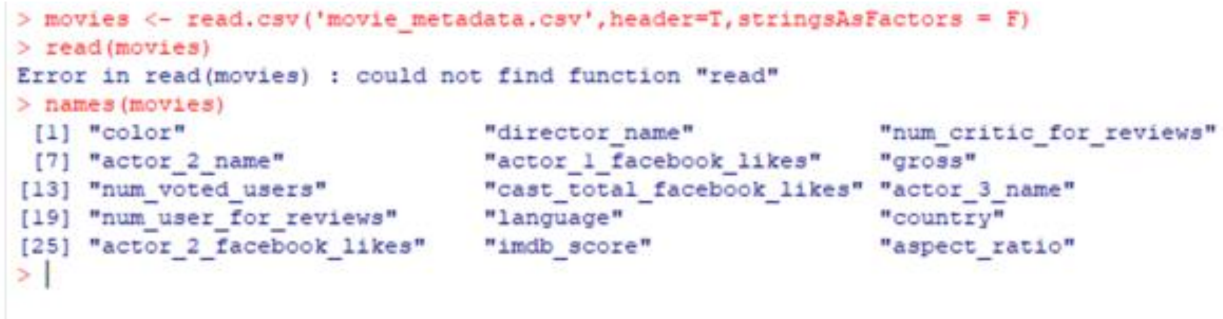
DATA SCIENCE FOR BUSINESS (DATA ANALYTICS)

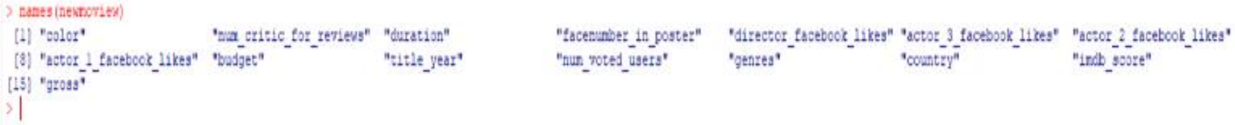
PREDICTION OF MOVIE SUCCESS USING RANDOM FOREST A MACHINE LEARNING ALGORITHMS AND THEIR COMPARISON

VI. APPENDIX



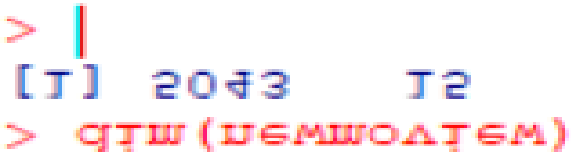
```
> movies <- read.csv('movie_metadata.csv',header=T,stringsAsFactors = F)
> read(movies)
Error in read(movies) : could not find function "read"
> names(movies)
 [1] "color"                    "director_name"             "num_critic_for_reviews"
 [7] "actor_2_name"             "actor_1_facebook_likes"    "gross"
[13] "num_voted_users"          "cast_total_facebook_likes" "actor_3_name"
[19] "num_user_for_reviews"     "language"                  "country"
[25] "actor_2_facebook_likes"   "imdb_score"                "aspect_ratio"
> |
```

Listing all variables from the dataset - Figure 1.



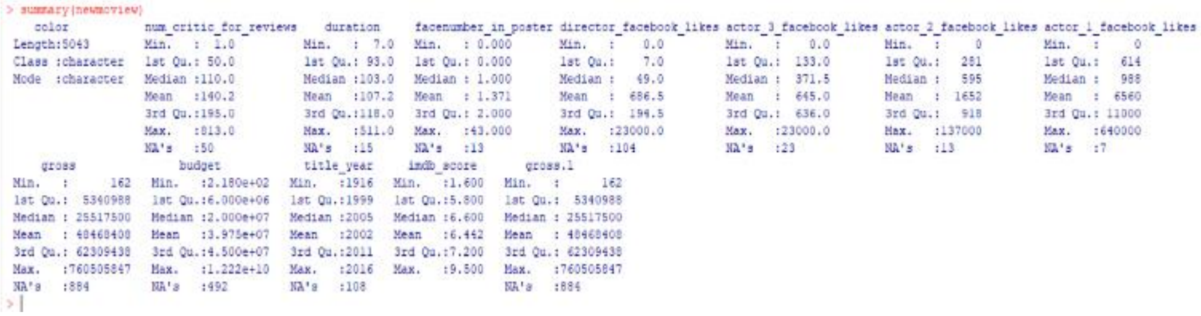
```
> names(newmoview)
 [1] "color"                  "num_critic_for_reviews" "duration"               "facenumber_in_poster"   "director_facebook_likes" "actor_3_facebook_likes" "actor_2_facebook_likes"
 [8] "actor_1_facebook_likes" "budget"                 "title_year"             "num_voted_users"        "genres"                  "country"                "imdb_score"
[15] "gross"
>|
```

New dataset with the attributes - Figure 2.



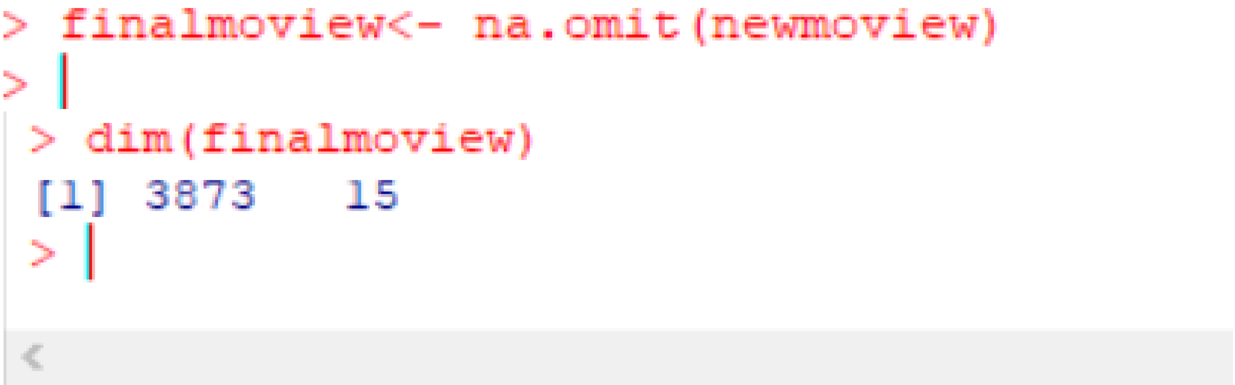
```
> dim(newmoview)
[1] 5043   15
> |
```

Summary of each attribute in the dataset -Figure 3.



```
> summary(newmoview)
    color           num_critic_for_reviews    duration     facenumber_in_poster director_facebook_likes actor_3_facebook_likes actor_2_facebook_likes actor_1_facebook_likes
 Length:5043        Min.   :  1.0          Min.   :  7.0   Min.   : 0.000       Min.   :    0.0         Min.   :    0.0        Min.   :     0         Min.   :     0
 Class :character   1st Qu.: 50.0          1st Qu.: 93.0   1st Qu.: 0.000       1st Qu.:    7.0         1st Qu.:  133.0        1st Qu.:   281         1st Qu.:   614
 Mode  :character   Median :110.0          Median :103.0   Median : 1.000       Median :   49.0         Median :  371.5        Median :   595         Median :   988
                    Mean   :140.2          Mean   :107.2   Mean   : 1.371       Mean   :  686.5         Mean   :  645.0        Mean   :  1652         Mean   :  6560
                    3rd Qu.:195.0          3rd Qu.:118.0   3rd Qu.: 2.000       3rd Qu.:  194.5         3rd Qu.:  636.0        3rd Qu.:   918         3rd Qu.: 11000
                    Max.   :813.0          Max.   :511.0   Max.   :43.000       Max.   :23000.0         Max.   :23000.0        Max.   :137000         Max.   :640000
                    NA's   :50             NA's   :15      NA's   :13           NA's   :104             NA's   :23             NA's   :13             NA's   :7
     gross              budget            title_year     imdb_score        gross.1
 Min.   :      162   Min.   :2.180e+02   Min.   :1916   Min.   :1.600   Min.   :      162
 1st Qu.:  5340988   1st Qu.:6.000e+06   1st Qu.:1999   1st Qu.:5.800   1st Qu.:  5340988
 Median : 25517500   Median :2.000e+07   Median :2005   Median :6.600   Median : 25517500
 Mean   : 48468408   Mean   :3.975e+07   Mean   :2002   Mean   :6.442   Mean   : 48468408
 3rd Qu.: 62309438   3rd Qu.:4.500e+07   3rd Qu.:2011   3rd Qu.:7.200   3rd Qu.: 62309438
 Max.   :760505847   Max.   :1.222e+10   Max.   :2016   Max.   :9.500   Max.   :760505847
 NA's   :884         NA's   :492         NA's   :108                    NA's   :884
> |
```

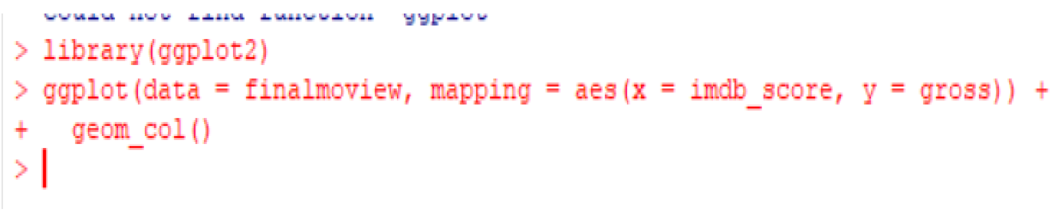
Various dimension of the dataset - Figure 4



```
> finalmovie<- na.omit(newmovie)
> |
```

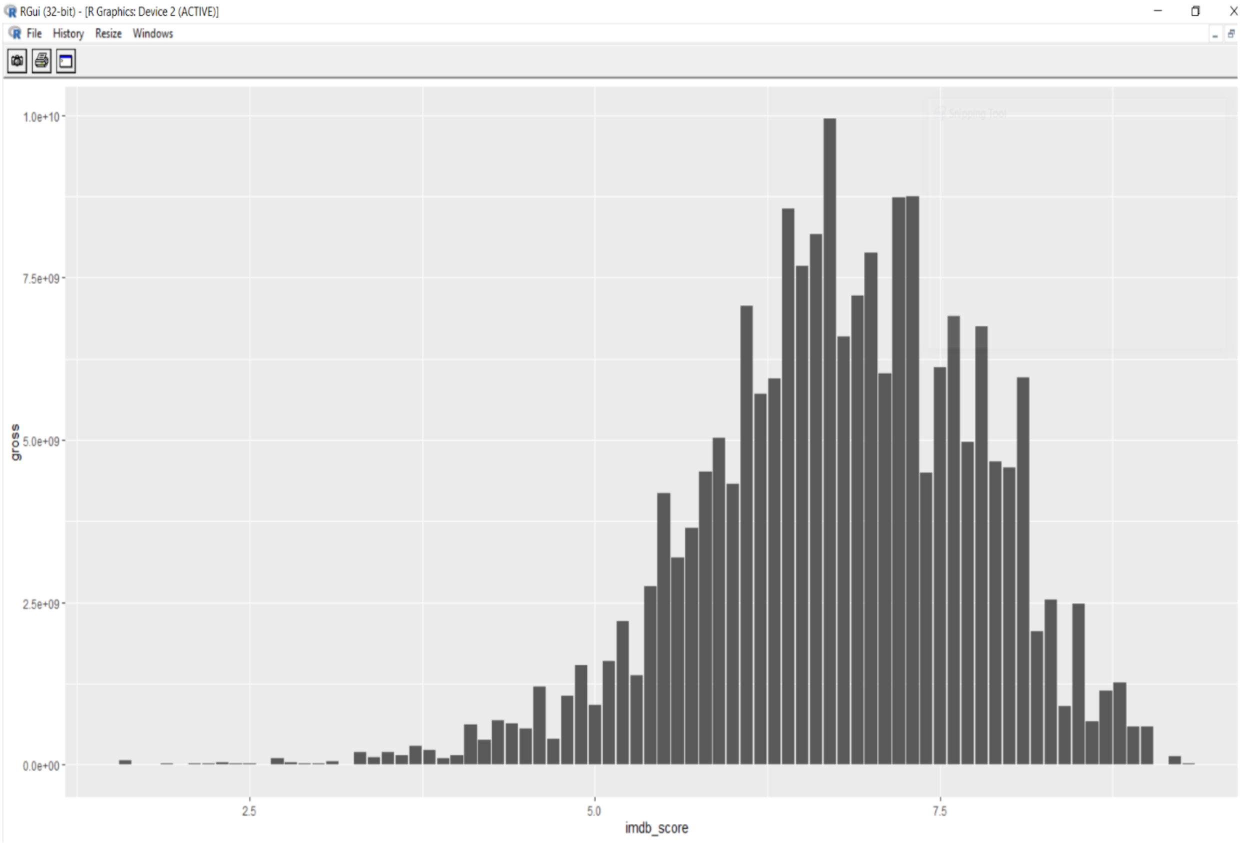
```
> dim(finalmoview)
[1] 3873   15
> |
```

After removing null values from the dataset - Figure 5.

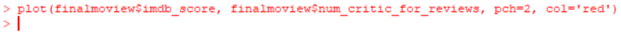


```
could not find function "ggplot"
> library(ggplot2)
> ggplot(data = finalmoview, mapping = aes(x = imdb_score, y = gross)) +
+   geom_col()
> |
```

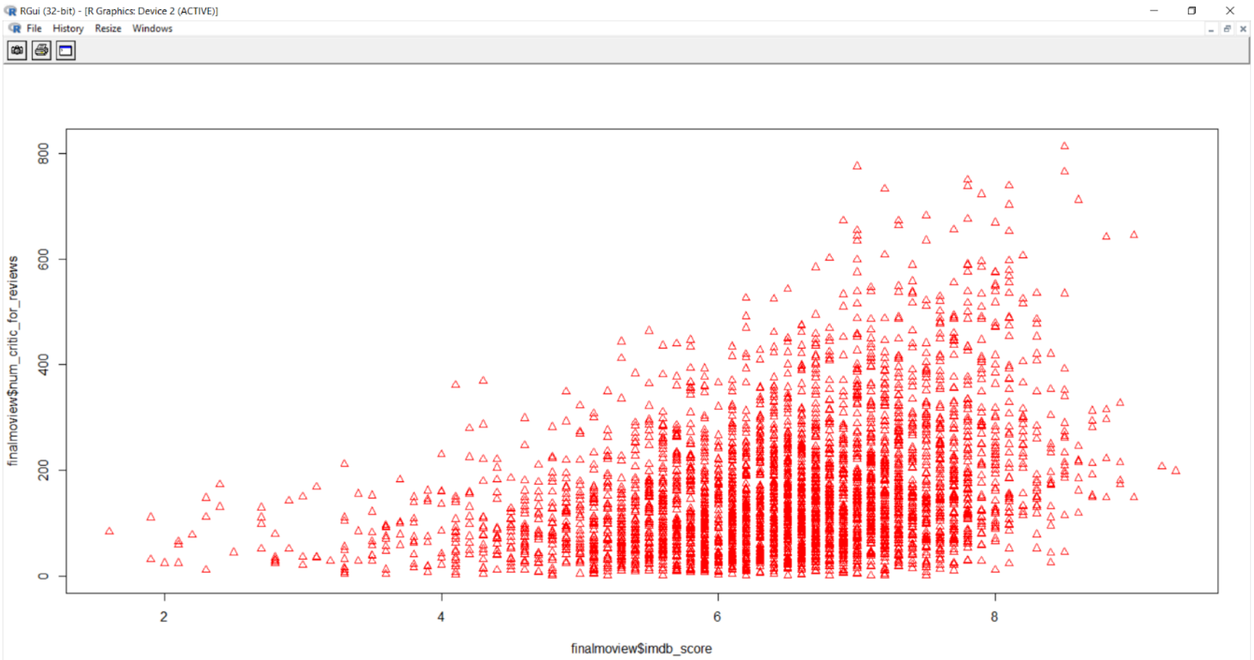
Relationship between IMDB score and gross - Figure 6.



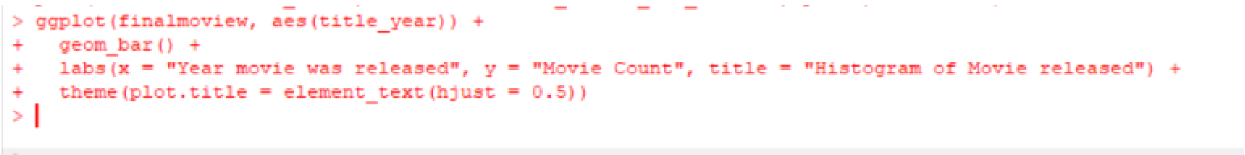
IMDB score is get reduced - Figure 7.



```
> plot(finalmoview$imdb_score, finalmoview$num_critic_for_reviews, pch=2, col='red')
> |
```

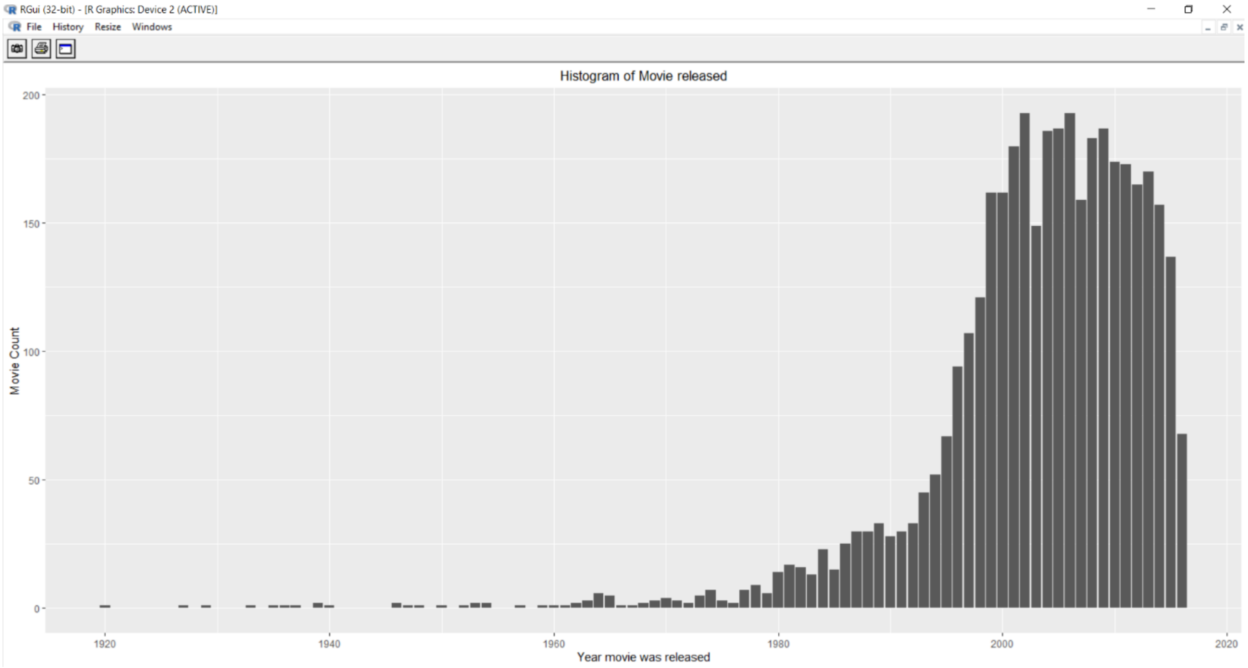


IMDB and number of critic reviews- Figure 8.

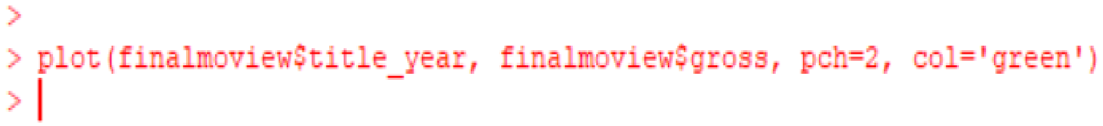


```
> ggplot(finalmoview, aes(title_year)) +
+   geom_bar() +
+   labs(x = "Year movie was released", y = "Movie Count", title = "Histogram of Movie released") +
+   theme(plot.title = element_text(hjust = 0.5))
> 
```

The movie released year and number of movies- Figure 9

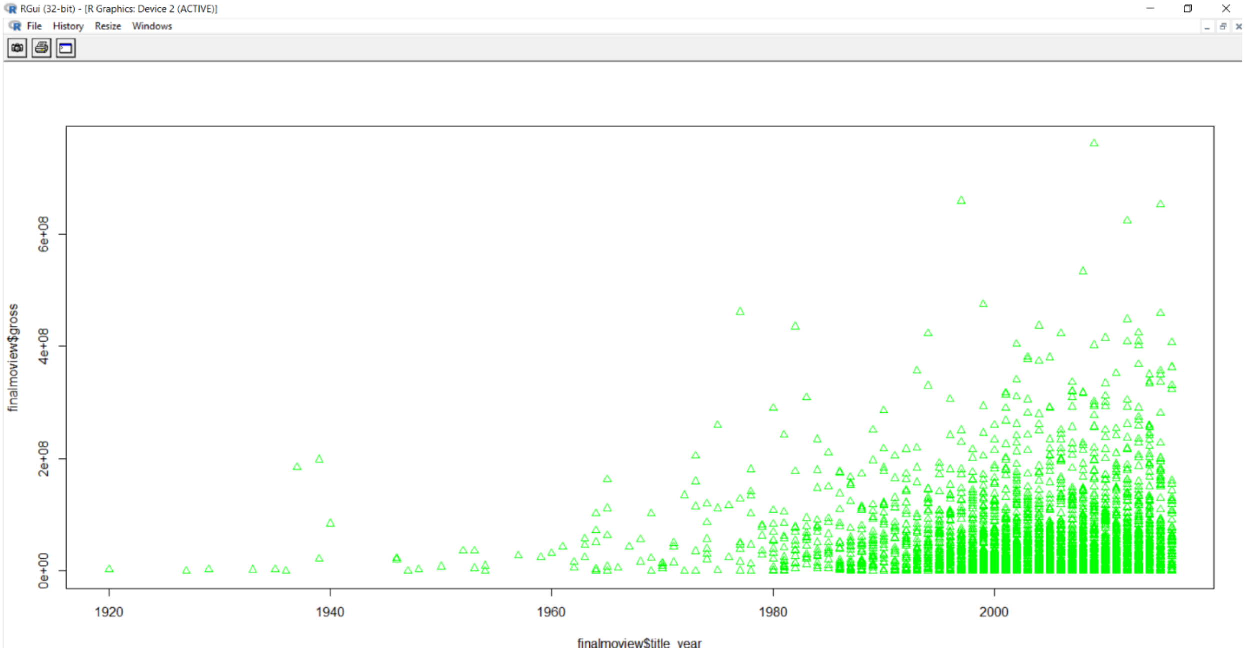


A greater number of movies are released after the year 2000 - Figure 10.

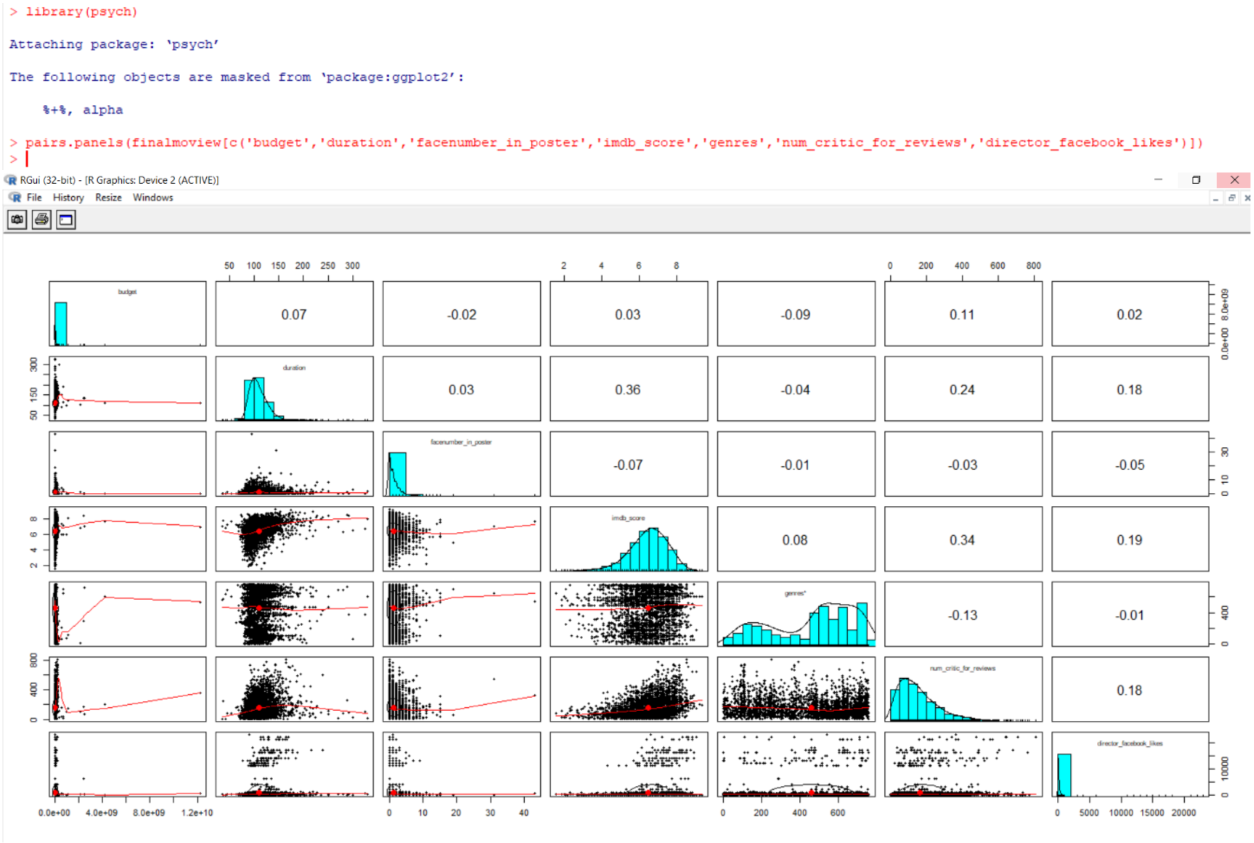


```
>
> plot(finalmoview$title_year, finalmoview$gross, pch=2, col='green')
> 
```

Movie release year and gross - Figure 11.

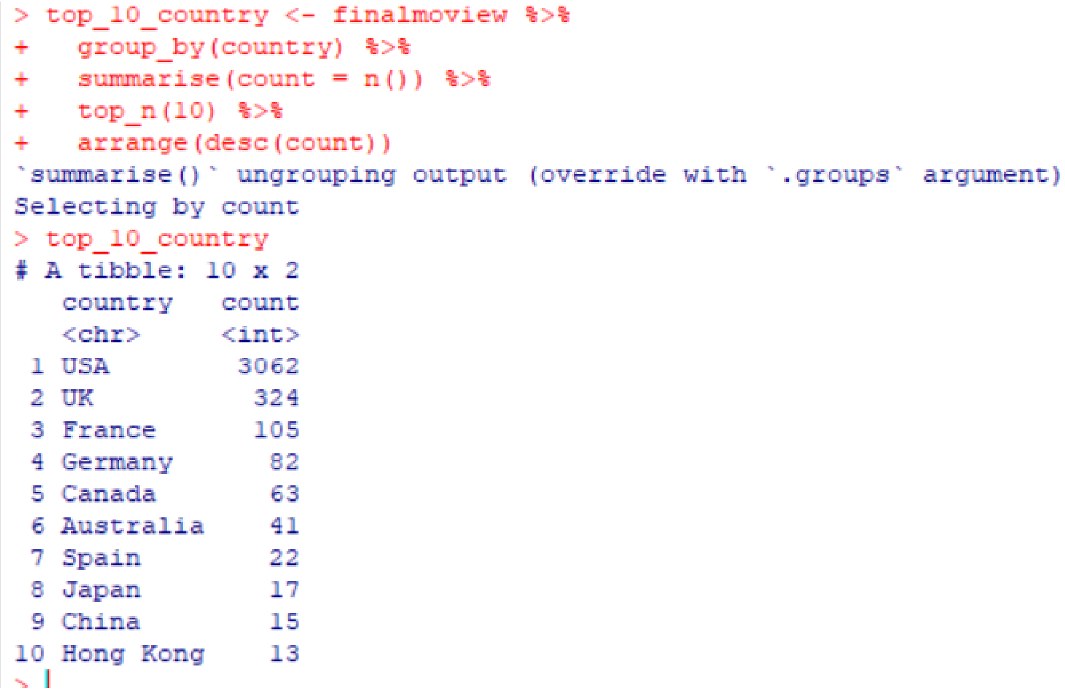


Movie release year and gross - Figure 12.



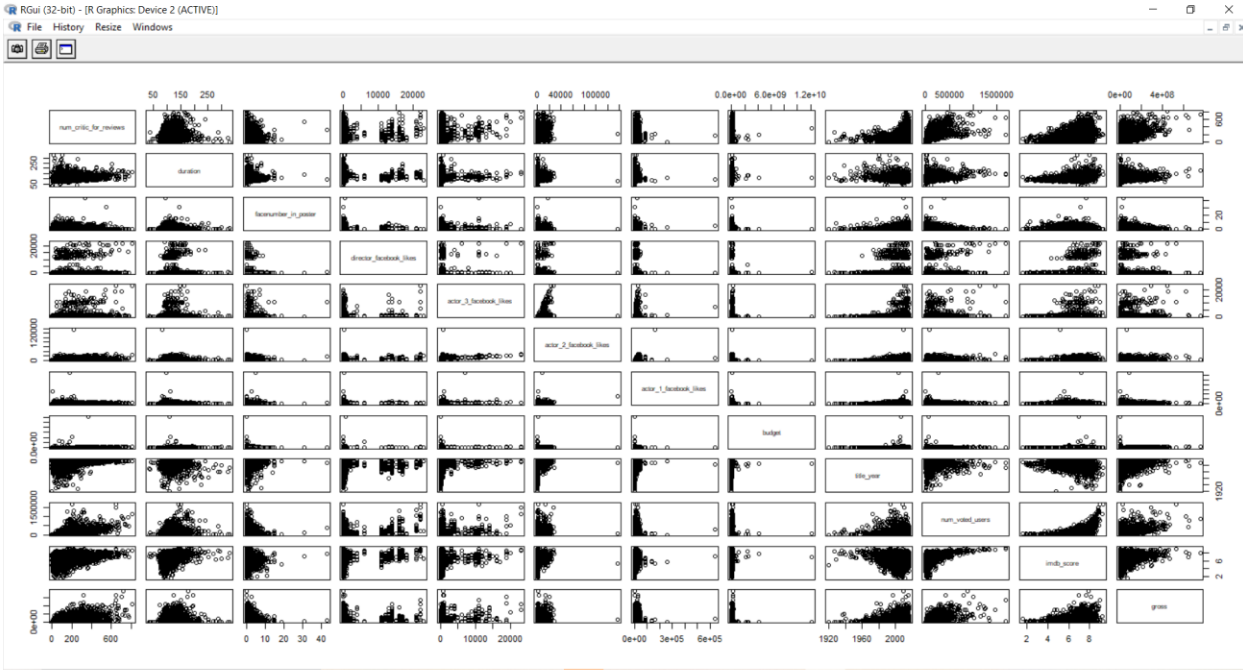
```
> library(psych)

Attaching package: 'psych'

The following objects are masked from 'package:ggplot2':

    %+%, alpha

> pairs.panels(finalmoview[c('budget','duration','facenumber_in_poster','imdb_score','genres','num_critic_for_reviews','director_facebook_likes')])
> 
```

Correlation analysis - Figure 13.

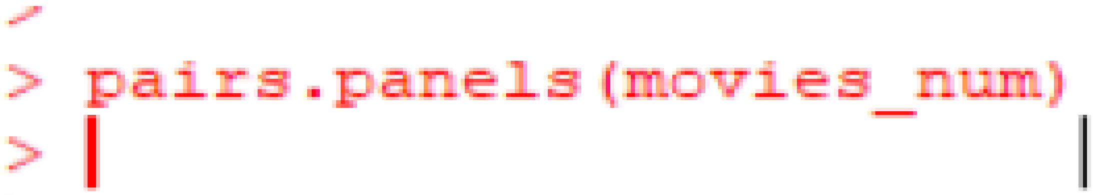


```
> top_10_country <- finalmoview %>%
+   group_by(country) %>%
+   summarise(count = n()) %>%
+   top_n(10) %>%
+   arrange(desc(count))
`summarise()` ungrouping output (override with `.groups` argument)
Selecting by count
> top_10_country
# A tibble: 10 x 2
   country   count
   <chr>     <int>
 1 USA        3062
 2 UK          324
 3 France      105
 4 Germany      82
 5 Canada       63
 6 Australia    41
 7 Spain        22
 8 Japan        17
 9 China        15
10 Hong Kong    13
> 
```

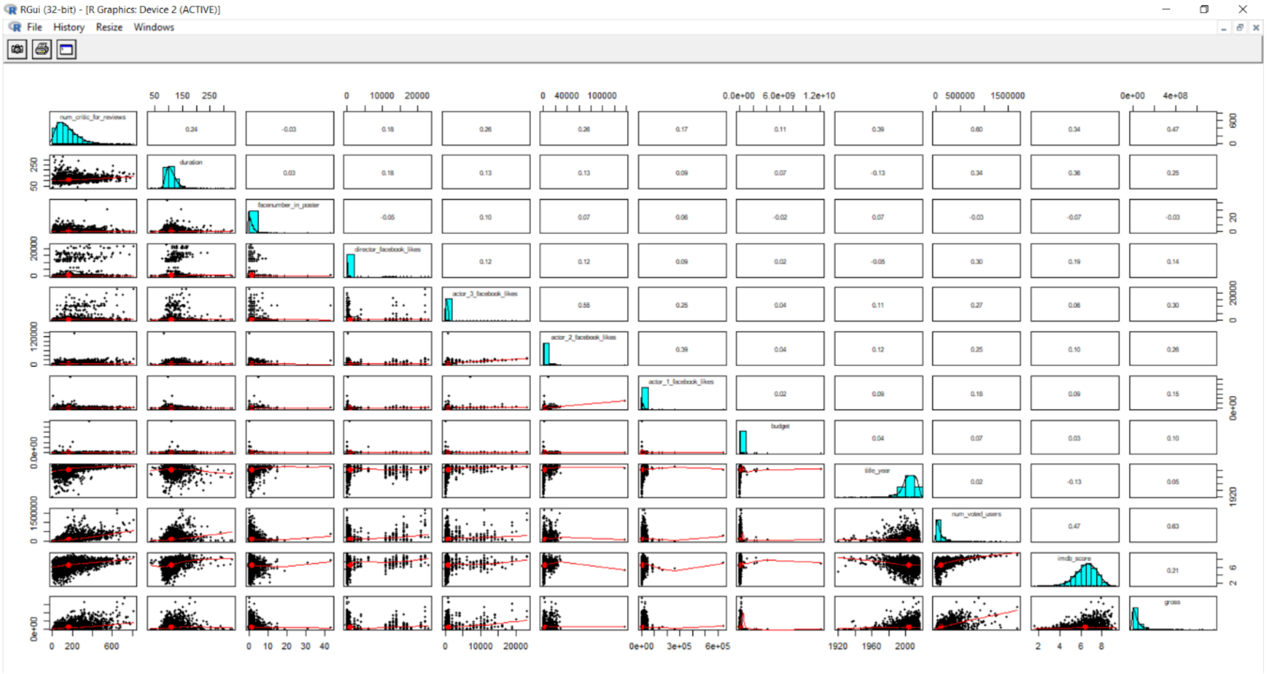
Country analysis - Figure 14.



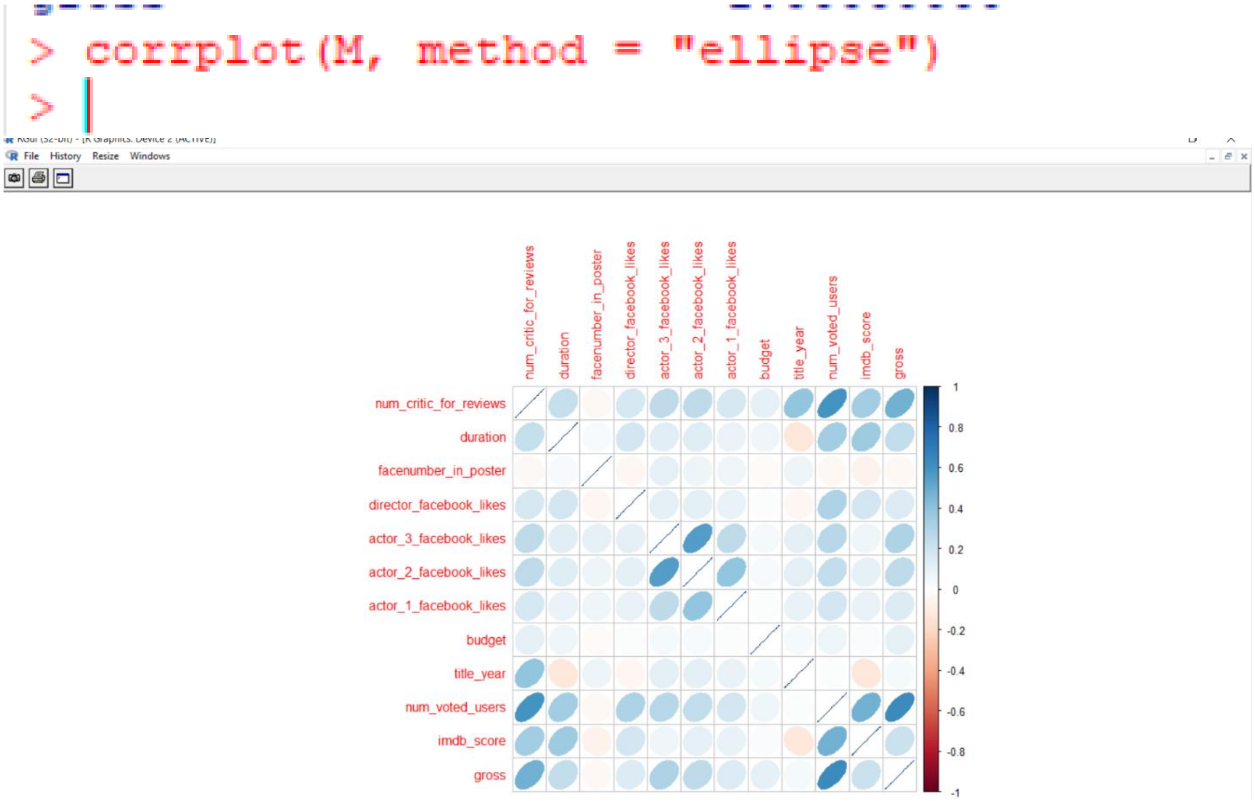
Movie Subset of variable analysis - Figure 15.



```
> pairs.panels(movies_num)
> 
```

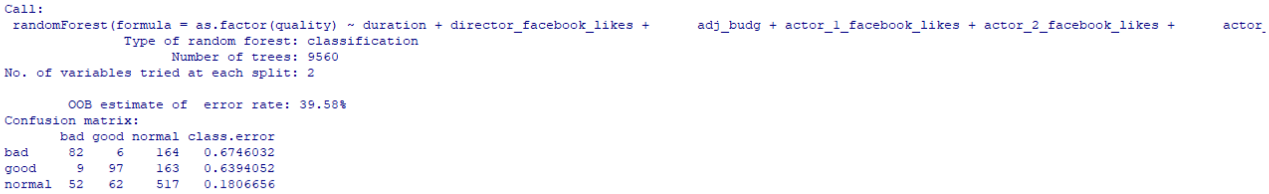


Movie Subset of variable analysis - Figure 16.



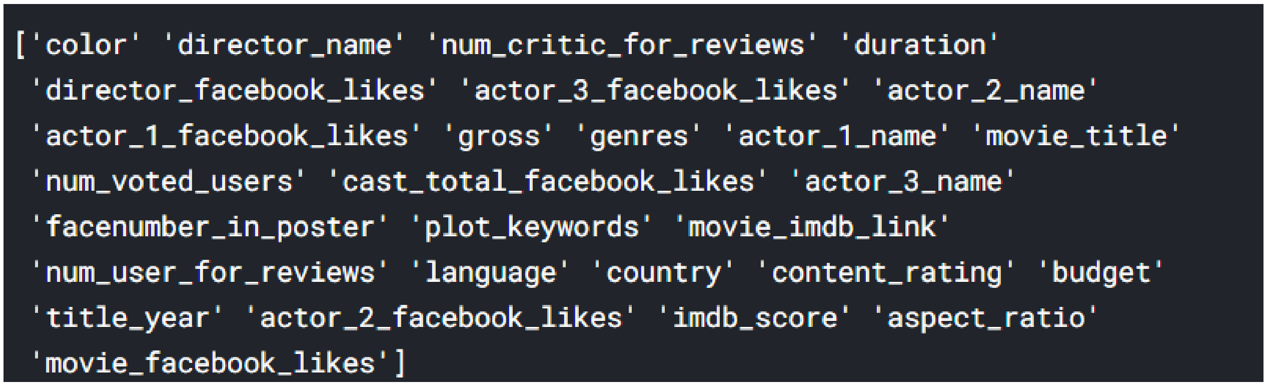
```
> corrplot(M, method = "ellipse")
>
```

Correlation plot for the analyzed variables - Figure 17.



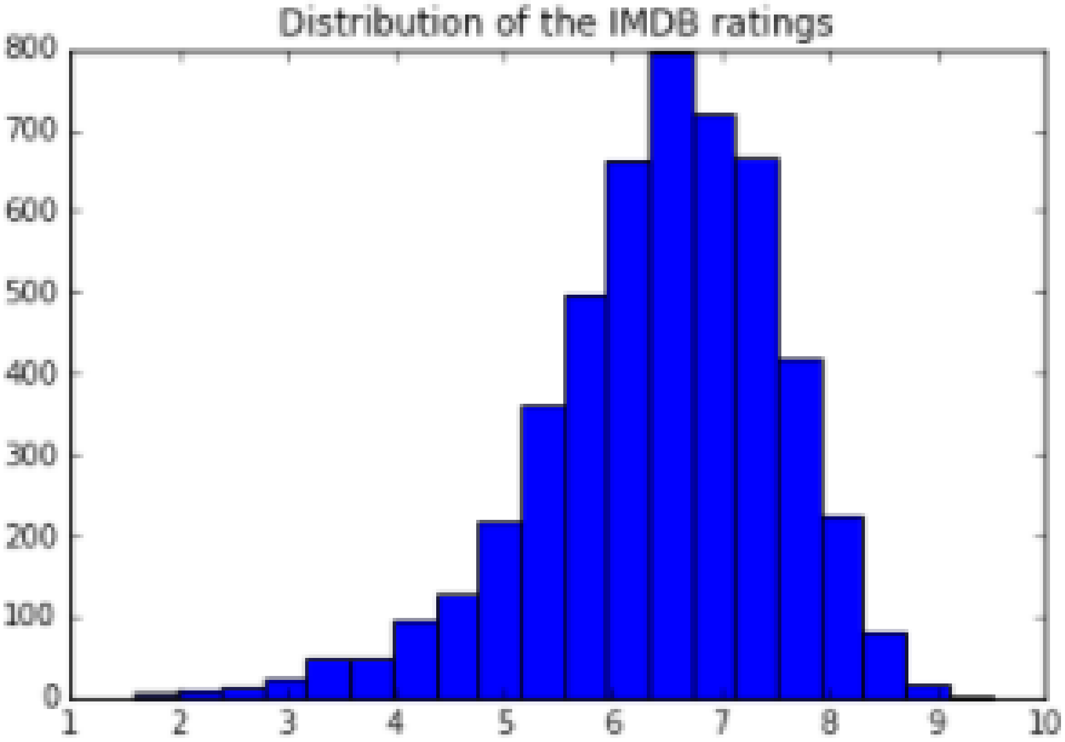
```
Call:
 randomForest(formula = as.factor(quality) ~ duration + director_facebook_likes +      adj_budg + actor_1_facebook_likes + actor_2_facebook_likes +      actor_
              Type of random forest: classification
                    Number of trees: 9560
No. of variables tried at each split: 2

        OOB estimate of  error rate: 39.58%
Confusion matrix:
       bad good normal class.error
bad     82    6    164   0.6746032
good     9   97    163   0.6394052
normal  52   62    517   0.1806656
```

Model build - Figure 18.

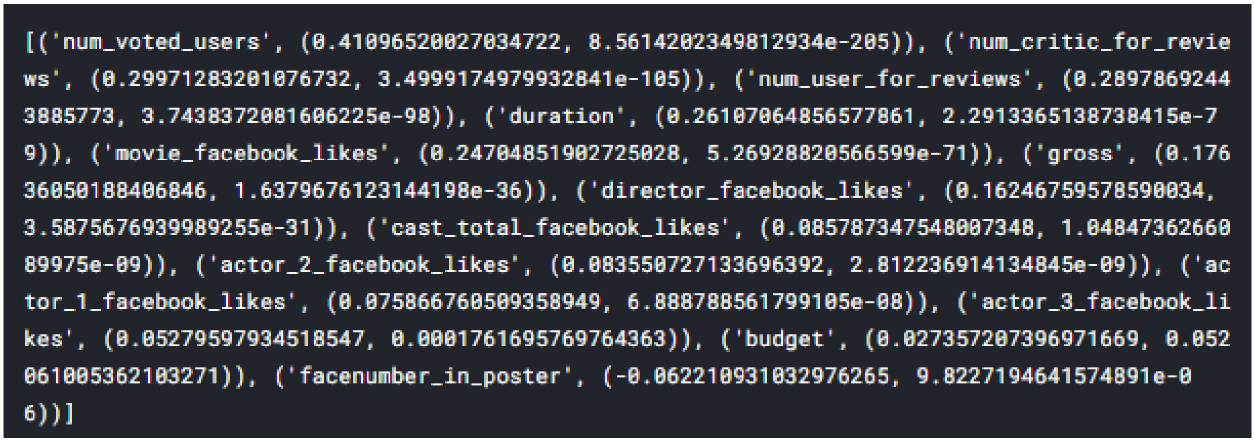


```
['color' 'director_name' 'num_critic_for_reviews' 'duration'
 'director_facebook_likes' 'actor_3_facebook_likes' 'actor_2_name'
 'actor_1_facebook_likes' 'gross' 'genres' 'actor_1_name' 'movie_title'
 'num_voted_users' 'cast_total_facebook_likes' 'actor_3_name'
 'facenumber_in_poster' 'plot_keywords' 'movie_imdb_link'
 'num_user_for_reviews' 'language' 'country' 'content_rating' 'budget'
 'title_year' 'actor_2_facebook_likes' 'imdb_score' 'aspect_ratio'
 'movie_facebook_likes']
```

The column names are displayed - Figure 19.

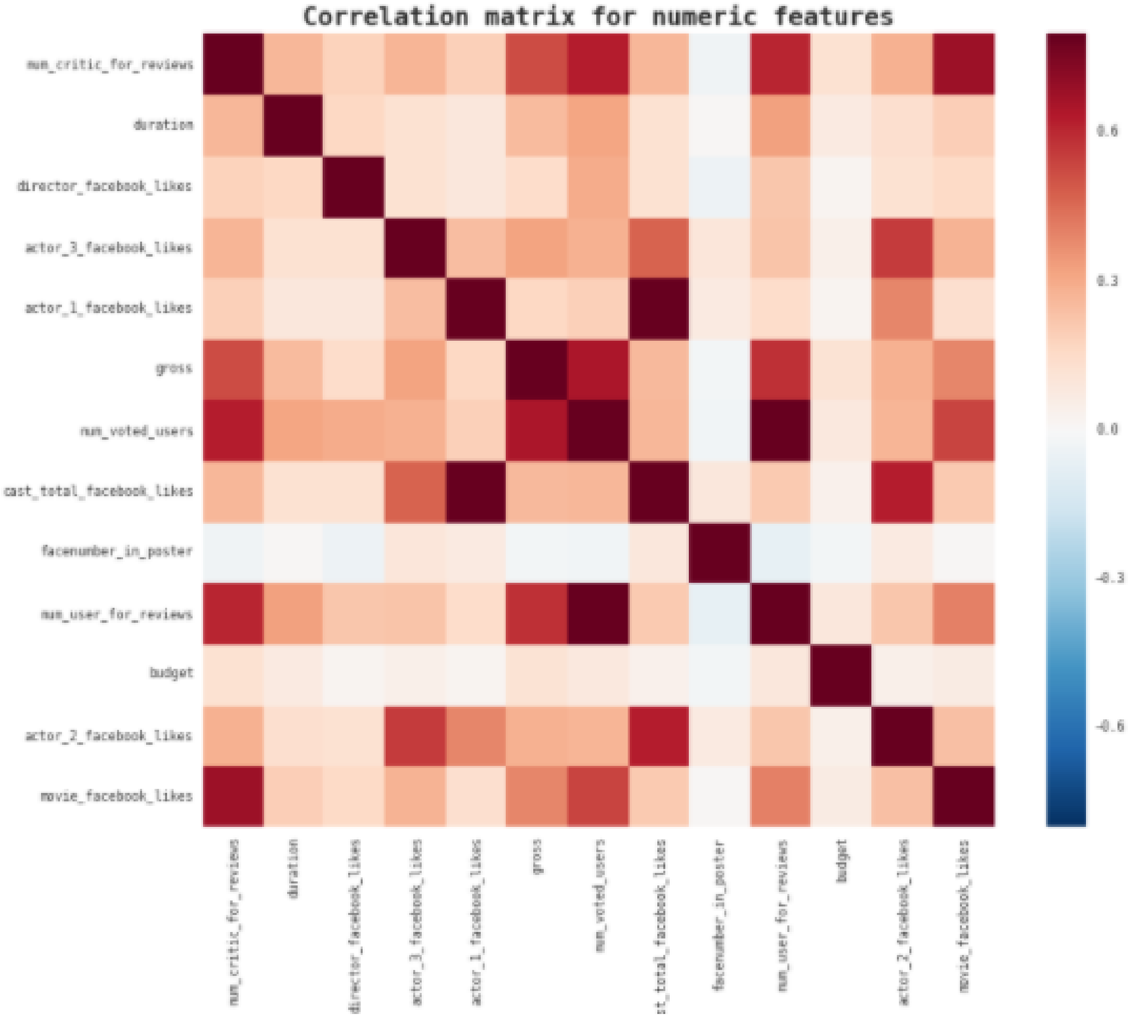


The imdb score distribution is illustrated - Figure 20.

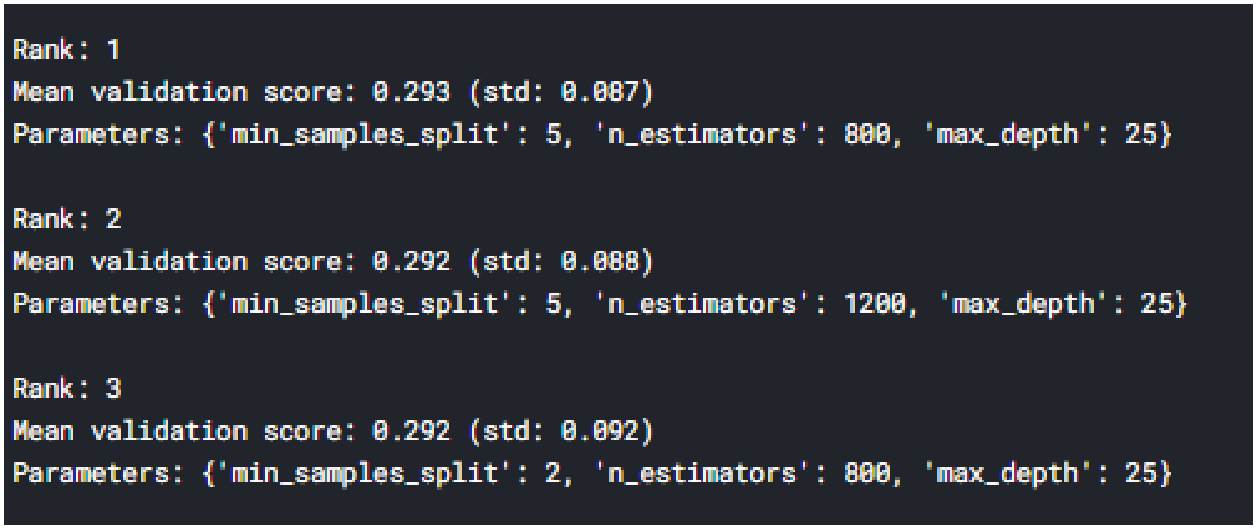


```
[('num_voted_users', (0.41096520027034722, 8.5614202349812934e-205)), ('num_critic_for_revie
ws', (0.29971283201076732, 3.4999174979932841e-105)), ('num_user_for_reviews', (0.2897869244
3885773, 3.7438372081606225e-98)), ('duration', (0.26107064856577861, 2.2913365138738415e-7
9)), ('movie_facebook_likes', (0.24704851902725028, 5.26928820566599e-71)), ('gross', (0.176
36050188406846, 1.6379676123144198e-36)), ('director_facebook_likes', (0.16246759578590034,
3.5875676939989255e-31)), ('cast_total_facebook_likes', (0.085787347548007348, 1.04847362660
89975e-09)), ('actor_2_facebook_likes', (0.083550727133696392, 2.812236914134845e-09)), ('ac
tor_1_facebook_likes', (0.075866760509358949, 6.888788561799105e-08)), ('actor_3_facebook_li
kes', (0.05279597934518547, 0.0001761695769764363)), ('budget', (0.027357207396971669, 0.052
061005362103271)), ('facenumber_in_poster', (-0.062210931032976265, 9.8227194641574891e-0
6))]
```

Illustrates that on average, most of the movies got a score of 6.5. - Figure 21.

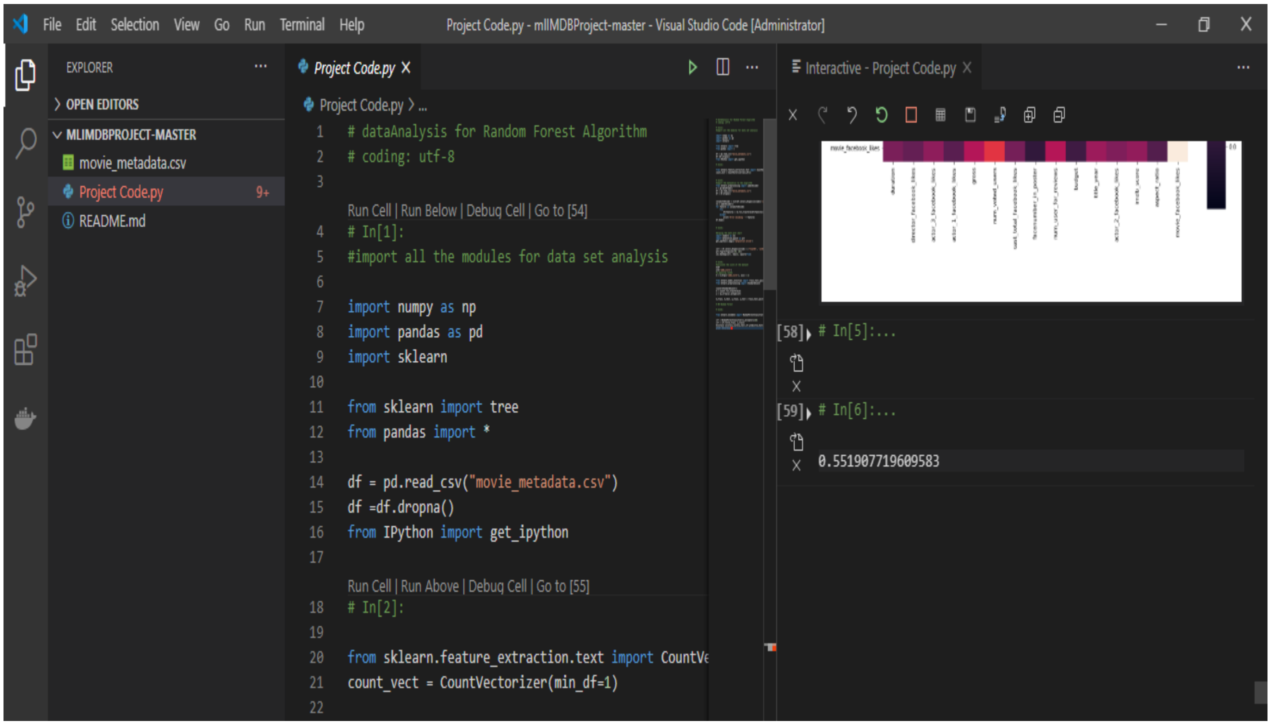


The correlation coefficient of the numeric features Figure 22.



```
Rank: 1
Mean validation score: 0.293 (std: 0.087)
Parameters: {'min_samples_split': 5, 'n_estimators': 800, 'max_depth': 25}

Rank: 2
Mean validation score: 0.292 (std: 0.088)
Parameters: {'min_samples_split': 5, 'n_estimators': 1200, 'max_depth': 25}

Rank: 3
Mean validation score: 0.292 (std: 0.092)
Parameters: {'min_samples_split': 2, 'n_estimators': 800, 'max_depth': 25}
```

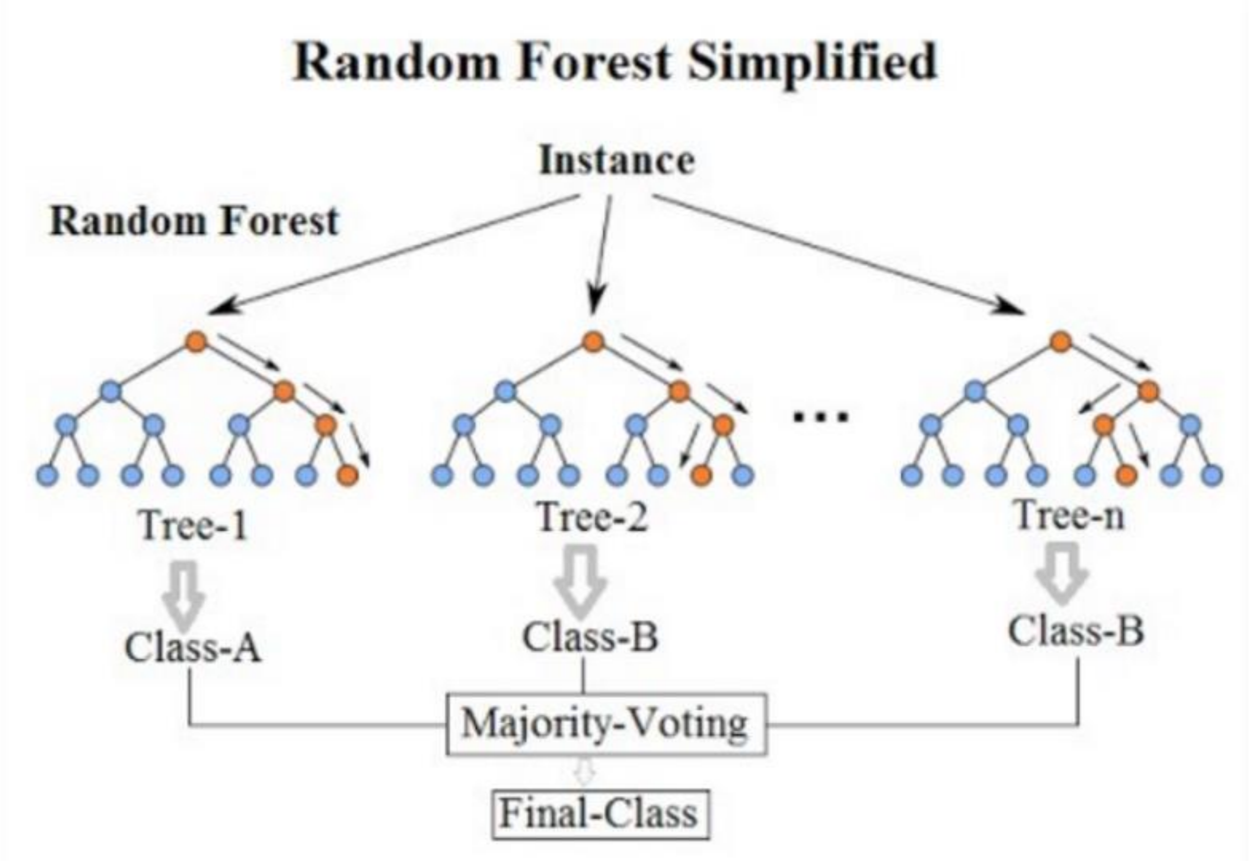
The correlation matrix shows the relationship between each considered variables. Figure 23.



```python
# dataAnalysis for Random Forest Algorithm
# coding: utf-8

# In[1]:
#import all the modules for data set analysis

import numpy as np
import pandas as pd
import sklearn

from sklearn import tree
from pandas import *

df = pd.read_csv("movie_metadata.csv")
df =df.dropna()
from IPython import get_ipython

# In[2]:

from sklearn.feature_extraction.text import CountVe
count_vect = CountVectorizer(min_df=1)
```

```
[58] # In[5]:...

[59] # In[6]:...

0.551907719609583
```

The effects of Facebook likeS on movie dataset Figure 24



Decision tree - Figure 25