pic_nom, best_pic_win,
                  best_actor_win, best_actress_win,
                  best_dir_win, top200_box) %>%
                  mutate_if(is.factor, as.numeric)
```

```
mov_data_num <- select(movies,
                       audience_score,
                       thtr_rel_year, thtr_rel_month, thtr_rel_day,
                       imdb_rating, imdb_num_votes,
                       critics_score) %>%
                       cbind(mov_data_factor)
```

Let's plot the correlation between the variables:



From the plot, we can see that several factors have correlation with 'popular':

- imdb_rating
- critics_score
- imdb_numVotes

And some of them have weak correlation:

- bestpic_nom
- bestpic_win

- bestdir_win
- top200

Part 4: Modeling

Based on the above plot, I will include the following 7 variables to the model, since they are more likely to be the reason that a movie is popular:

- imdb_rating
- critics_score
- imdb_numVotes
- bestpic_nom
- bestpic_win
- bestdir_win
- top200

This is the same reason that we exclude other variables based on the plot.

Backward elimination will be used with adjusted R^2 so that we can get more reliable prediction.

Step 1

```
#lm all
summary(lm(audience_score~imdb_rating + imdb_num_votes + critics_score + best_pic_nom +
 best_pic_win + best_dir_win + top200_box, mov_data_num))$adj.r.squared
```

```
## [1] 0.7512168
```

```
# - top200_box
summary(lm(audience_score~imdb_rating + imdb_num_votes + critics_score + best_pic_nom +
 best_pic_win + best_dir_win, mov_data_num))$adj.r.squared
```

```
## [1] 0.751533
```

```
# - best_dir_win
summary(lm(audience_score~imdb_rating + imdb_num_votes + critics_score + best_pic_nom +
 best_pic_win + top200_box, mov_data_num))$adj.r.squared
```

```
## [1] 0.7509296
```

```
# - best_pic_win
summary(lm(audience_score~imdb_rating + imdb_num_votes + critics_score + best_pic_nom +
 best_dir_win + top200_box, mov_data_num))$adj.r.squared
```

```
## [1] 0.7515261
```

```
# - best_pic_nom
summary(lm(audience_score~imdb_rating + imdb_num_votes + critics_score + best_pic_win +
 best_dir_win + top200_box, mov_data_num))$adj.r.squared
```

```
## [1] 0.7509734
```

```
# - critics_score
summary(lm(audience_score~imdb_rating + imdb_num_votes + best_pic_nom + best_pic_win +
best_dir_win + top200_box, mov_data_num))$adj.r.squared
```

```
## [1] 0.747215
```

```
# - imdb_num_votes
summary(lm(audience_score~imdb_rating + critics_score + best_pic_nom + best_pic_win + b
est_dir_win + top200_box, mov_data_num))$adj.r.squared
```

```
## [1] 0.7515791
```

```
# - imdb_rating
summary(lm(audience_score~imdb_num_votes + critics_score + best_pic_nom + best_pic_win
 + best_dir_win + top200_box, mov_data_num))$adj.r.squared
```

```
## [1] 0.5156582
```

- imdb_num_votes lm(audience_score~imdb_rating + critics_score + best_pic_nom + best_pic_win + best_dir_win + top200_box, mov_data_num)

Step 2

```
# - top200_box
summary(lm(audience_score~imdb_rating + critics_score + best_pic_nom + best_pic_win + b
est_dir_win, mov_data_num))$adj.r.squared
```

```
## [1] 0.7518628
```

```
# - best_dir_win
summary(lm(audience_score~imdb_rating + critics_score + best_pic_nom + best_pic_win + t
op200_box, mov_data_num))$adj.r.squared
```

```
## [1] 0.7513076
```

```
# - best_pic_win
summary(lm(audience_score~imdb_rating + critics_score + best_pic_nom + best_dir_win + t
op200_box, mov_data_num))$adj.r.squared
```

```
## [1] 0.7518988
```

```
# - best_pic_nom
summary(lm(audience_score~imdb_rating + critics_score  + best_pic_win + best_dir_win +
top200_box, mov_data_num))$adj.r.squared
```

```
## [1] 0.7512726
```

```
# - critics_score
summary(lm(audience_score~imdb_rating + best_pic_nom + best_pic_win + best_dir_win + to
p200_box, mov_data_num))$adj.r.squared
```

```
## [1] 0.7476008
```

```
# - imdb_rating
summary(lm(audience_score~ critics_score + best_pic_nom + best_pic_win + best_dir_win +
 top200_box, mov_data_num))$adj.r.squared
```

```
## [1] 0.4984289
```

- best_pic_win summary(lm(audience_score~imdb_rating + critics_score + best_pic_nom + best_dir_win + top200_box, mov_data_num))$adj.r.squared

Step 3

```
# - top200_box
summary(lm(audience_score~imdb_rating + critics_score + best_pic_nom + best_dir_win, mo
v_data_num))$adj.r.squared
```

```
## [1] 0.7521881
```

```
# - best_dir_win
summary(lm(audience_score~imdb_rating + critics_score + best_pic_nom + top200_box, mov_
data_num))$adj.r.squared
```

```
## [1] 0.7514255
```

```
# - best_pic_nom
summary(lm(audience_score~imdb_rating + critics_score + best_dir_win + top200_box, mov_
data_num))$adj.r.squared
```

```
## [1] 0.7516418
```

```
# - critics_score
summary(lm(audience_score~imdb_rating + best_pic_nom + best_dir_win + top200_box, mov_d
ata_num))$adj.r.squared
```

```
## [1] 0.7479213
```

```
# - imdb_rating
summary(lm(audience_score~ critics_score + best_pic_nom + best_dir_win + top200_box, mo
v_data_num))$adj.r.squared
```

```
## [1] 0.4991791
```

- top200_box summary(lm(audience_score~imdb_rating + critics_score + best_pic_nom + best_dir_win, mov_data_num))$adj.r.squared

Step 4

```
# - best_dir_win
summary(lm(audience_score~imdb_rating + critics_score + best_pic_nom, mov_data_num))$ad
j.r.squared
```

```
## [1] 0.7517259
```

```
# - best_pic_nom
summary(lm(audience_score~imdb_rating + critics_score + best_dir_win, mov_data_num))$ad
j.r.squared
```

```
## [1] 0.7518977
```

```
# - critics_score
summary(lm(audience_score~imdb_rating + best_pic_nom + best_dir_win, mov_data_num))$ad
j.r.squared
```

```
## [1] 0.7481622
```

```
# - imdb_rating
summary(lm(audience_score~critics_score + best_pic_nom + best_dir_win, mov_data_num))$a
dj.r.squared
```

```
## [1] 0.4995601
```

- best_pic_nom summary(lm(audience_score~imdb_rating + critics_score + best_dir_win, mov_data_num))$adj.r.squared

```
# - best_dir_win
summary(lm(audience_score~imdb_rating + critics_score, mov_data_num))$adj.r.squared
```

```
## [1] 0.7516082
```

```
# - critics_score
summary(lm(audience_score~imdb_rating + best_dir_win, mov_data_num))$adj.r.squared
```

```
## [1] 0.7477247
```

```
# - imdb_rating
summary(lm(audience_score~critics_score + best_dir_win, mov_data_num))$adj.r.squared
```

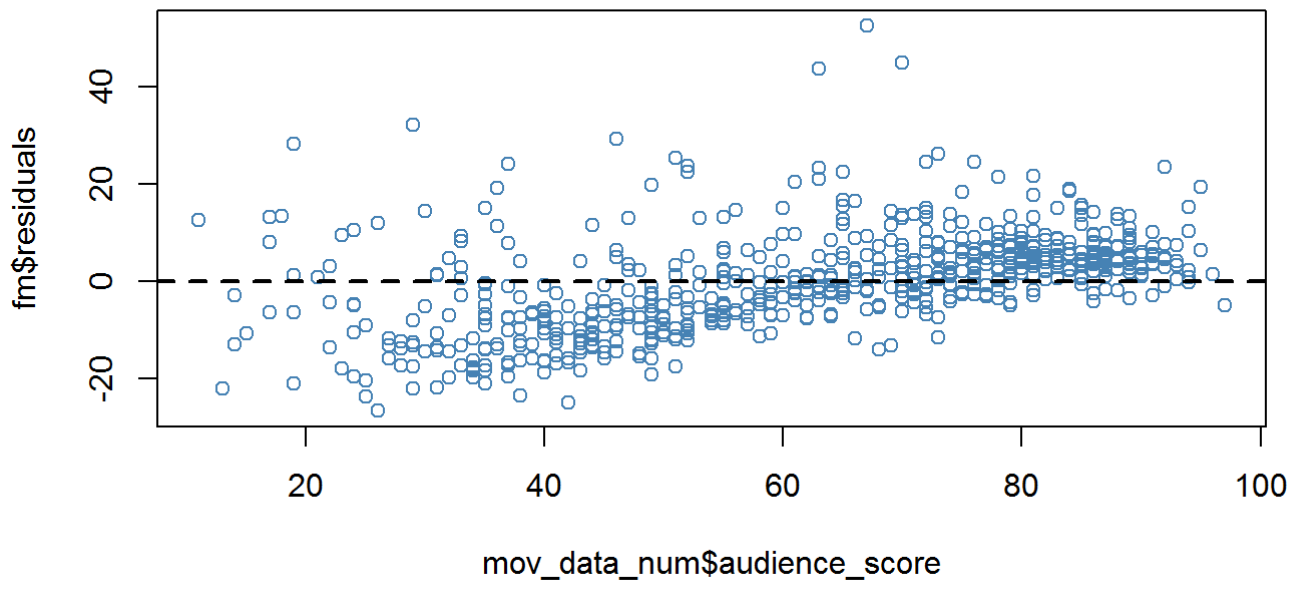
```
## [1] 0.4944501
```

Final model: fm.

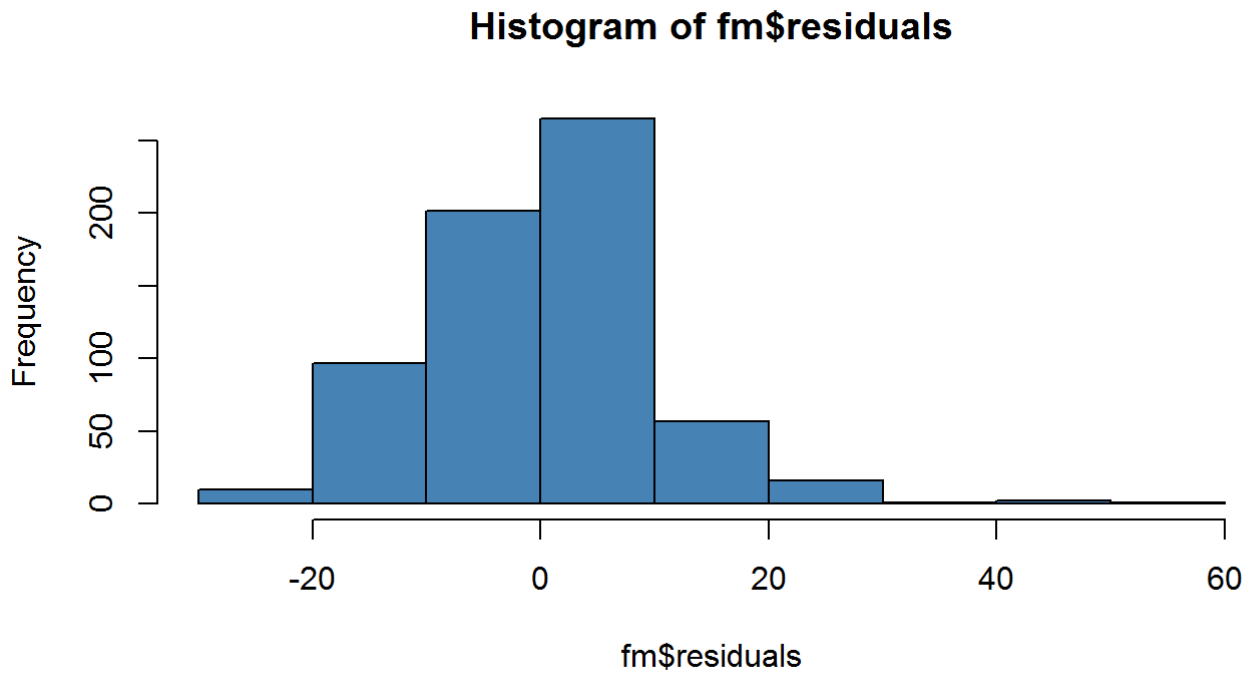
```
fm <- lm(audience_score~imdb_rating + critics_score, mov_data_num)
```

We'll check the residual plot.

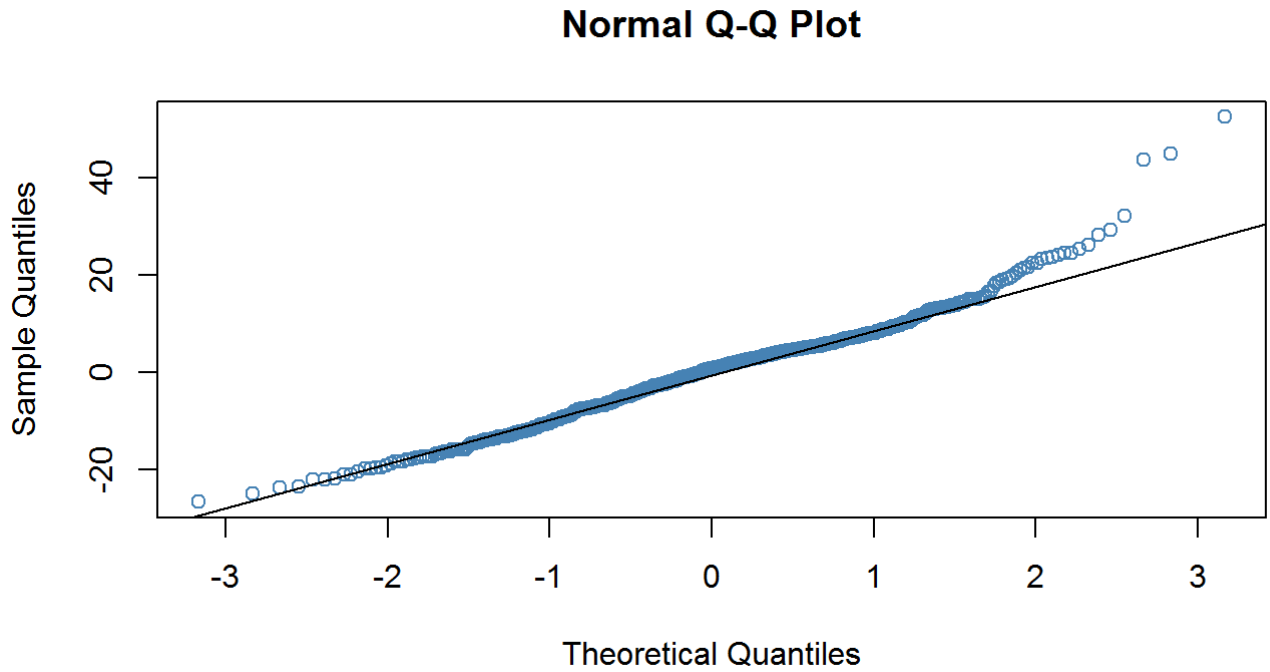
```
par(mfrow = c(1,1))
plot(fm$residuals ~ mov_data_num$audience_score, col = 'steelblue')
abline(h = 0, lwd = 2, lty = 2)
```




```
hist(fm$residuals, col = 'steelblue')
```




```
qqnorm(fm$residuals, col = 'steelblue')
qqline(fm$residuals)
```



We can see that the plot is random and around 0, the histogram is nearly normal distribution.

Model coefficients

```
fm$coefficients
```

```
##   (Intercept)   imdb_rating critics_score
##  -37.03195003   14.65760034    0.07317595
```

All else held constant, for each imdb rating increase the model predicts the movie on audience score will be higher on average by 14.66.

All else held constant, for each critics score on Rotten Tomato the model predicts the movie on audience score will be higher on average by 0.07.

Part 5: Prediction

I will use Doctor Strange in the prediction. critics_score is 90 and imdb rating is 8.

The predicted audience score is

```
-37.03195003 + 14.65760034 * 8 + 0.07317595 * 90
```

```
## [1] 86.81469
```

On Rotten Tomatoes, the score is 90. This is a close prediction.

Part 6: Conclusion

From the research, we have found the two variables that will affect 'popular' - audience score, that is imdb rating and critics score on Rotten tomatoes, it is surprising that best pic, director, win may not affect the popularity from the data and model.

Some shortcomings: we have a lot of variables in the dataset, it can be tedious to do backward elimination. The corrplot can be a way to move some unrelated variables but this is not mentioned in the class and not 100% percent for sure that will work.