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
...
```

```
movies_new[c(2,3,7,9,11:14)] <- lapply(movies_new[c(2,3,7,9,11:14)],
as.numeric)

movies_data <- movies_new

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

summary(movies_new)
```

```
##     title             title_type        genre           runtime
##  Length:651         Min.   :1.000   Min.   : 1.000   Min.   : 39.0
##  Class :character   1st Qu.:2.000   1st Qu.: 4.000   1st Qu.: 92.0
##  Mode  :character   Median :2.000   Median : 6.000   Median :103.0
##                     Mean   :1.923   Mean   : 5.545   Mean   :105.8
##                     3rd Qu.:2.000   3rd Qu.: 6.000   3rd Qu.:115.8
##                     Max.   :3.000   Max.   :11.000   Max.   :267.0
##                                                      NA's   :1
##   imdb_rating    imdb_num_votes   critics_rating  critics_score
##  Min.   :1.900   Min.   :   180   Min.   :1.000   Min.   :  1.00
##  1st Qu.:5.900   1st Qu.:  4546   1st Qu.:2.000   1st Qu.: 33.00
##  Median :6.600   Median : 15116   Median :2.000   Median : 61.00
##  Mean   :6.493   Mean   : 57533   Mean   :2.264   Mean   : 57.69
##  3rd Qu.:7.300   3rd Qu.: 58301   3rd Qu.:3.000   3rd Qu.: 83.00
##  Max.   :9.000   Max.   :893008   Max.   :3.000   Max.   :100.00
##
##  audience_rating audience_score   best_pic_win   best_actor_win
##  Min.   :1.000   Min.   :11.00   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.:46.00   1st Qu.:1.000   1st Qu.:1.000
##  Median :2.000   Median :65.00   Median :1.000   Median :1.000
##  Mean   :1.578   Mean   :62.36   Mean   :1.011   Mean   :1.143
##  3rd Qu.:2.000   3rd Qu.:80.00   3rd Qu.:1.000   3rd Qu.:1.000
##  Max.   :2.000   Max.   :97.00   Max.   :2.000   Max.   :2.000
##
##  best_actress_win  best_dir_win
##  Min.   :1.000    Min.   :1.000
```

```
##  1st Qu.:1.000    1st Qu.:1.000
##  Median :1.000    Median :1.000
##  Mean   :1.111    Mean   :1.066
##  3rd Qu.:1.000    3rd Qu.:1.000
##  Max.   :2.000    Max.   :2.000
##

View(movies_new)
```

Drop missing value

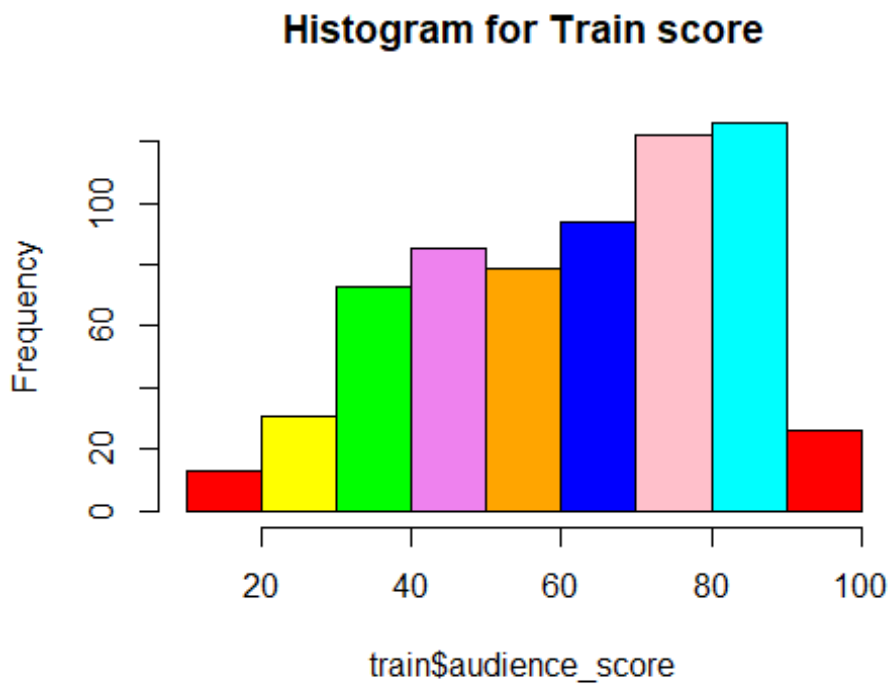
```
movies_new <- na.omit(movies_new)
```

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")
```



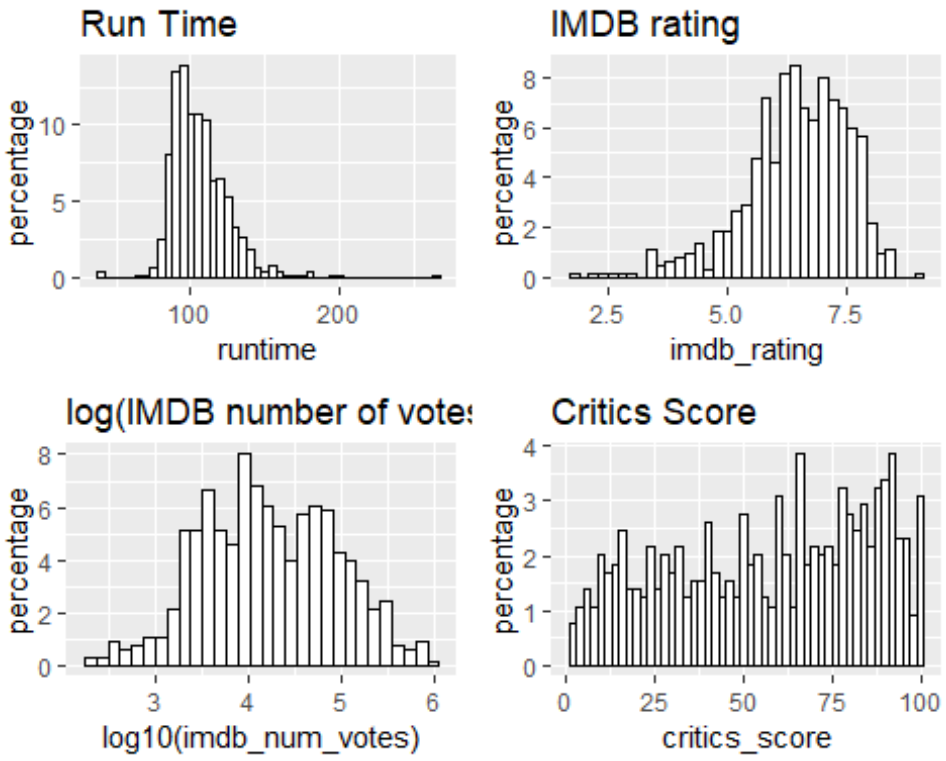
```
summary(train$audience_score)
```

```
##    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

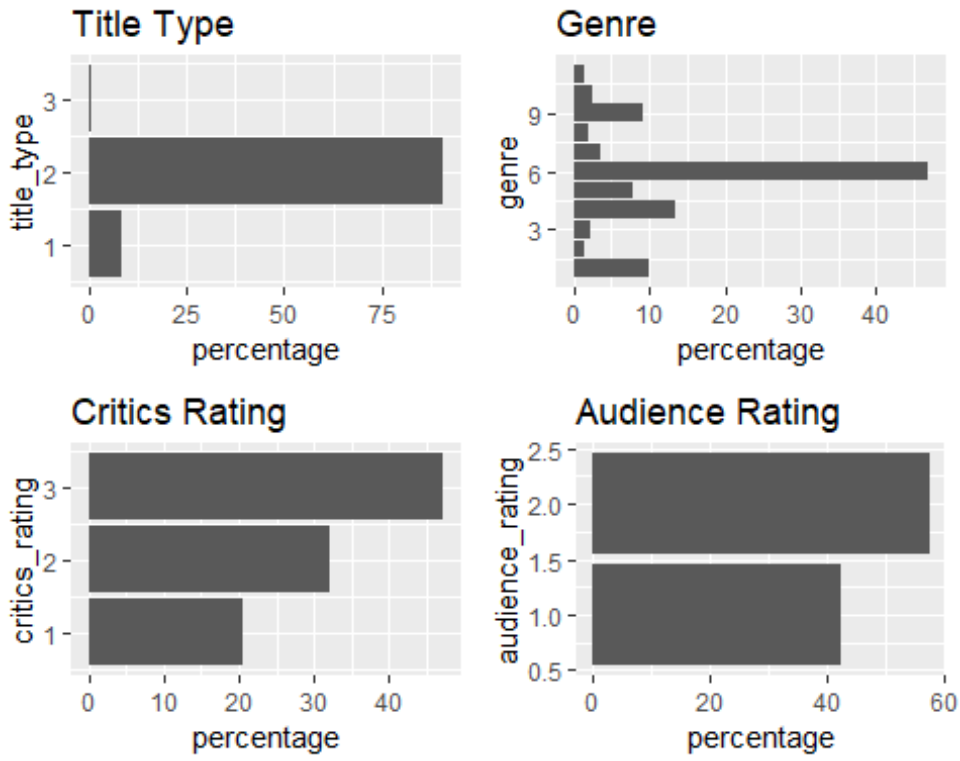
```
p1 <- ggplot(aes(x=runtime), data=train) +
  geom_histogram(aes(y=100*(..count..)/sum(..count..)), color='black',
fill='white', binwidth = 5) + ylab('percentage') + ggtitle('Run Time')
p2 <- ggplot(aes(x=imdb_rating), data=train) +
  geom_histogram(aes(y=100*(..count..)/sum(..count..)), color='black',
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`.
```



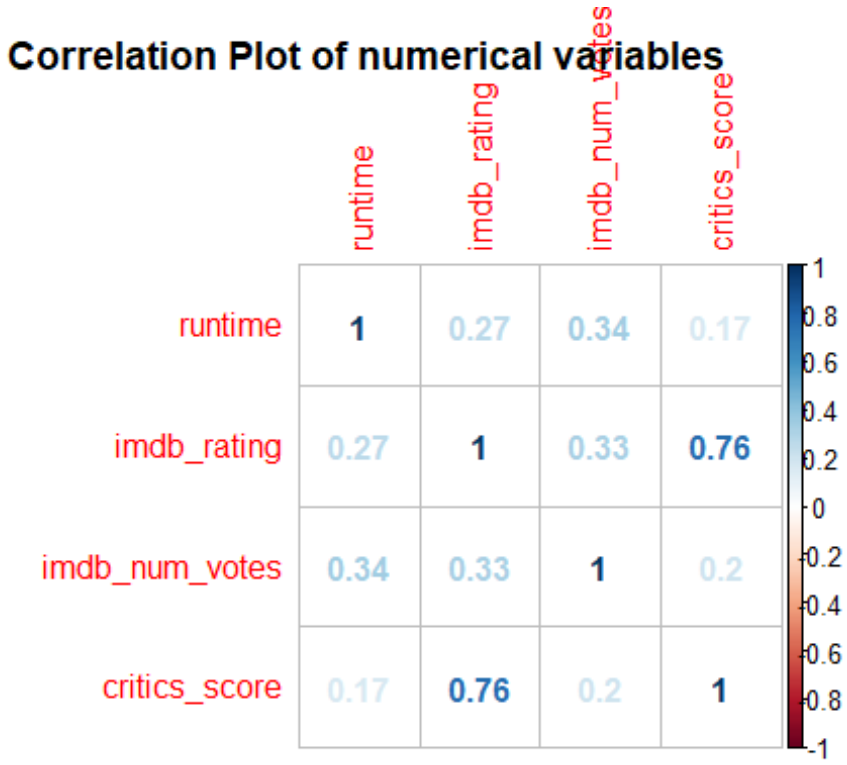
#Regression analysis: Run time, IMDB rating, log(IMDB number of votes) and Critics Scores all have reasonable broad distribution, therefore, they will be considered for the regression analysis.

```
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)
```

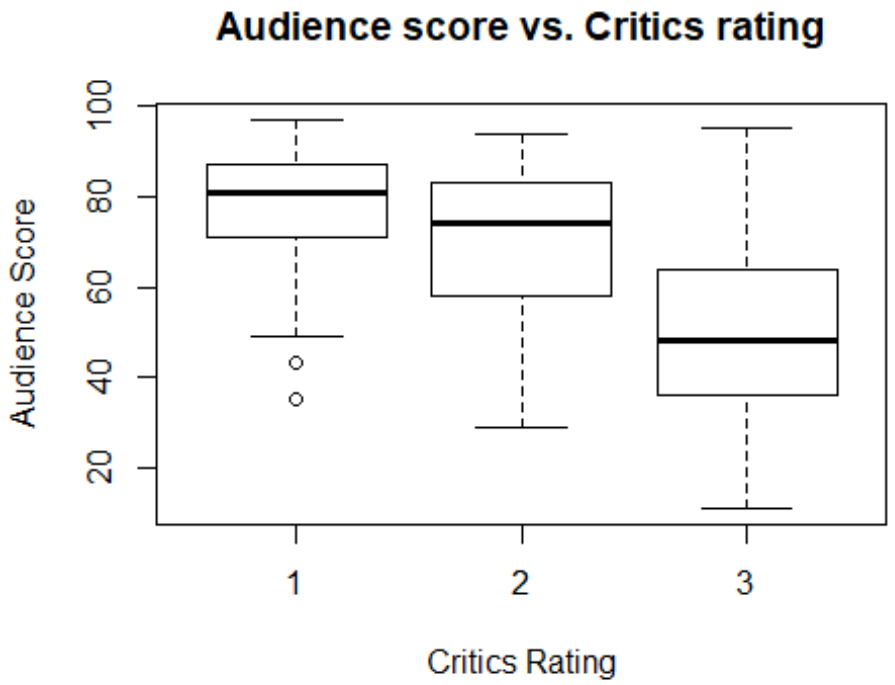


#Most movies in the data are in the "Feature Film" title type and majority of the movies are drama. Therefore, we must be aware that the results could be biased toward drama movies.

```
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")
```



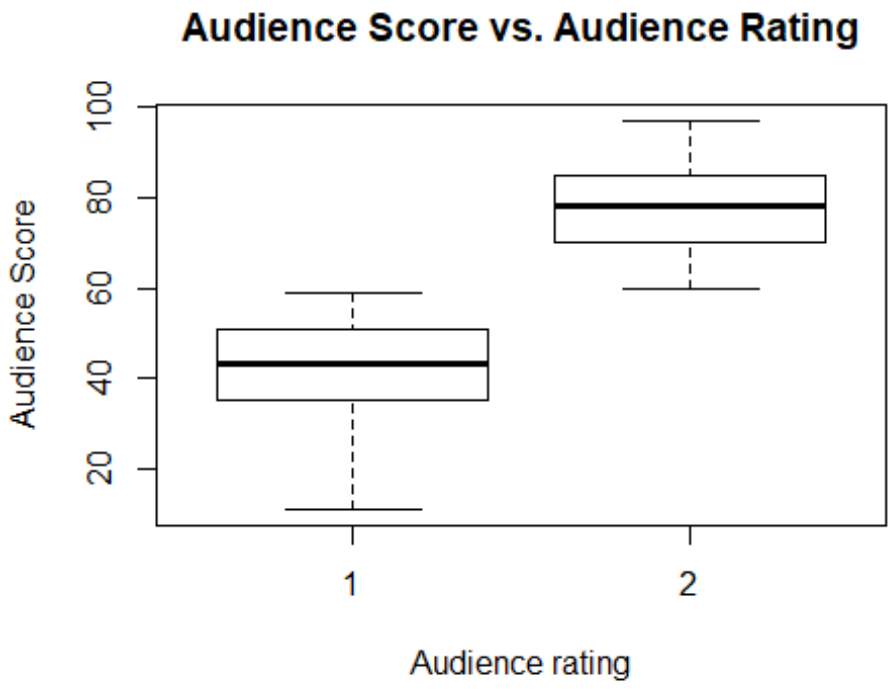

```
boxplot(audience_score~critics_rating, data=train, main='Audience score vs.
Critics rating', xlab='Critics Rating', ylab='Audience Score')
```



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

```
## train$critics_rating: 1
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   35.00   71.00   81.00   79.26   87.00   97.00
## ------------------------------------------------------------
## train$critics_rating: 2
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   29.00   58.00   74.00   69.96   83.00   94.00
## ------------------------------------------------------------
## train$critics_rating: 3
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    11.0    36.0    48.0    49.7    64.0    95.0
```

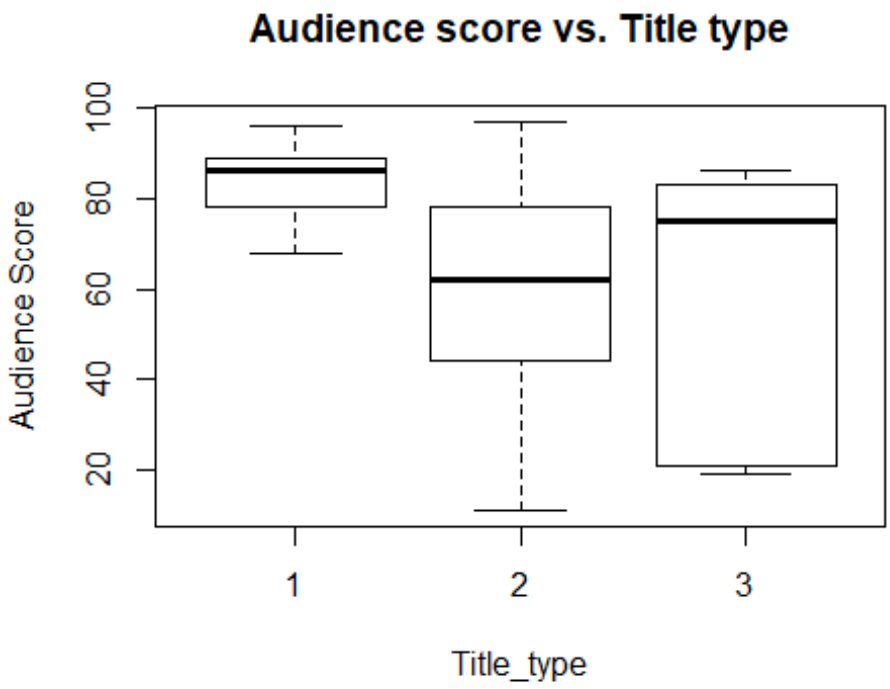
```
boxplot(audience_score~audience_rating, data=train, main='Audience Score vs. Audience Rating', xlab='Audience rating', ylab='Audience Score')
```




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

```
## train$audience_rating: 1
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11.00   35.00   43.00   41.93   51.00   59.00
## ------------------------------------------------------------
## train$audience_rating: 2
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   60.00   70.00   78.00   77.27   85.00   97.00
```

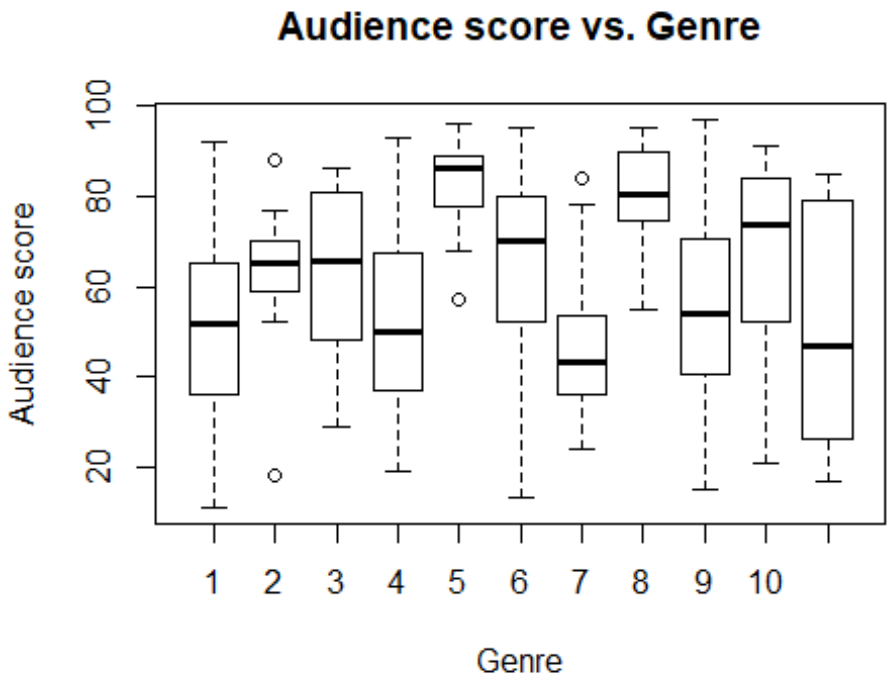
```r
boxplot(audience_score~title_type, data=train, main='Audience score vs. Title type', xlab='Title_type', ylab='Audience Score')
```




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

```
## train$title_type: 1
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   68.00   78.00   86.00   83.46   89.00   96.00
## ------------------------------------------------------------
## train$title_type: 2
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11.00   44.25   62.00   60.41   78.00   97.00
## ------------------------------------------------------------
## train$title_type: 3
##    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$genre: 1
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11.00   36.50   51.50   53.16   65.00   92.00
## ------------------------------------------------------------
## train$genre: 2
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   59.00   65.00   62.44   70.00   88.00
## ------------------------------------------------------------
## train$genre: 3
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   29.00   51.25   65.50   64.00   80.25   86.00
## ------------------------------------------------------------
## train$genre: 4
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   19.00   37.00   50.00   52.51   67.50   93.00
## ------------------------------------------------------------
## train$genre: 5
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   57.00   77.50   86.00   82.96   89.00   96.00
## ------------------------------------------------------------
## train$genre: 6
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   13.00   52.00   70.00   65.35   80.00   95.00
## ------------------------------------------------------------
## train$genre: 7
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   24.00   36.00   43.00   45.83   53.50   84.00
## ------------------------------------------------------------
## train$genre: 8
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   55.00   75.75   80.50   80.17   89.50   95.00
## ------------------------------------------------------------
## train$genre: 9
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15.00   40.50   54.00   55.95   70.50   97.00
## ------------------------------------------------------------
## train$genre: 10
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   21.00   53.00   73.50   66.69   82.50   91.00
## ------------------------------------------------------------
## train$genre: 11
##    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.

```
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 <- cor(movies_new[2:14])
movies_pca <- prcomp(movies_new,scale=TRUE)
str(movies_new)
```

```
##  num [1:13, 1:13] 1 0.0608 0.113 -0.3212 0.1209 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:13] "title_type" "genre" "runtime" "imdb_rating" ...
##   ..$ : chr [1:13] "title_type" "genre" "runtime" "imdb_rating" ...
```

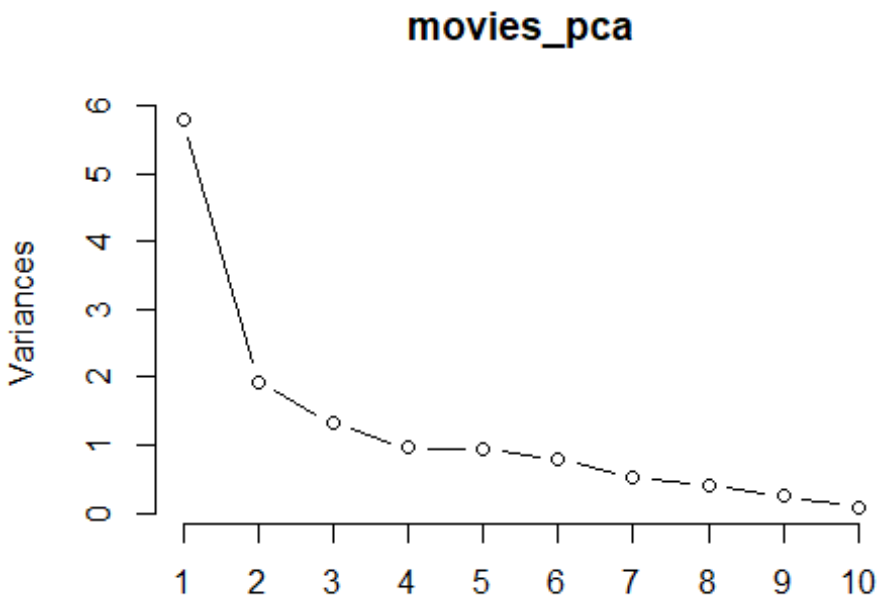
```
summary(movies_pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.408 1.3912 1.1453 0.97711 0.96805 0.88458 0.72389
## Proportion of Variance 0.446 0.1489 0.1009 0.07344 0.07209 0.06019 0.04031
## Cumulative Proportion  0.446 0.5949 0.6958 0.76924 0.84132 0.90152 0.94182
##                           PC8     PC9    PC10    PC11    PC12      PC13
## Standard deviation     0.64149 0.49241 0.27999 0.14785 0.04536 2.029e-17
## Proportion of Variance 0.03165 0.01865 0.00603 0.00168 0.00016 0.000e+00
## Cumulative Proportion  0.97348 0.99213 0.99816 0.99984 1.00000 1.000e+00

#movies_pca$x
movies_pca$rotation

##                           PC1         PC2         PC3         PC4
## title_type       -0.32269503  0.16569082  0.00762755  0.07160147
## genre             0.04310016 -0.14884743  0.40959301  0.78400436
## runtime           0.12008106  0.43703631  0.41396795 -0.10221820
## imdb_rating       0.40879921 -0.08153826  0.03391444 -0.04209855
## imdb_num_votes    0.23280253  0.39052376 -0.03142011 -0.02624040
## critics_rating   -0.39989770 -0.04763634 -0.07801802 -0.07587005
## critics_score     0.40287506 -0.05311574  0.07044580  0.04430504
## audience_rating   0.39159825 -0.15609335 -0.05209685 -0.09476602
## audience_score    0.40387951 -0.12534228 -0.02784244 -0.08589222
## best_pic_win      0.10049013  0.50810179 -0.32760853  0.17924451
## best_actor_win   -0.06540099  0.14026877  0.59959226 -0.49019822
## best_actress_win -0.03611079  0.21856368  0.35021999  0.24038694
## best_dir_win      0.06247517  0.47901336 -0.23374676  0.11408303
##                           PC5           PC6         PC7          PC8
## title_type       -0.029025485  0.4162739801  0.01774850  0.51653145
## genre            -0.360600526  0.0185128933 -0.15853725 -0.02708045
## runtime          -0.125209527  0.3030350675  0.46537975 -0.50361627
## imdb_rating       0.009515376 -0.0031518082  0.04416907  0.01586742
## imdb_num_votes    0.045881844  0.5578095682 -0.25858029  0.25634424
## critics_rating   -0.008756563  0.0007332718  0.08362822 -0.26251140
## critics_score    -0.009920595 -0.0783633052 -0.00368331  0.13927328
## audience_rating   0.059492939  0.0326932014  0.05748864  0.04286068
## audience_score    0.034679182  0.0119591729  0.04233168  0.04049258
## best_pic_win      0.057484522 -0.1759953771 -0.55336582 -0.37403981
## best_actor_win   -0.239867169 -0.3161240765 -0.41419767  0.17777832
## best_actress_win  0.810854818 -0.2537922760  0.11764874  0.14524478
## best_dir_win     -0.357782413 -0.4697967748  0.43062643  0.35843478
##                           PC9        PC10        PC11         PC12
## title_type        0.609588297 -0.11465573  0.17200410  0.015128057
## genre            -0.080012638 -0.17041941  0.02631126 -0.010669461
## runtime           0.175141815  0.02540983 -0.05418366 -0.020862999
## imdb_rating       0.030334445  0.04925575  0.76774969  0.460131010
## imdb_num_votes   -0.578047291 -0.03061876 -0.03371824  0.006050424
## critics_rating   -0.222581327 -0.32914731  0.32309040  0.052112713
## critics_score     0.203730186  0.52445370 -0.15781325  0.009988180
## audience_rating   0.171590033 -0.65559527 -0.41486269  0.395033321
```

```
## audience_score    0.096642276 -0.31574270  0.26284467 -0.792766840
## best_pic_win      0.317076348 -0.08081960  0.05810095  0.006167972
## best_actor_win    0.008586551 -0.11645863  0.02720766  0.006469670
## best_actress_win -0.057200308 -0.08913389  0.03536160 -0.003956082
## best_dir_win     -0.150174106 -0.11346573  0.01878072  0.004821782
##                         PC13
## title_type         0.08616940
## genre              0.03261523
## runtime           -0.01884143
## imdb_rating       -0.12846261
## imdb_num_votes     0.09765190
## critics_rating     0.69642147
## critics_score      0.67877284
## audience_rating    0.11588825
## audience_score     0.04050832
## best_pic_win       0.03737670
## best_actor_win     0.03447072
## best_actress_win   0.03420331
## best_dir_win       0.02357961

print(movies_pca)

## Standard deviations (1, .., p=13):
##  [1] 2.407948e+00 1.391189e+00 1.145311e+00 9.771066e-01 9.680529e-01
##  [6] 8.845760e-01 7.238888e-01 6.414916e-01 4.924064e-01 2.799901e-01
## [11] 1.478504e-01 4.535746e-02 2.029175e-17
##
## Rotation (n x k) = (13 x 13):
##                            PC1         PC2         PC3         PC4
## title_type       -0.32269503  0.16569082  0.00762755  0.07160147
## genre             0.04310016 -0.14884743  0.40959301  0.78400436
## runtime           0.12008106  0.43703631  0.41396795 -0.10221820
## imdb_rating       0.40879921 -0.08153826  0.03391444 -0.04209855
## imdb_num_votes    0.23280253  0.39052376 -0.03142011 -0.02624040
## critics_rating   -0.39989770 -0.04763634 -0.07801802 -0.07587005
## critics_score     0.40287506 -0.05311574  0.07044580  0.04430504
## audience_rating   0.39159825 -0.15609335 -0.05209685 -0.09476602
## audience_score    0.40387951 -0.12534228 -0.02784244 -0.08589222
## best_pic_win      0.10049013  0.50810179 -0.32760853  0.17924451
## best_actor_win   -0.06540099  0.14026877  0.59959226 -0.49019822
## best_actress_win -0.03611079  0.21856368  0.35021999  0.24038694
## best_dir_win      0.06247517  0.47901336 -0.23374676  0.11408303
##                           PC5           PC6         PC7         PC8
## title_type       -0.029025485  0.4162739801  0.01774850  0.51653145
## genre            -0.360600526  0.0185128933 -0.15853725 -0.02708045
## runtime          -0.125209527  0.3030350675  0.46537975 -0.50361627
## imdb_rating       0.009515376 -0.0031518082  0.04416907  0.01586742
## imdb_num_votes    0.045881844  0.5578095682 -0.25858029  0.25634424
## critics_rating   -0.008756563  0.0007332718  0.08362822 -0.26251140
## critics_score    -0.009920595 -0.0783633052 -0.00368331  0.13927328
```

```
## audience_rating   0.059492939  0.0326932014  0.05748864  0.04286068
## audience_score    0.034679182  0.0119591729  0.04233168  0.04049258
## best_pic_win      0.057484522 -0.1759953771 -0.55336582 -0.37403981
## best_actor_win   -0.239867169 -0.3161240765 -0.41419767  0.17777832
## best_actress_win  0.810854818 -0.2537922760  0.11764874  0.14524478
## best_dir_win     -0.357782413 -0.4697967748  0.43062643  0.35843478
##                          PC9         PC10         PC11         PC12
## title_type        0.609588297 -0.11465573  0.17200410  0.015128057
## genre            -0.080012638 -0.17041941  0.02631126 -0.010669461
## runtime           0.175141815  0.02540983 -0.05418366 -0.020862999
## imdb_rating       0.030334445  0.04925575  0.76774969  0.460131010
## imdb_num_votes   -0.578047291 -0.03061876 -0.03371824  0.006050424
## critics_rating   -0.222581327 -0.32914731  0.32309040  0.052112713
## critics_score     0.203730186  0.52445370 -0.15781325  0.009988180
## audience_rating   0.171590033 -0.65559527 -0.41486269  0.395033321
## audience_score    0.096642276 -0.31574270  0.26284467 -0.792766840
## best_pic_win      0.317076348 -0.08081960  0.05810095  0.006167972
## best_actor_win    0.008586551 -0.11645863  0.02720766  0.006469670
## best_actress_win -0.057200308 -0.08913389  0.03536160 -0.003956082
## best_dir_win     -0.150174106 -0.11346573  0.01878072  0.004821782
##                         PC13
## title_type        0.08616940
## genre             0.03261523
## runtime          -0.01884143
## imdb_rating      -0.12846261
## imdb_num_votes    0.09765190
## critics_rating    0.69642147
## critics_score     0.67877284
## audience_rating   0.11588825
## audience_score    0.04050832
## best_pic_win      0.03737670
## best_actor_win    0.03447072
## best_actress_win  0.03420331
## best_dir_win      0.02357961
```

```r
plot(movies_pca, type='l')
```




```
(movies_pca_eigens <- movies_pca$sdev^2)
```

```
##  [1] 5.798216e+00 1.935407e+00 1.311737e+00 9.547373e-01 9.371265e-01
##  [6] 7.824747e-01 5.240150e-01 4.115115e-01 2.424641e-01 7.839445e-02
## [11] 2.185975e-02 2.057299e-03 4.117550e-34
```

```
names(movies_pca_eigens) <- paste("PC",1:8,sep="")
sumlambdas <- sum(movies_pca_eigens)
sumlambdas
```

```
## [1] 13
```

```
dim(movies_new)
```

```
## [1] 13 13
```

```
#corr.matrix
movies_pca_new <- prcomp(corr.matrix, scale = TRUE)
summary(movies_pca_new)
```

```
## Importance of components:
##                           PC1    PC2     PC3       PC4
## Standard deviation     1.6686 1.0321 0.38787 4.602e-17
## Proportion of Variance 0.6961 0.2663 0.03761 0.000e+00
## Cumulative Proportion  0.6961 0.9624 1.00000 1.000e+00
```

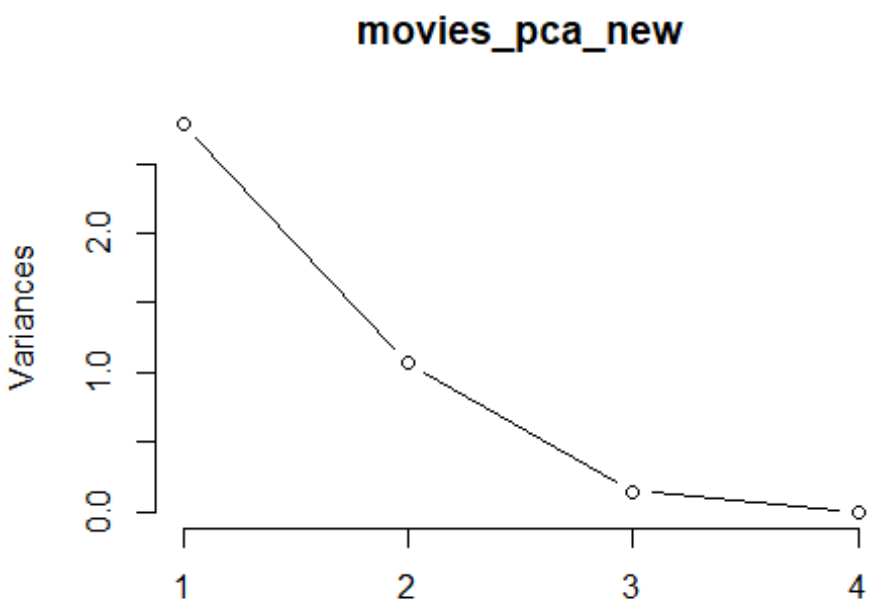
```
movies_pca_new$rotation
```

```
##                       PC1         PC2        PC3        PC4
## runtime        -0.4448468 -0.64136349 -0.2684187 0.56454904
## imdb_rating     0.5681143  0.02661214 -0.8176770 0.08911907
## imdb_num_votes -0.3647911  0.76594319 -0.1740329 0.49997106
## critics_score   0.5884598 -0.03571694  0.4786108 0.65066974
```

```
print(movies_pca_new)
```

```
## Standard deviations (1, .., p=4):
## [1] 1.668638e+00 1.032088e+00 3.878682e-01 4.602008e-17
##
## Rotation (n x k) = (4 x 4):
##                       PC1         PC2        PC3        PC4
## runtime        -0.4448468 -0.64136349 -0.2684187 0.56454904
## imdb_rating     0.5681143  0.02661214 -0.8176770 0.08911907
## imdb_num_votes -0.3647911  0.76594319 -0.1740329 0.49997106
## critics_score   0.5884598 -0.03571694  0.4786108 0.65066974
```

```
plot(movies_pca_new, type='l')
```

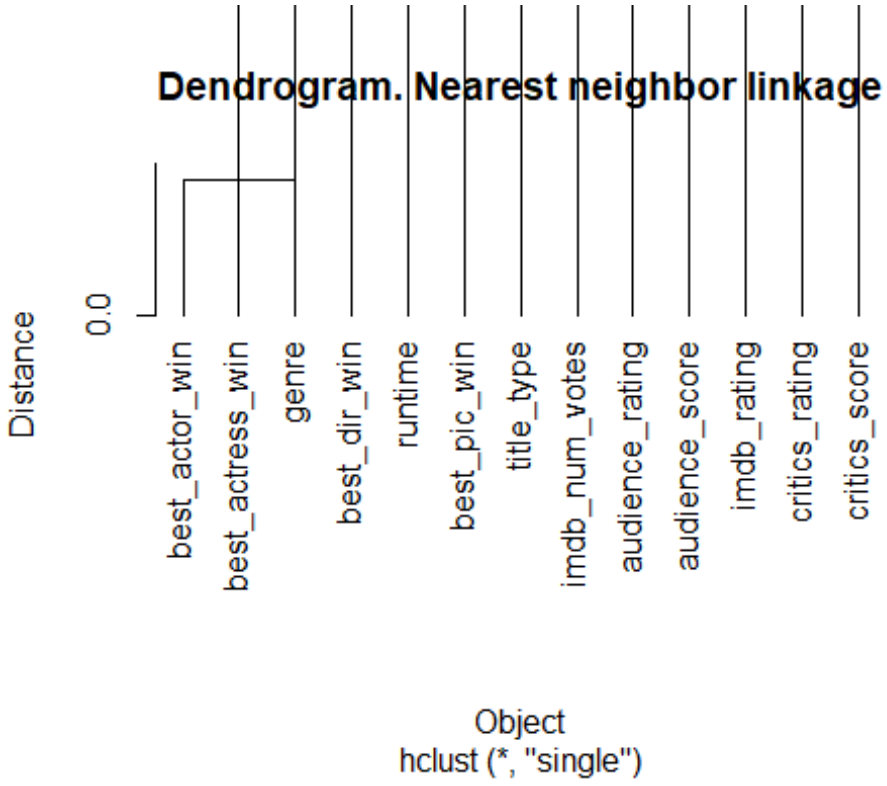



```
(movies_pca_eigens_new <- movies_pca_new$sdev^2)
```

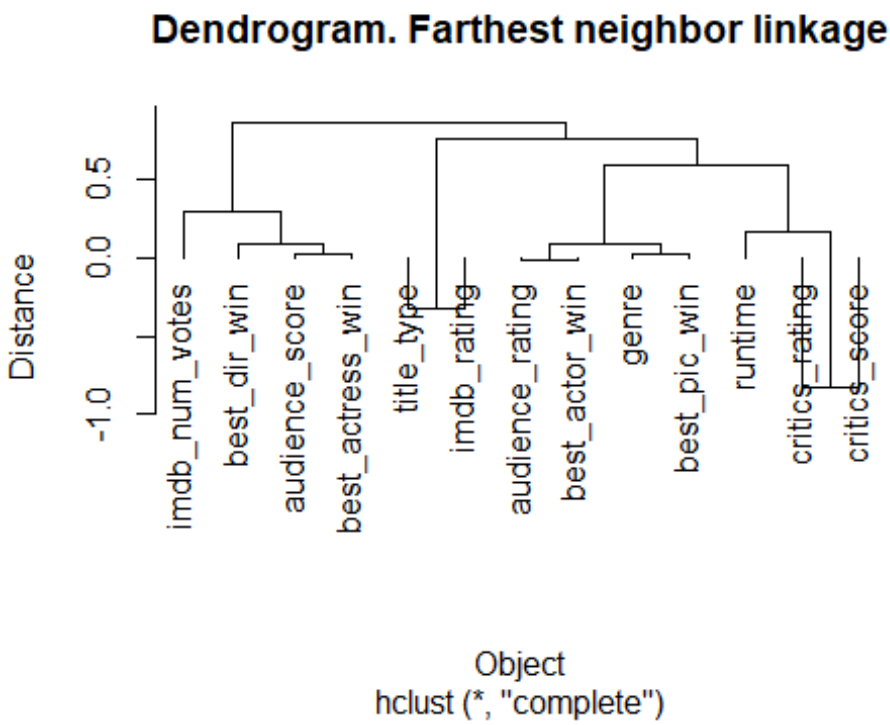
```
## [1] 2.784354e+00 1.065205e+00 1.504417e-01 2.117848e-33
```

```
names(movies_pca_eigens_new) <- paste("PC",1:2,sep="")
sumlambdas <- sum(movies_pca_eigens_new)
sumlambdas
```

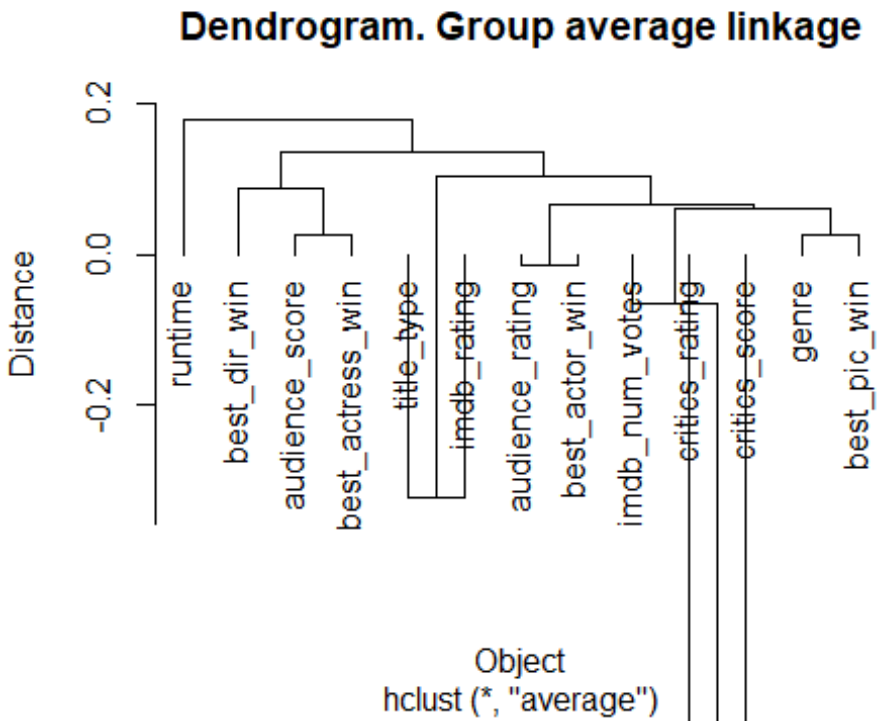
```
## [1] 4

dim(corr.matrix)

## [1] 4 4

colnames(movies_new) <- rownames(movies_new)
movies_new <- as.dist(movies_new)
mat5.nn <- hclust(movies_new, method = "single")
plot(mat5.nn, hang=-1,xlab="Object",ylab="Distance",
main="Dendrogram. Nearest neighbor linkage")
```




```
#Default - Complete
mat5.fn <- hclust(movies_new)
plot(mat5.fn,hang=-1,xlab="Object",ylab="Distance",
main="Dendrogram. Farthest neighbor linkage")
```




```
#Average
mat5.avl <- hclust(movies_new,method="average")
plot(mat5.avl,hang=-1,xlab="Object",ylab="Distance",
main="Dendrogram. Group average linkage")
```




```
# Standardizing the data with scale()
matstd.movies_new <- scale(movies_new[2:14])
# K-means, k=2, 3, 4, 5, 6
# Centers (k's) are numbers thus, 10 random sets are chosen

(kmeans2.movies_new <- kmeans(matstd.movies_new,2,nstart = 10))
```

```
## K-means clustering with 2 clusters of sizes 4, 9
##
## Cluster means:
##         [,1]
## 1 -1.3986451
## 2  0.6216201
##
## Clustering vector:
##  [1] 2 1 2 2 1 1 1 2 2 2 2 2 2
##
## Within cluster sum of squares by cluster:
## [1] 0.09262216 0.60484152
##  (between_SS / total_SS =  94.2 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```r
# Computing the percentage of variation accounted for. Two clusters
perc.var.2 <- round(100*(1 -
kmeans2.movies_new$betweenss/kmeans2.movies_new$totss),1)
names(perc.var.2) <- "Perc. 2 clus"
perc.var.2
```

```
## Perc. 2 clus
##          5.8
```

```r
# Computing the percentage of variation accounted for. Three clusters
(kmeans3.movies_new <- kmeans(matstd.movies_new,3,nstart = 10))
```

```
## K-means clustering with 3 clusters of sizes 4, 3, 6
##
## Cluster means:
##         [,1]
## 1 -1.3986451
## 2  0.9159192
## 3  0.4744705
##
## Clustering vector:
##  [1] 3 1 3 2 1 1 1 3 3 3 3 2 2
##
## Within cluster sum of squares by cluster:
## [1] 0.09262216 0.07224741 0.14284023
##  (between_SS / total_SS =  97.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```r
perc.var.3 <- round(100*(1 -
kmeans3.movies_new$betweenss/kmeans3.movies_new$totss),1)
names(perc.var.3) <- "Perc. 3 clus"
perc.var.3
```

```
## Perc. 3 clus
##          2.6
```

```r
# Computing the percentage of variation accounted for. Four clusters
(kmeans4.movies_new <- kmeans(matstd.movies_new,4,nstart = 10))
```

```
## K-means clustering with 4 clusters of sizes 1, 4, 4, 4
##
## Cluster means:
##         [,1]
## 1  1.1351204
## 2  0.3895597
## 3  0.7253054
```

```
## 4 -1.3986451
##
## Clustering vector:
##  [1] 3 4 3 1 4 4 4 2 2 2 2 3 3
##
## Within cluster sum of squares by cluster:
## [1] 0.00000000 0.05555089 0.02719733 0.09262216
##  (between_SS / total_SS =  98.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
perc.var.4 <- round(100*(1 -
kmeans4.movies_new$betweenss/kmeans4.movies_new$totss),1)
names(perc.var.4) <- "Perc. 4 clus"
perc.var.4
```

```
## Perc. 4 clus
##          1.5
```

```
# Computing the percentage of variation accounted for. Five clusters
(kmeans5.movies_new <- kmeans(matstd.movies_new,5,nstart = 10))
```

```
## K-means clustering with 5 clusters of sizes 3, 4, 1, 1, 4
##
## Cluster means:
##         [,1]
## 1 -1.4848499
## 2  0.3895597
## 3 -1.1400308
## 4  1.1351204
## 5  0.7253054
##
## Clustering vector:
##  [1] 5 1 5 4 1 3 1 2 2 2 2 5 5
##
## Within cluster sum of squares by cluster:
## [1] 0.003447006 0.055550893 0.000000000 0.000000000 0.027197330
##  (between_SS / total_SS =  99.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

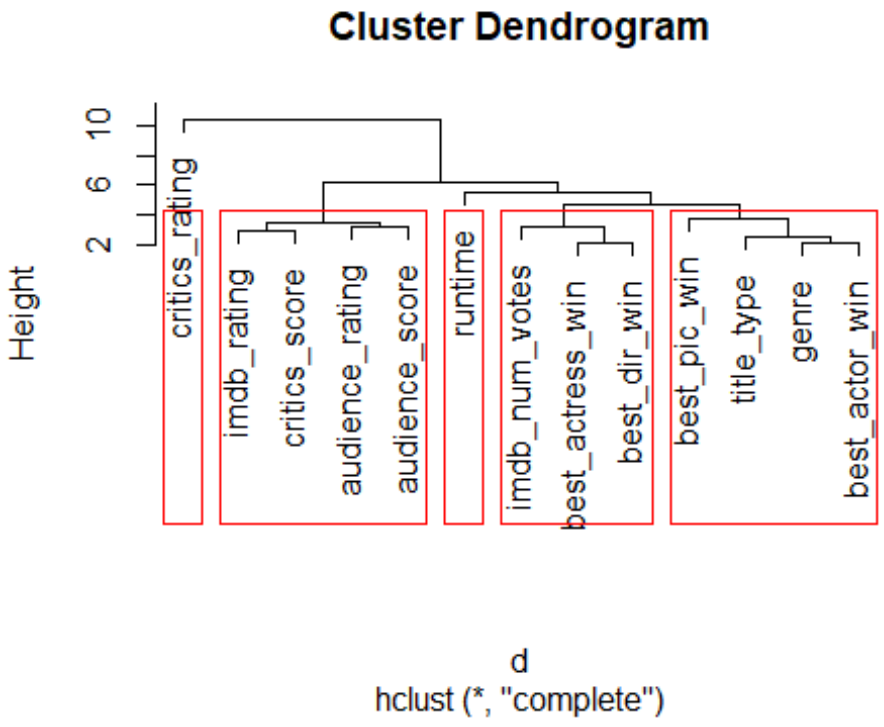
```
perc.var.5 <- round(100*(1 -
kmeans5.movies_new$betweenss/kmeans5.movies_new$totss),1)
```

```r
names(perc.var.5) <- "Perc. 5 clus"
perc.var.5

## Perc. 5 clus
##          0.7

(kmeans6.movies_new <- kmeans(matstd.movies_new,6,nstart = 10))

## K-means clustering with 6 clusters of sizes 3, 3, 1, 1, 4, 1
##
## Cluster means:
##         [,1]
## 1 -1.4848499
## 2  0.4534757
## 3  0.1978115
## 4 -1.1400308
## 5  0.7253054
## 6  1.1351204
##
## Clustering vector:
##  [1] 5 1 5 6 1 4 1 3 2 2 2 5 5
##
## Within cluster sum of squares by cluster:
## [1] 0.003447006 0.006527777 0.000000000 0.000000000 0.027197330
0.000000000
##  (between_SS / total_SS =  99.7 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"

# Computing the percentage of variation accounted for. Six clusters
perc.var.6 <- round(100*(1 -
kmeans6.movies_new$betweenss/kmeans6.movies_new$totss),1)
names(perc.var.6) <- "Perc. 6 clus"
perc.var.6

## Perc. 6 clus
##          0.3

#


movies_new <- scale(movies_new)
wss <- (nrow(movies_new)-1)*sum(apply(movies_new,2,var))
for (i in 1:5) wss[i] <- sum(kmeans(movies_new,centers=i)$withinss)
fit <- kmeans(movies_new, 5)
aggregate(movies_new,by=list(fit$cluster),FUN=mean)
```

```
##   Group.1 title_type       genre    runtime imdb_rating imdb_num_votes
## 1       1  0.6552446  0.04226622  0.8162421 -0.03810044    -0.24881835
## 2       2  0.3780973 -0.26494004  0.1309401 -0.45607298    -0.03945204
## 3       3 -1.3828952  0.57741865  0.2009378  0.93213257     0.62112031
## 4       4  0.7568126  1.12858724 -1.3198094  0.17582146     1.06823469
## 5       5  1.2966450 -2.50530033 -2.4564286 -1.96575851    -2.64845272
##   critics_rating critics_score audience_rating audience_score best_pic_win
## 1      0.2629309   -0.01309940     -0.12930698    -0.09763822   1.20588787
## 2      0.7716620   -0.31970319     -0.47528871    -0.47882304  -0.62514504
## 3     -1.2794226    0.89219453      1.02579941     1.00897691   0.06420556
## 4      0.3842178    0.07548263     -0.01807412     0.00538053   0.70500306
## 5      0.8580318   -2.32614977     -1.79604772    -1.83308136  -2.07890877
##   best_actor_win best_actress_win best_dir_win
## 1      0.5170993       -0.1269452  -0.12663683
## 2     -0.2036942        0.5825442   0.31126960
## 3     -0.5191122       -0.4360077  -0.01235528
## 4      2.5810254        1.8312463   1.26181933
## 5     -1.2410975       -2.0365567  -2.07756614

mydata <- data.frame(movies_new, fit$cluster)
mydata

##                  title_type       genre     runtime imdb_rating
## title_type        0.1639182 -0.06215298 -0.42093850  -1.2489511
## genre             0.4831010 -0.86114424 -0.11474062  -0.1154920
## runtime           0.7568126  1.12858724 -1.31980935   0.1758215
## imdb_rating      -1.5214408  1.07925282  0.81411896  -0.4725742
## imdb_num_votes    0.7983436 -0.22189773  1.44238236   0.3303364
## critics_rating    1.2966450 -2.50530033 -2.45642864  -1.9657585
## critics_score    -1.4659827  1.37334280  0.05246795   1.3760734
## audience_rating  -1.1094020 -0.19839699 -0.18263708   1.2069316
## audience_score   -1.4347552  0.05547596  0.11980135   1.6180995
## best_pic_win      0.3054102 -0.49503085  0.28063492  -0.1448832
## best_actor_win    0.5599598  0.35856789  0.77880479  -0.3149656
## best_actress_win  0.6425379  0.42671005  0.34771772  -0.2986760
## best_dir_win      0.5248525 -0.07801366  0.65862615  -0.1459617
##                  imdb_num_votes critics_rating critics_score
## title_type           -0.1571934      1.5739081   -1.08058046
## genre                -0.5483535      0.4429313    0.06977534
## runtime               1.0682347      0.3842178    0.07548263
## imdb_rating           0.9867322     -1.1905156    1.49277724
## imdb_num_votes       -0.8118744     -0.2667625    0.16523268
## critics_rating       -2.6484527      0.8580318   -2.32614977
## critics_score         0.3252682     -1.8982366   -0.33729511
## audience_rating       0.4124688     -0.8903713    1.06559223
## audience_score        0.7600121     -1.1385670    1.34770375
## best_pic_win          0.9107319      0.2977674   -0.04151619
## best_actor_win       -0.3629931      0.7720413   -0.22649147
## best_actress_win     -0.1537126      0.6279862   -0.19064690
## best_dir_win          0.2191319      0.4275691   -0.01388397
```

```
##                   audience_rating audience_score best_pic_win
## title_type            -1.04110305    -1.13482403  -0.60536625
## genre                 -0.26345269    -0.25565727  -0.59853774
## runtime               -0.01807412     0.00538053   0.70500306
## imdb_rating            1.44603725     1.61121240   0.21147849
## imdb_num_votes         0.20256169     0.26207558   1.58462884
## critics_rating        -1.79604772    -1.83308136  -2.07890877
## critics_score          1.15806169     1.23378457   0.12152087
## audience_rating       -0.39731303     1.61041151  -0.13613298
## audience_score         1.89641173    -0.41950083   0.05995585
## best_pic_win          -0.16026498    -0.14850663  -0.80818282
## best_actor_win        -0.43633413    -0.37630423  -0.48849334
## best_actress_win      -0.33251676    -0.35690498   0.34371469
## best_dir_win          -0.25796588    -0.19808525   1.68932009
##                   best_actor_win best_actress_win best_dir_win fit.cluster
## title_type            0.09663716       0.18974833  -0.32547473           2
## genre                 0.32594753       0.28448204  -0.40632841           2
## runtime               2.58102537       1.83124630   1.26181933           4
## imdb_rating          -0.03888245      -0.07742750   0.25977475           3
## imdb_num_votes        0.19456281       0.61031855   0.74753935           1
## critics_rating       -1.24109754      -2.03655670  -2.07756614           5
## critics_score        -0.28818236      -0.22518295   0.26007161           3
## audience_rating      -1.09305950      -0.73621677  -0.46894130           3
## audience_score       -0.65632455      -0.70520373  -0.10032617           3
## best_pic_win         -0.33833093       1.04839758   1.99720881           2
## best_actor_win       -0.89903054       0.80754895  -0.02032727           2
## best_actress_win      0.89209531      -1.07482430  -0.19510588           1
## best_dir_win          0.46463970       0.08367019  -0.93234395           1
```

```r
d <- dist(mydata, method = "euclidean") # distance matrix
fit <- hclust(d, method="complete")

plot(fit)
# cut tree into 5 clusters
groups <- cutree(fit, k=5)
# draw dendogram with red borders around the 5 clusters
rect.hclust(fit, k=5, border="red")
```



Factor Analysis

```
#head(movies_data)

#Loading the required library
library(psych)

## Warning: package 'psych' was built under R version 3.5.3

##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha

#Applying Factor Analysis on the data with 4 factors
#fit_pc <- principal(movies_data,nfactors = 4, rotate = "varimax")

#Printing the results of Factor Analysis
#fit_pc

#rounding the values to 3 decimal places
#round(fit.pc$values, 3)

#Printing the loading data to console for the
#fit.pc$loadings
```

Now we look at the cummunality

```r
#fit.pc$communality

#Printing the scores
#fit.pc$scores

# See Correlations within Factors
#fa.plot(fit.pc)

#Visualize the relationship
#fa.diagram(fit.pc)

fit1 <- lm(audience_score~., data = train[,-1])
g1 <- step(fit1)

## Start:  AIC=2523.57
## audience_score ~ title_type + genre + runtime + imdb_rating +
##     imdb_num_votes + critics_rating + critics_score + audience_rating +
##     best_pic_win + best_actor_win + best_actress_win + best_dir_win
##
##                    Df Sum of Sq   RSS    AIC
## - best_pic_win      1         1 30449 2521.6
## - best_dir_win      1         1 30449 2521.6
## - best_actor_win    1         3 30451 2521.6
## - critics_rating    1         7 30454 2521.7
## - title_type        1        22 30469 2522.0
## - critics_score     1        46 30494 2522.6
## - imdb_num_votes    1        48 30496 2522.6
## - best_actress_win  1        69 30517 2523.0
## <none>                        30448 2523.6
## - runtime           1       152 30600 2524.8
## - genre             1       205 30653 2525.9
## - imdb_rating       1     17711 48159 2819.1
## - audience_rating   1     32608 63056 2994.1
##
## Step:  AIC=2521.59
## audience_score ~ title_type + genre + runtime + imdb_rating +
##     imdb_num_votes + critics_rating + critics_score + audience_rating +
##     best_actor_win + best_actress_win + best_dir_win
##
##                    Df Sum of Sq   RSS    AIC
## - best_dir_win      1         2 30450 2519.6
## - best_actor_win    1         3 30451 2519.7
## - critics_rating    1         7 30455 2519.7
## - title_type        1        22 30470 2520.1
## - critics_score     1        46 30495 2520.6
## - imdb_num_votes    1        53 30502 2520.7
## - best_actress_win  1        68 30517 2521.0
## <none>                        30449 2521.6
## - runtime           1       151 30600 2522.8
## - genre             1       205 30654 2523.9
```

```
## - imdb_rating       1     17717 48165 2817.2
## - audience_rating   1     32609 63058 2992.1
##
## Step:  AIC=2519.62
## audience_score ~ title_type + genre + runtime + imdb_rating +
##     imdb_num_votes + critics_rating + critics_score + audience_rating +
##     best_actor_win + best_actress_win
##
##                    Df Sum of Sq   RSS    AIC
## - best_actor_win    1         3 30453 2517.7
## - critics_rating    1         7 30457 2517.8
## - title_type        1        21 30471 2518.1
## - critics_score     1        47 30497 2518.6
## - imdb_num_votes    1        55 30505 2518.8
## - best_actress_win  1        68 30518 2519.1
## <none>                          30450 2519.6
## - runtime           1       150 30600 2520.8
## - genre             1       205 30655 2522.0
## - imdb_rating       1     17726 48176 2815.4
## - audience_rating   1     32665 63115 2990.7
##
## Step:  AIC=2517.69
## audience_score ~ title_type + genre + runtime + imdb_rating +
##     imdb_num_votes + critics_rating + critics_score + audience_rating +
##     best_actress_win
##
##                    Df Sum of Sq   RSS    AIC
## - critics_rating    1         7 30460 2515.8
## - title_type        1        21 30474 2516.1
## - critics_score     1        48 30501 2516.7
## - imdb_num_votes    1        55 30508 2516.8
## - best_actress_win  1        66 30519 2517.1
## <none>                          30453 2517.7
## - runtime           1       148 30601 2518.8
## - genre             1       204 30657 2520.0
## - imdb_rating       1     17754 48208 2813.8
## - audience_rating   1     32759 63213 2989.7
##
## Step:  AIC=2515.83
## audience_score ~ title_type + genre + runtime + imdb_rating +
##     imdb_num_votes + critics_score + audience_rating + best_actress_win
##
##                    Df Sum of Sq   RSS    AIC
## - title_type        1        20 30480 2514.3
## - best_actress_win  1        65 30525 2515.2
## - imdb_num_votes    1        77 30536 2515.5
## <none>                          30460 2515.8
## - runtime           1       155 30614 2517.1
## - critics_score     1       164 30624 2517.3
## - genre             1       205 30665 2518.2
```

```
## - imdb_rating       1     18166 48625 2817.4
## - audience_rating   1     33435 63895 2994.6
##
## Step:  AIC=2514.26
## audience_score ~ genre + runtime + imdb_rating + imdb_num_votes +
##     critics_score + audience_rating + best_actress_win
##
##                     Df Sum of Sq   RSS    AIC
## - imdb_num_votes     1        64 30544 2513.6
## - best_actress_win   1        71 30551 2513.8
## <none>                         30480 2514.3
## - runtime            1       172 30652 2515.9
## - critics_score      1       177 30658 2516.0
## - genre              1       221 30702 2517.0
## - imdb_rating        1     19195 49675 2829.2
## - audience_rating    1     33431 63911 2992.8
##
## Step:  AIC=2513.63
## audience_score ~ genre + runtime + imdb_rating + critics_score +
##     audience_rating + best_actress_win
##
##                     Df Sum of Sq   RSS    AIC
## - best_actress_win   1        63 30608 2513.0
## <none>                         30544 2513.6
## - runtime            1       130 30675 2514.4
## - critics_score      1       165 30709 2515.1
## - genre              1       226 30771 2516.4
## - imdb_rating        1     20291 50836 2842.2
## - audience_rating    1     33506 64050 2992.2
##
## Step:  AIC=2512.97
## audience_score ~ genre + runtime + imdb_rating + critics_score +
##     audience_rating
##
##                    Df Sum of Sq   RSS    AIC
## <none>                          30608 2513.0
## - critics_score     1       162 30769 2514.4
## - runtime           1       173 30781 2514.6
## - genre             1       242 30850 2516.1
## - imdb_rating       1     20277 50885 2840.9
## - audience_rating   1     33586 64194 2991.7
```

```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:psych':
##
##     logit

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

```
compareCoefs(fit1,g1,se=FALSE)
```

```
## Calls:
## 1: lm(formula = audience_score ~ ., data = train[, -1])
## 2: lm(formula = audience_score ~ genre + runtime + imdb_rating +
##   critics_score + audience_rating, data = train[, -1])
##
##                   Model 1 Model 2
## (Intercept)         -24.0   -27.4
## title_type         -0.701
## genre              -0.262  -0.282
## runtime           -0.0290 -0.0279
## imdb_rating          9.18    9.30
## imdb_num_votes   3.02e-06
## critics_rating      -0.25
## critics_score      0.0217  0.0276
## audience_rating      20.4    20.5
## best_pic_win        0.363
## best_actor_win        0.2
## best_actress_win    -1.08
## best_dir_win        0.171
```

```
fit_final <- lm(audience_score ~
genre+runtime+imdb_rating+critics_score+audience_rating, data=train[,-1])
summary(fit_final)
```

```
##
## Call:
## lm(formula = audience_score ~ genre + runtime + imdb_rating +
##     critics_score + audience_rating, data = train[, -1])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.0752  -4.7253   0.6766   4.3219  24.4640
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -27.43046    2.25795 -12.148   <2e-16 ***
## genre            -0.28172    0.12495  -2.255   0.0245 *
## runtime          -0.02790    0.01463  -1.907   0.0569 .
## imdb_rating       9.30480    0.45083  20.639   <2e-16 ***
## critics_score     0.02764    0.01500   1.842   0.0659 .
## audience_rating  20.47743    0.77092  26.562   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.899 on 643 degrees of freedom
## Multiple R-squared:  0.8844, Adjusted R-squared:  0.8835
## F-statistic: 983.6 on 5 and 643 DF,  p-value: < 2.2e-16

newmovie <- test %>% select(genre, imdb_rating,
audience_rating,critics_score,runtime)
predict(fit_final, newmovie)

##        1
## 90.28938

predict(fit_final, newmovie, interval = "prediction", level = 0.95)

##        fit      lwr      upr
## 1 90.28938 76.61211 103.9666

test$audience_score

## [1] 94
```