Prediction of movies popularity

Saurabh Sankhe

February 22, 2019

The purpose of this project is to develop mutliple linear regression model to analyze the factors that will make a movie popular. The dataset contains the information that are extracted from IMDB for random sample movies. For popularity we are going to measure the audience_score as an output variable and the attributes will be the type of movie, genre, runtime, imdb rating, imdb number of votes, critics rating, critics score, audience rating, Oscar awards obtained (actor, actress, director and picture).

if all these attributes are related significantly then we can find the popularity of movie.

Load packages

```
library(ggplot2)
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(statsr)

## Warning: package 'statsr' was built under R version 3.5.2

## Loading required package: BayesFactor

## Warning: package 'BayesFactor' was built under R version 3.5.2
## Loading required package: coda

## Warning: package 'coda' was built under R version 3.5.2

## Loading required package: Matrix

## ************
## Welcome to BayesFactor 0.9.12-4.2. If you have questions, please contact
Richard Morey (richarddmorey@gmail.com).
##
## Type BFManual() to open the manual.
## ************

library(gridExtra)

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

library(corrplot)

## corrplot 0.84 loaded
```

Load the data

```
mydata <- load("C:/Users/Saurabh/Desktop/Sem-2 Course Documents/Multivariate
Analysis/Movies/movies.Rdata")

movies_new <- movies %>% select(title, title_type, genre, runtime,
imdb_rating, imdb_num_votes, critics_rating, critics_score, audience_rating,
audience_score, best_pic_win, best_actor_win, best_actress_win, best_dir_win)

str(movies_new)

## Classes 'tbl_df', 'tbl' and 'data.frame':    651 obs. of  14 variables:
##  $ title           : chr  "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  80 101 84 139 90 78 142 93 88 119 ...
##  $ imdb_rating     : num  5.5 7.3 7.6 7.2 5.1 7.8 7.2 5.5 7.5 6.6 ...
##  $ imdb_num_votes  : int  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  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  73 81 91 76 27 86 76 47 89 66 ...
##  $ 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
...
```

```
summary(movies_new)
```

```
##     title                  title_type                 genre
##  Length:651         Documentary : 55   Drama             :305
##  Class :character   Feature Film:591   Comedy            : 87
##  Mode  :character   TV Movie    :  5   Action & Adventure: 65
##                                        Mystery & Suspense: 59
##                                        Documentary       : 52
##                                        Horror            : 23
##                                        (Other)           : 60
##     runtime       imdb_rating    imdb_num_votes           critics_rating
##  Min.   : 39.0   Min.   :1.900   Min.   :   180   Certified Fresh:135
##  1st Qu.: 92.0   1st Qu.:5.900   1st Qu.:  4546   Fresh          :209
##  Median :103.0   Median :6.600   Median : 15116   Rotten         :307
##  Mean   :105.8   Mean   :6.493   Mean   : 57533
##  3rd Qu.:115.8   3rd Qu.:7.300   3rd Qu.: 58301
##  Max.   :267.0   Max.   :9.000   Max.   :893008
##  NA's   :1
##  critics_score    audience_rating audience_score  best_pic_win
##  Min.   :  1.00   Spilled:275     Min.   :11.00   no :644
##  1st Qu.: 33.00   Upright:376     1st Qu.:46.00   yes:  7
##  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_actor_win best_actress_win best_dir_win
##  no :558        no :579          no :608
##  yes: 93        yes: 72          yes: 43
##
##
##
##
##
```

Drop missing value

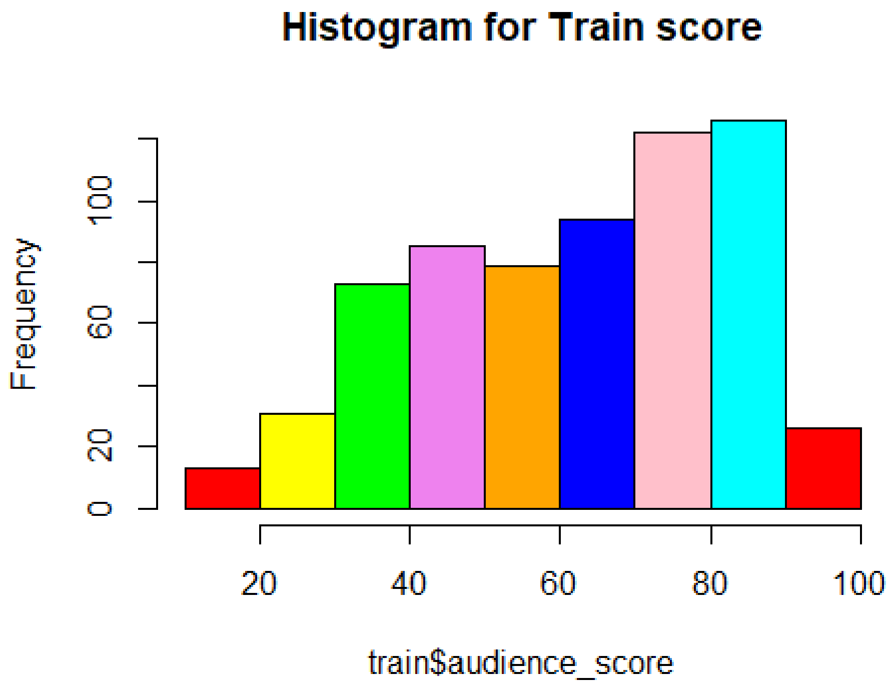
```
movies_new <- na.omit(movies_new)
```

Split data into train and test

```
set.seed(2017)
split <- sample(seq_len(nrow(movies_new)), size = floor(0.999 *
nrow(movies_new)))
train <- movies_new[split, ]
test <- movies_new[-split, ]
```

histogram

```
colors = c("red", "yellow", "green", "violet", "orange", "blue", "pink",
"cyan")
hist(train$audience_score, col=colors, main = "Histogram for Train score")
```

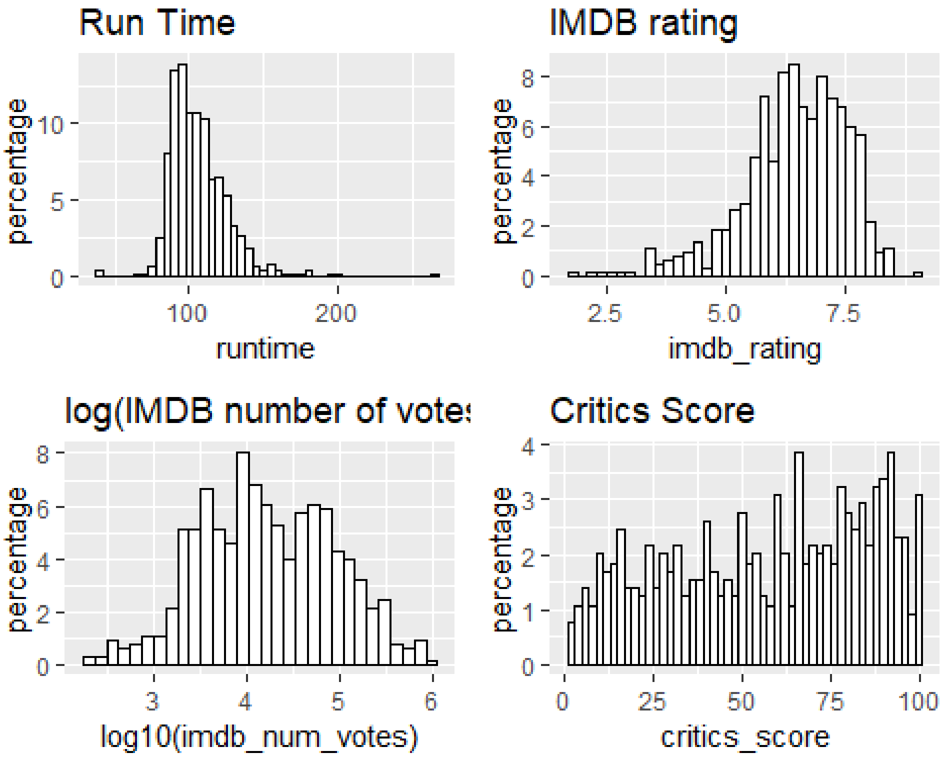


```
summary(train$audience_score)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    11.0    46.0    65.0    62.3    80.0    97.0
```

The median of our response variable - audience score distribution is 65; 75% of the movie in the training set have an audience score higher than 80; 25% of the movie in the training set have an audience score lower than 46; very few movie have an audience score lower than 20 or higher than 90

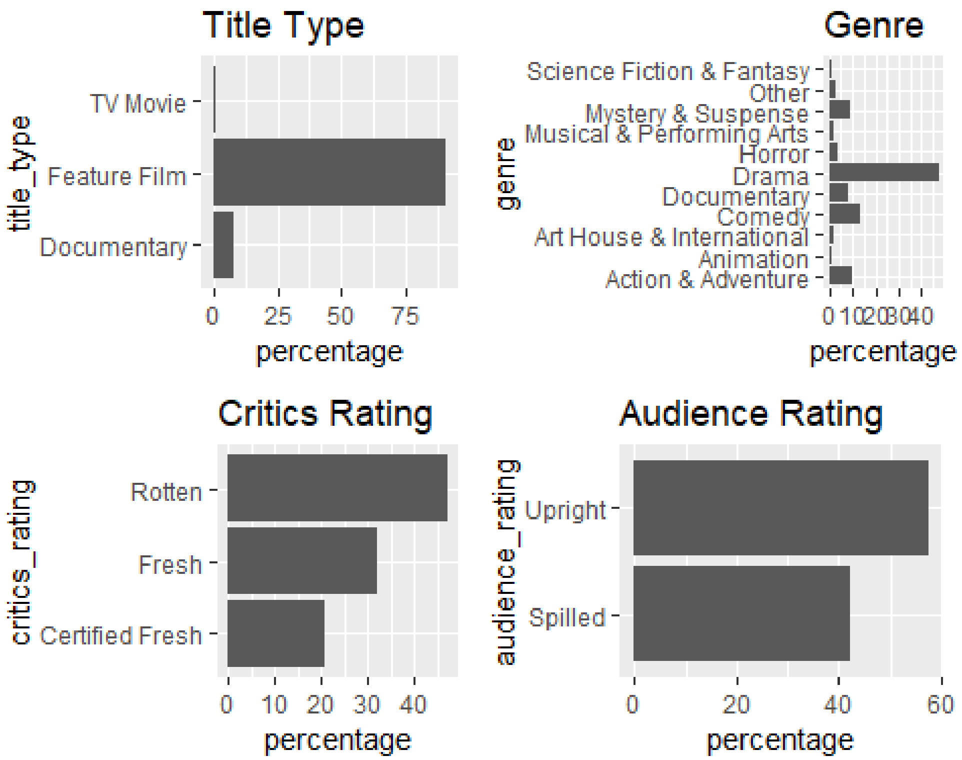
```
p1 <- ggplot(aes(x=runtime), data=train) +
  geom_histogram(aes(y=100*(..count..)/sum(..count..)), color='black',
fill='white', binwidth = 5) + ylab('percentage') + ggtitle('Run Time')
p2 <- ggplot(aes(x=imdb_rating), data=train) +
  geom_histogram(aes(y=100*(..count..)/sum(..count..)), color='black',
fill='white', binwidth = 0.2) + ylab('percentage') + ggtitle('IMDB rating')
p3 <- ggplot(aes(x=log10(imdb_num_votes)), data=train) +
  geom_histogram(aes(y=100*(..count..)/sum(..count..)), color='black',
fill='white') + ylab('percentage') + ggtitle('log(IMDB number of votes)')
p4 <- ggplot(aes(x=critics_score), data=train) +
  geom_histogram(aes(y=100*(..count..)/sum(..count..)), color='black',
fill='white', binwidth = 2) + ylab('percentage') + ggtitle('Critics Score')
grid.arrange(p1, p2, p3, p4, ncol=2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



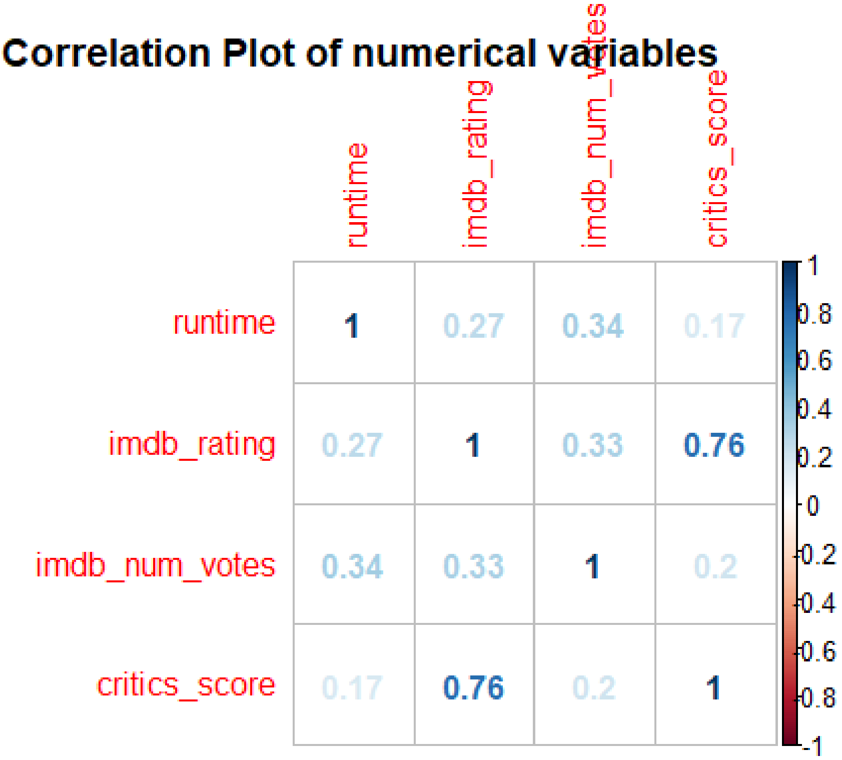
#Regression analysis: Run time, IMDB rating, log(IMDB number of votes) and Critics Scores all have reasonable broad distribution, therefore, they will be considered for the regression analysis.

```
p1 <- ggplot(aes(x=title_type), data=train) +
geom_bar(aes(y=100*(..count..)/sum(..count..))) + ylab('percentage') +
  ggtitle('Title Type') + coord_flip()
p2 <- ggplot(aes(x=genre), data=train) +
geom_bar(aes(y=100*(..count..)/sum(..count..))) + ylab('percentage') +
  ggtitle('Genre') + coord_flip()
p3 <- ggplot(aes(x=critics_rating), data=train) +
geom_bar(aes(y=100*(..count..)/sum(..count..))) + ylab('percentage') +
  ggtitle('Critics Rating') + coord_flip()
p4 <- ggplot(aes(x=audience_rating), data=train) +
geom_bar(aes(y=100*(..count..)/sum(..count..))) + ylab('percentage') +
  ggtitle('Audience Rating') + coord_flip()
grid.arrange(p1, p2, p3, p4, ncol=2)
```



#Most movies in the data are in the "Feature Film" title type and majority of the movies are drama. Therefore, we must be aware that the results could be biased toward drama movies.

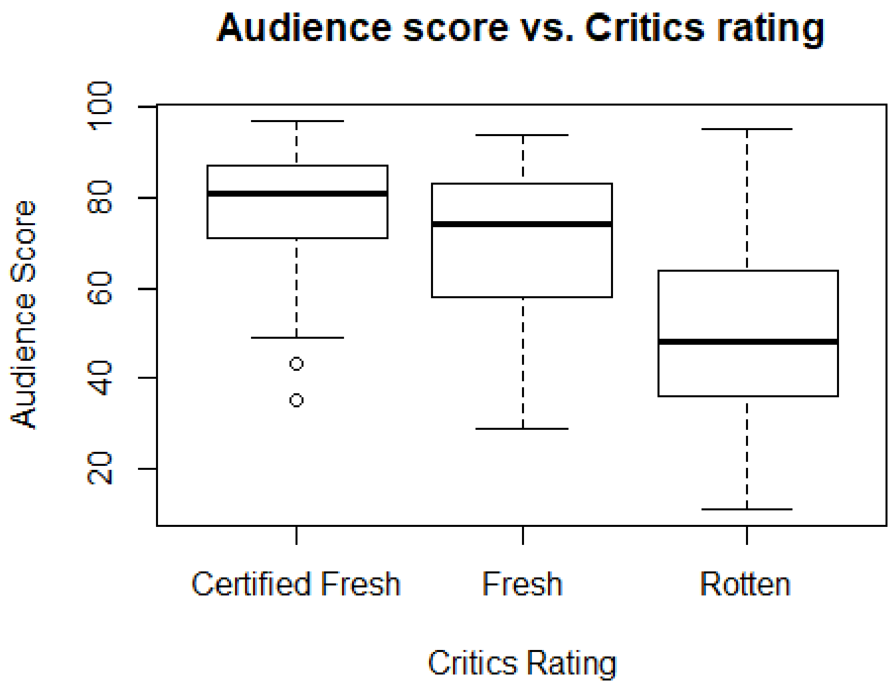
```
vars <- names(train) %in% c('runtime', 'imdb_rating', 'imdb_num_votes',
'critics_score')
selected_train <- train[vars]
corr.matrix <- cor(selected_train)
corrplot(corr.matrix, main="\n\nCorrelation Plot of numerical variables",
method="number")
```



Correlation Plot of numerical variables

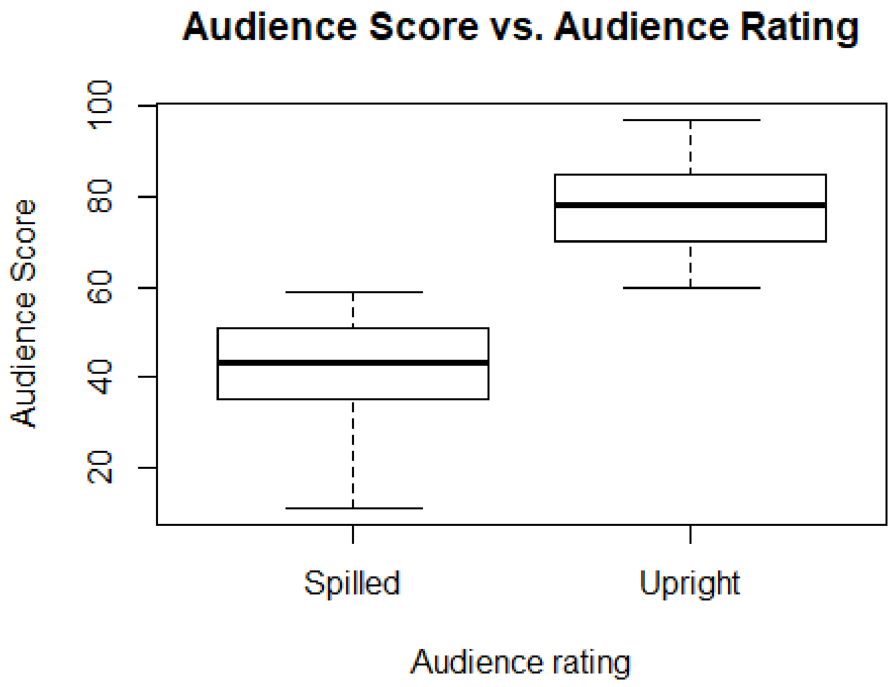
	runtime	imdb_rating	imdb_num_votes	critics_score
runtime	1	0.27	0.34	0.17
imdb_rating	0.27	1	0.33	0.76
imdb_num_votes	0.34	0.33	1	0.2
critics_score	0.17	0.76	0.2	1

```
boxplot(audience_score~critics_rating, data=train, main='Audience score vs.
Critics rating', xlab='Critics Rating', ylab='Audience Score')
```



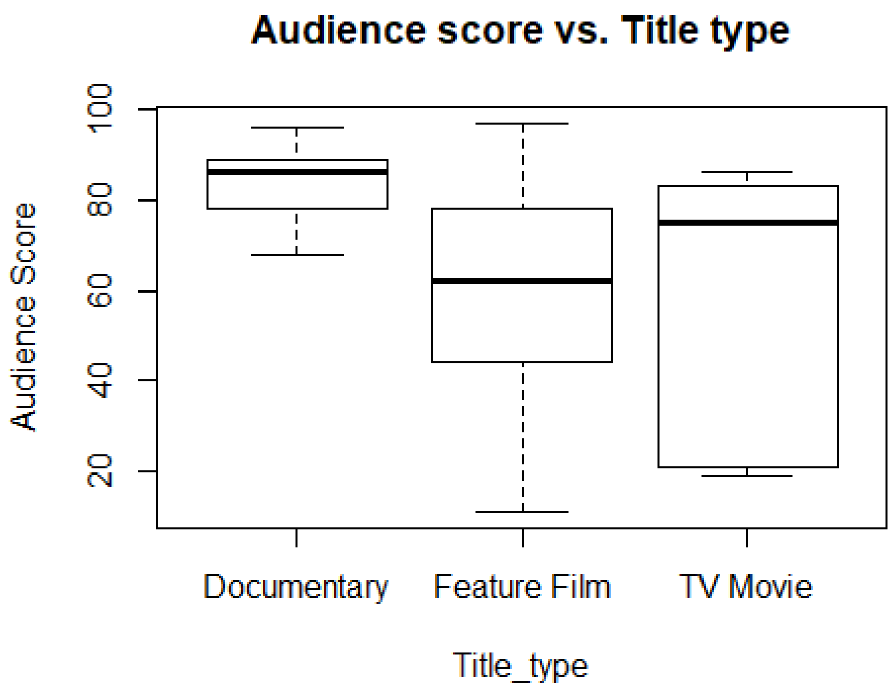
Audience score vs. Critics rating

```
by(train$audience_score, train$critics_rating, summary)

## train$critics_rating: Certified Fresh
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   35.00   71.00   81.00   79.26   87.00   97.00 
## -------------------------------------------------------- 
## train$critics_rating: Fresh
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   29.00   58.00   74.00   69.96   83.00   94.00 
## -------------------------------------------------------- 
## train$critics_rating: Rotten
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    11.0    36.0    48.0    49.7    64.0    95.0

boxplot(audience_score~audience_rating, data=train, main='Audience Score vs. 
Audience Rating', xlab='Audience rating', ylab='Audience Score')
```



```
by(train$audience_score, train$audience_rating, summary)

## train$audience_rating: Spilled
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   11.00   35.00   43.00   41.93   51.00   59.00 
## -------------------------------------------------------- 
## train$audience_rating: Upright
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   60.00   70.00   78.00   77.27   85.00   97.00
```

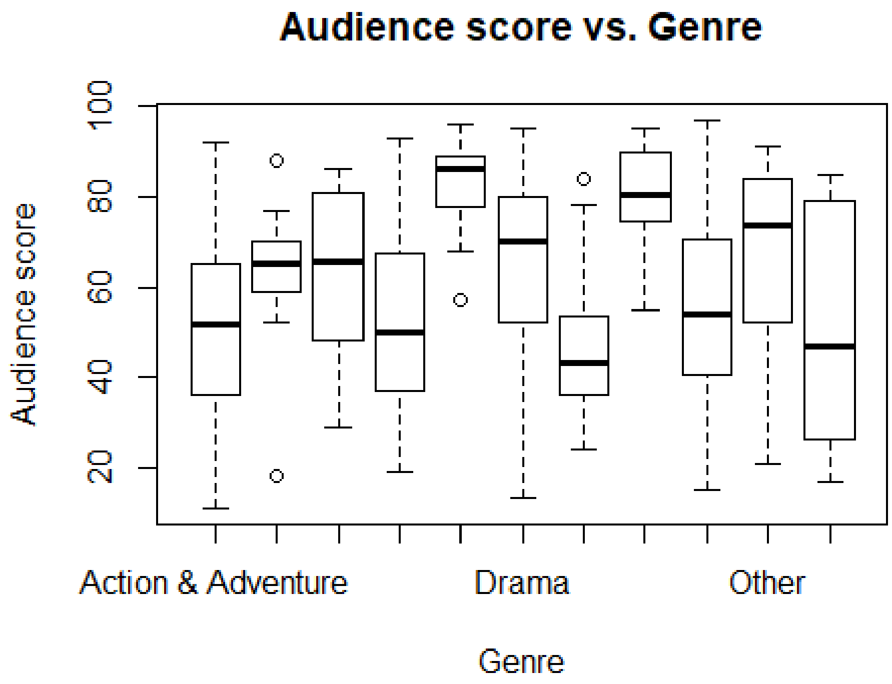
```r
boxplot(audience_score~title_type, data=train, main='Audience score vs. Title
type', xlab='Title_type', ylab='Audience Score')
```



```r
by(train$audience_score, train$title_type, summary)
```

```
## train$title_type: Documentary
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   68.00   78.00   86.00   83.46   89.00   96.00 
## -------------------------------------------------------- 
## train$title_type: Feature Film
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   11.00   44.25   62.00   60.41   78.00   97.00 
## -------------------------------------------------------- 
## train$title_type: TV Movie
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    19.0    21.0    75.0    56.8    83.0    86.0
```

```r
boxplot(audience_score~genre, data=train, main='Audience score vs. Genre',
xlab='Genre', ylab='Audience score')
```



```
by(train$audience_score, train$genre, summary)

## train$genre: Action & Adventure
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   11.00   36.50   51.50   53.16   65.00   92.00 
## -------------------------------------------------------- 
## train$genre: Animation
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   18.00   59.00   65.00   62.44   70.00   88.00 
## -------------------------------------------------------- 
## train$genre: Art House & International
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   29.00   51.25   65.50   64.00   80.25   86.00 
## -------------------------------------------------------- 
## train$genre: Comedy
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   19.00   37.00   50.00   52.51   67.50   93.00 
## -------------------------------------------------------- 
## train$genre: Documentary
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   57.00   77.50   86.00   82.96   89.00   96.00 
## -------------------------------------------------------- 
## train$genre: Drama
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   13.00   52.00   70.00   65.35   80.00   95.00 
## -------------------------------------------------------- 
## train$genre: Horror
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   24.00   36.00   43.00   45.83   53.50   84.00 
## -------------------------------------------------------- 
## train$genre: Musical & Performing Arts
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   55.00   75.75   80.50   80.17   89.50   95.00 
## -------------------------------------------------------- 
## train$genre: Mystery & Suspense
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   15.00   40.50   54.00   55.95   70.50   97.00 
## -------------------------------------------------------- 
## train$genre: Other
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   21.00   53.00   73.50   66.69   82.50   91.00 
## -------------------------------------------------------- 
## train$genre: Science Fiction & Fantasy
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   17.00   26.00   47.00   50.89   79.00   85.00
```

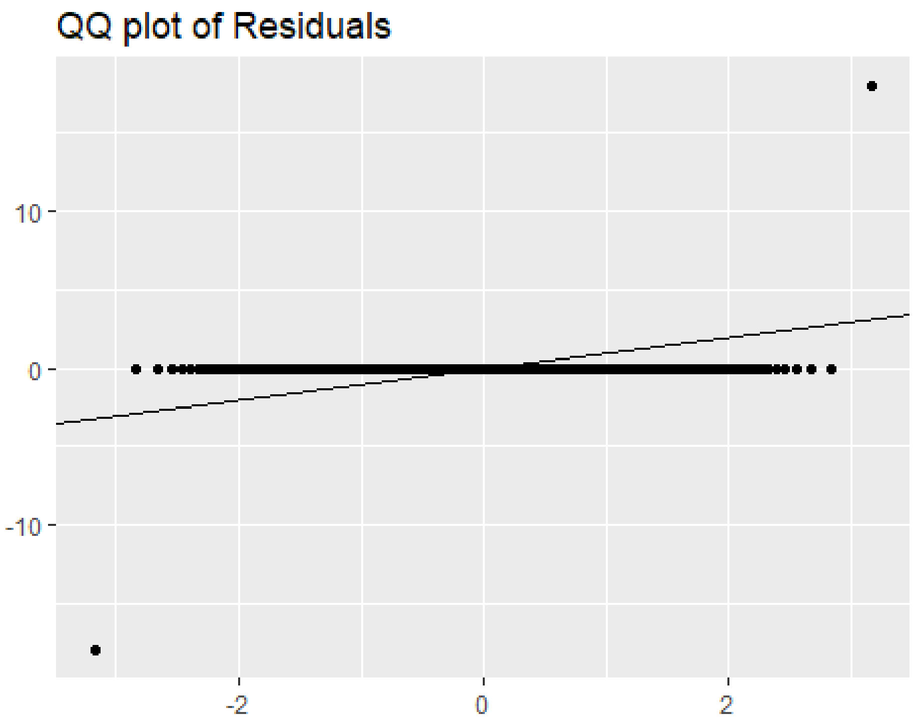
All the categorical variables seems to have reasonable significant correlation with audience score.

T-test

```
x <- 
c(movies_new$imdb_num_votes,movies_new$best_pic_win,movies_new$best_actor_win
,movies_new$best_actress_win,movies_new$best_dir_win)
t.test(movies_new$audience_score, x)

## 
##  Welch Two Sample t-test
## 
## data:  movies_new$audience_score and x
## t = -11.841, df = 3249, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -13360.601  -9564.579
## sample estimates:
##   mean of x   mean of y 
##    62.34769 11524.93785

movies_new$audience_score <- as.integer(movies_new$audience_score)
library(mosaic)

## Loading required package: lattice

## Loading required package: ggformula

## Loading required package: ggstance
```

```
##
## Attaching package: 'ggstance'

## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh

##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")

## Loading required package: mosaicData

##
## The 'mosaic' package masks several functions from core packages in order
to add
## additional features.  The original behavior of these functions should not
be affected by this.
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading
mosaic.

##
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':
##
##     mean

## The following objects are masked from 'package:dplyr':
##
##     count, do, tally

## The following object is masked from 'package:ggplot2':
##
##     stat

## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median,
##     prop.test, quantile, sd, t.test, var

## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```
model <- lm(audience_score~., data = movies_new)
plot <- qplot(sample= zscore(model$residuals), geom="qq", main="QQ plot of
Residuals")+ geom_abline()
plot
```



QQ plot of Residuals

Most movies in the data are in the “Feature Film” title type and majority of the movies are drama. Therefore, we must be aware that the results could be biased toward drama movies.