Modeling and prediction for movies

Author: Jorge Álvarez de la Fuente

Setup

The data set contains information from Rotten Tomatoes, a website that provides the world's most trusted recommendation resources for quality entertainment, and Internet Movie Database IMDB, a website that provides information about millions of films and television shows as well as their cast and crew.

Load packages

```
library(ggplot2)
```

```
## Warning in as.POSIXlt.POSIXct(Sys.time()): unable to identify current timezone 'H':
## please set environment variable 'TZ'
```

```
library(dplyr)
library(statsr)
```

Load data

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

Part 1: Data

The following code shows the number of observations and variables within movies data set:

```
#dimension of data
dim(movies)
```

```
## [1] 651  32
```

As shown above the data set "movies" consists of 651 observations (rows) and 32 variables (columns), which means the observations are less than 10% of entire population (number of movies released between 1997 and 2015) and therefore independence can be assumed. Each observation refers to a different movie released before 2016 and variables represent features and indicators for each movie within the "movies" data set (such as genre, ratings, release year, etc.). The data was collected using randomly sampled movies instead of randomly assigned movies, and constitutes an observational study. Conclusions then can only be generalized to the population of interest. Therefore, causation cannot be established because random assignment was not used to collect the data. Furthermore, a potential source of bias can be attributed to nonvoting and non rating given, resulting in a Non-response type of sampling bias.

Part 2: Research question

The following research question aims to find whether a movie reputation can be predicted based on certain explanatory variables contained in "movies" data set:

Can the popularity of a movie be predicted based on different rating measurements provided by a community of existing users?

Moghaddam, F. et al. 2019 in their study "predicting Movie Popularity and Ratings with Visual Features" found in their extensive experiments on a large data set of more than 13,000 movies trailers that an hybrid approach achieves promising results by exploiting visual attractiveness features of movies in comparison to the other baseline features.

The following exploratory data analysis consider an hybrid approach using a variety of predictors to measure the popularity of a given movie, aiming to learn what attributes make a movie popular as Paramount Pictures is interested to know about.

The explanatory variables / predictors used to perform the exploratory data analysis are the following:

1. imdb_rating
2. imdb_num_votes
3. critics_rating
4. critics_score
5. audicence_rating
6. audience_score

Part 3: Exploratory data analysis

The exploratory data analysis section address finding the correct predictors to measure the popularity of a movie.

The following code gets the data of the potential predictors: title, imdb_rating, imdb_num_votes, critics_rating, critics_score, audience_rating, audience_score to be used in the model:

```
#This code selects the explanatory variables mentioned above
#and stores the results into a new variable: movies_filtered
movies_filtered <- movies %>%
  select(title, imdb_rating, imdb_num_votes, critics_rating, critics_score, audience_rating, audience_s
```

The structure of the movies data set filtered with the explanatory variables mentioned in the code above is the following:

```
#This code shows the structure of the filtered movies data set
str(movies_filtered)
```

```
## tibble [651 x 7] (S3: tbl_df/tbl/data.frame)
##  $ title          : chr [1:651] "Filly Brown" "The Dish" "Waiting for Guffman" "The Age of Innocence
##  $ 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 ...
```

The summary statistics for the new movies data set filtered is presented below:

```
#This code gives the summary statistics for the movies data set filtered
summary(movies_filtered)
```

```
##     title              imdb_rating     imdb_num_votes          critics_rating
##  Length:651         Min.   :1.900   Min.   :   180   Certified Fresh:135
##  Class :character   1st Qu.:5.900   1st Qu.:  4546   Fresh          :209
##  Mode  :character   Median :6.600   Median : 15116   Rotten         :307
##                     Mean   :6.493   Mean   : 57533
##                     3rd Qu.:7.300   3rd Qu.: 58301
##                     Max.   :9.000   Max.   :893008
##  critics_score    audience_rating audience_score
##  Min.   :  1.00   Spilled:275     Min.   :11.00
##  1st Qu.: 33.00   Upright:376     1st Qu.:46.00
##  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
```

From the results above, it can be observed that the mean and median imdb_rating are both above 6 (and both are close to each other) out of a maximum value of 9. The median of imdb_num_votes differ significantly from the mean (15116 and 57533 respectively) as a result of a possible potential source of Non-response bias. Most of the critics_rating are classified as "Rotten". Median and mean of critics_score are both close to each other, above 50 out of 100. Audience_rating upright surpasses spilled category and audience_score mean and median are above 60 out of 97 and close to each other.

Figure 1 and 2 present the histogram for numerical variables: critics_score and audience_score. The summary statistics for both variables are also presented:

```
#This code gives the histogram for Critics Score
hist(movies_filtered$critics_score, main="Histogram for Critics Score",
     xlab = "Critics Score", border = "black", col = "darkseagreen4")
```

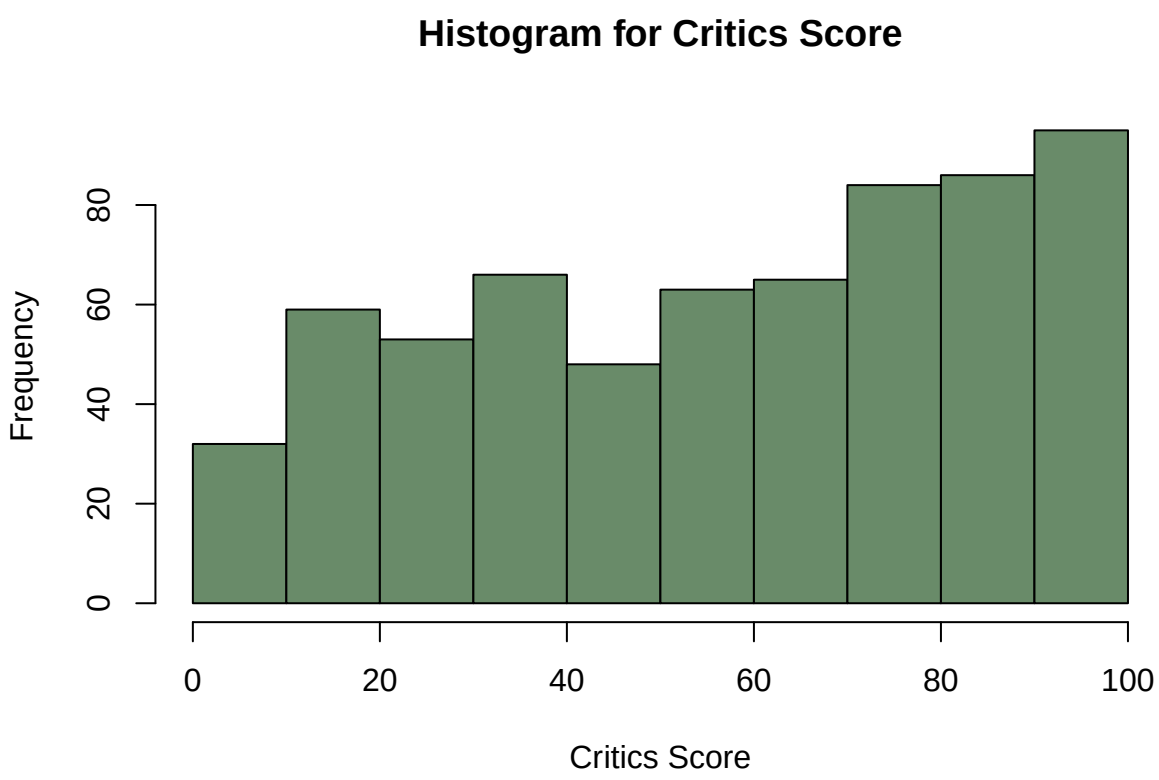

Figure 1: Histogram plot for Critics Score

```
#This code presents the summary statistics for Critics Score explanatory numerical variable:
summary(movies_filtered$critics_score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00   33.00   61.00   57.69   83.00  100.00
```

```
#This code gives the histogram for Audience Score
hist(movies_filtered$audience_score, main="Histogram for Audience Score",
     xlab = "Audience Score", border = "black", col = "darkseagreen4")
```

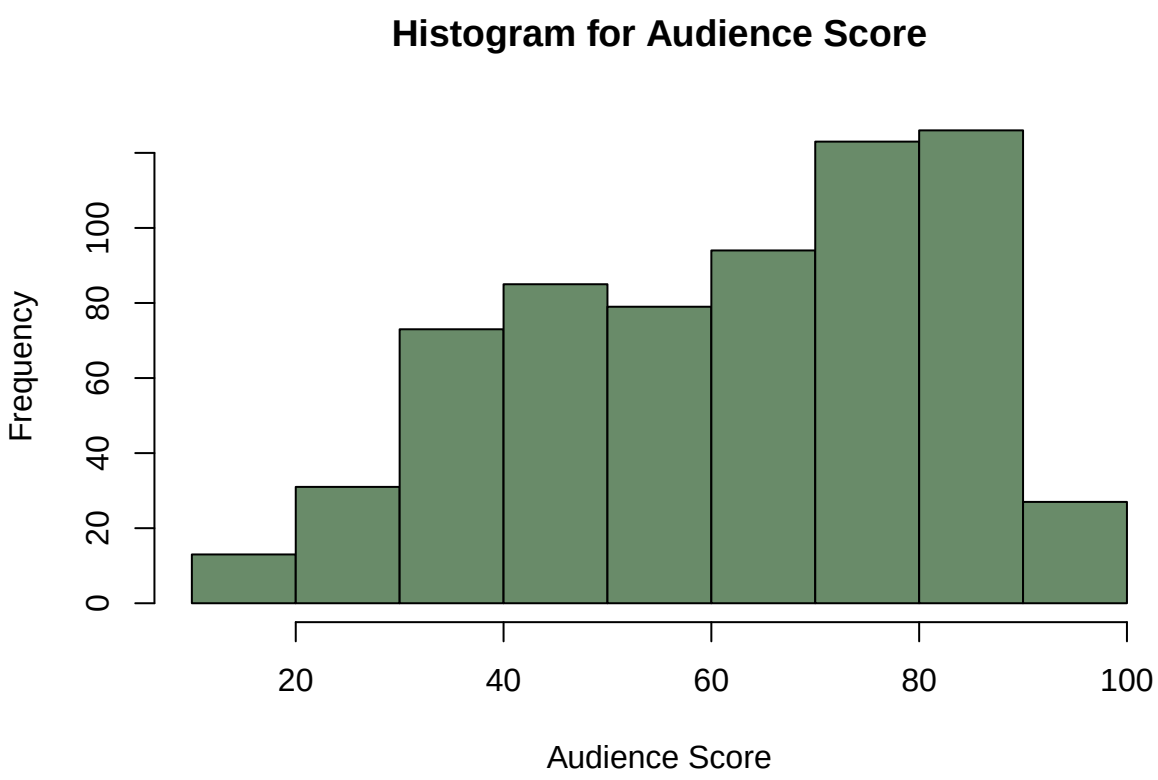

Figure 2: Histogram plot for Audience Score

```
#This code presents the summary statistics for the explanatory numerical variable Audience Score:
summary(movies_filtered$audience_score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11.00   46.00   65.00   62.36   80.00   97.00
```

The median of the response variable critics score is 61 out of 100, while the median of the response variable audience score is 65 out of 97, both are close to each other. The mean of the response variable critics score is 58 while the mean of the response variable audience score is 63. For both response variables, 25% of the scores are below 50 (33 for critics score and 46 for audience score respectively), while 25% of the scores are above 80 (83 for critics score and 80 for audience score respectively). Therefore, there are more scores above 50 than below 50.

Figure 3 explores whether there is a linear relationship between the explanatory variable imdb_rating and the response variable audience_score.

```
#This code plots the Audience Score and IMDB rating
ggplot(data = movies_filtered, aes(x = imdb_rating, y = audience_score)) +
  geom_point() + stat_smooth(method = lm, level = 0.99) +
  xlab("IMDb Rating") + ylab("Audience Score") +
  ggtitle("Relationship between Audience Score and IMDb Rating")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

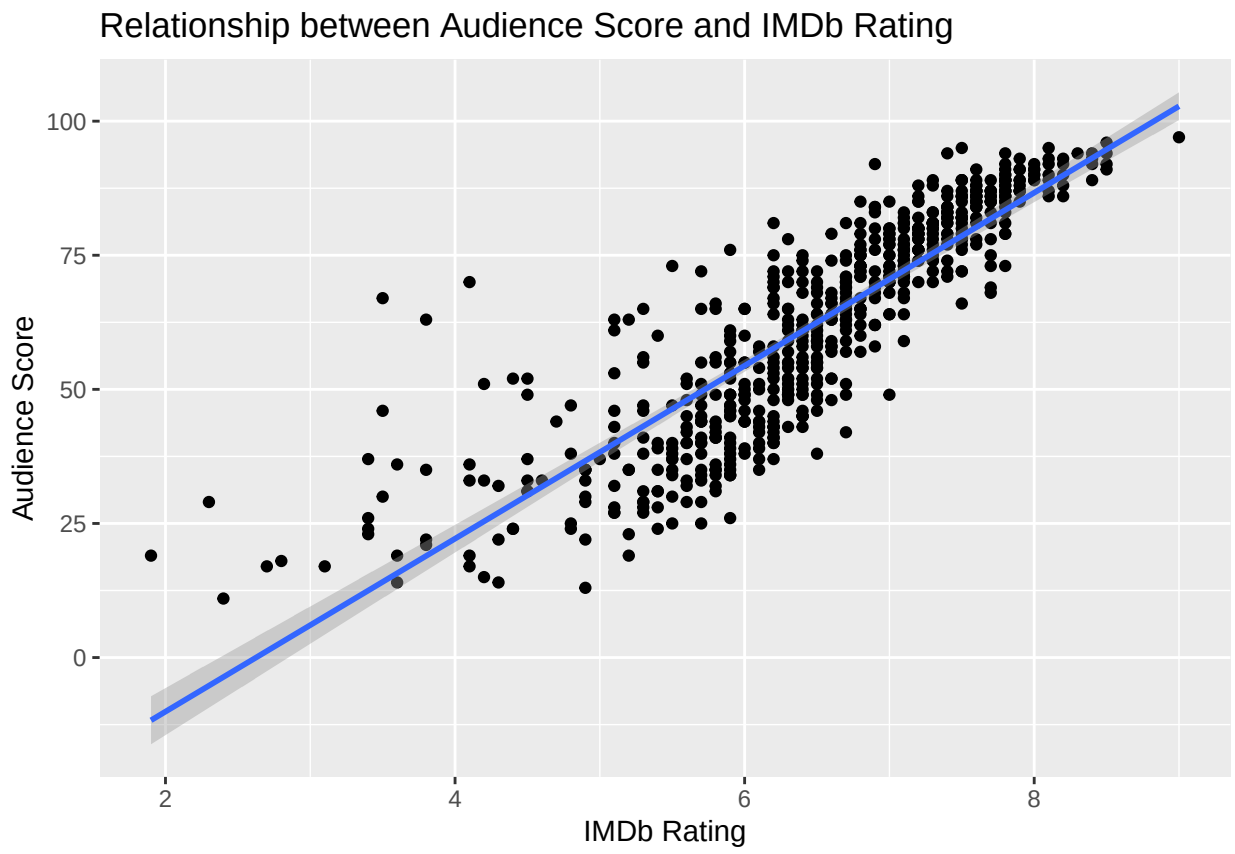

Figure 3: Scatter plot for Audience Score and IMDb Rating

There is a strong positive linear relationship between the explanatory/predictor variable IMDb Rating and the response variable Audience Score.

Figure 4 explores whether there is a linear relationship between the explanatory variable imdb_num_votes and the response variable critics_score.

```
#This code plots the Audience Score and IMDB number of votes
ggplot(data = movies_filtered, aes(x = imdb_num_votes, y = audience_score)) +
  geom_point() + stat_smooth(method = lm, level = 0.99) +
  xlab("IMDb number of votes") + ylab("Audience Score") +
  ggtitle("Relationship between Audience Score and IMDb number of votes")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

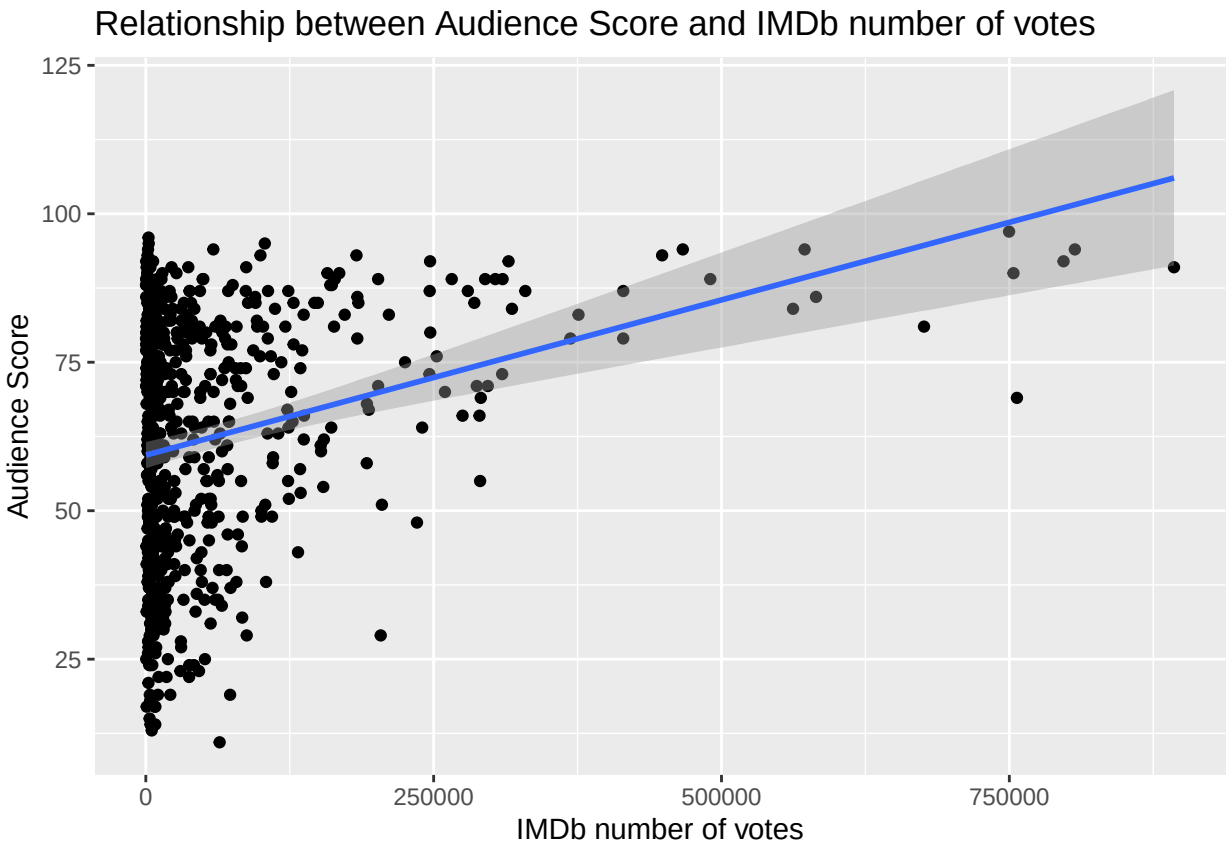

Figure 4: Scatter plot for Audience Score and IMDb number of votes

There is a weak but still positive relationship between the explanatory/predictor variable IMDb number of votes and the response variable Audience Score.

Figures 5 and 6 shows the correlation between categorical variables: critics_rating and audience_rating with the numerical variable audience_score.

```
#This code shows the boxplot for the predictor critics_rating and the response variable audience_score.
ggplot(data=movies_filtered, aes(x = critics_rating, y = audience_score, fill = critics_rating)) +
  geom_boxplot() +
  scale_fill_manual(values=c("coral1","cadetblue3","burlywood"), guide = guide_legend("Critics Rating"))
  labs(x = "Critics Rating", y = "Audience Score")
```

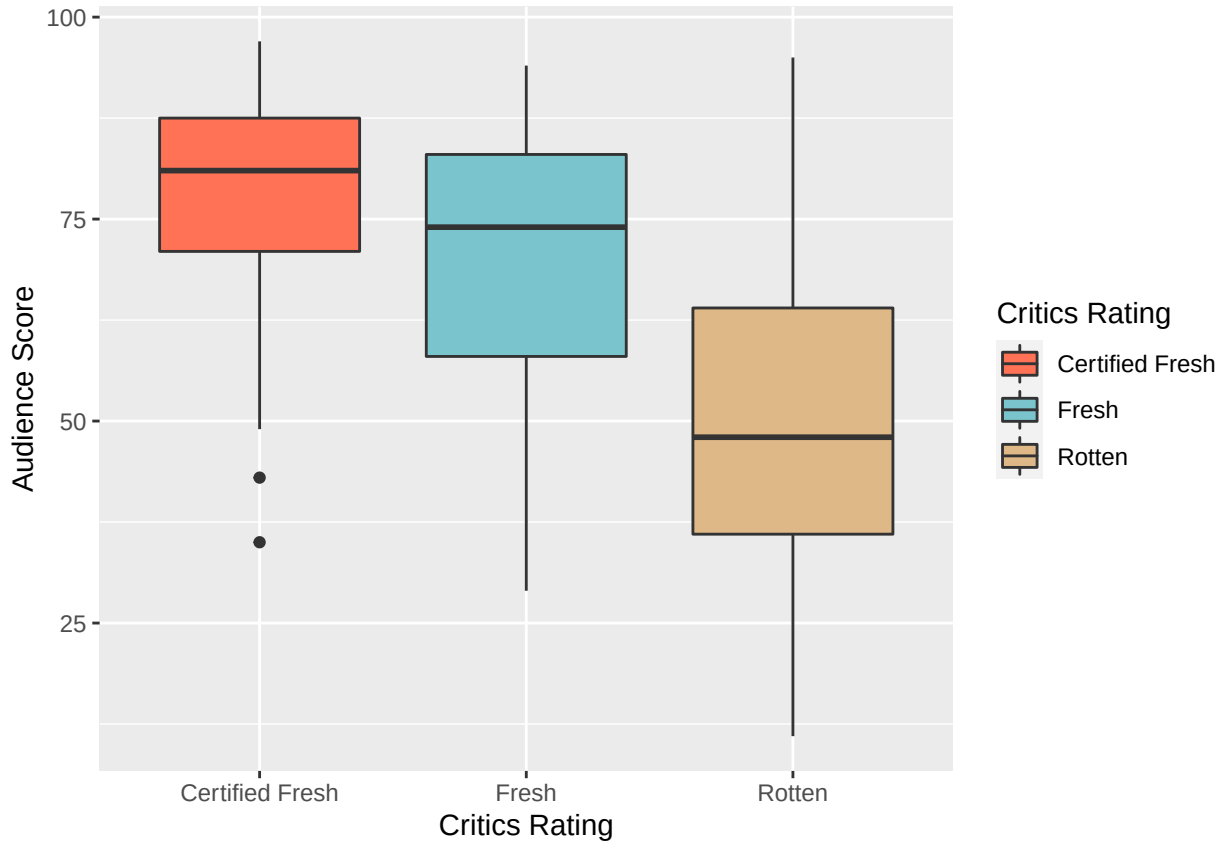

Figure 5: Box plot for Audience Score and Critics Rating

```
#This code shows the boxplot for the predictor audience_rating and the response variable audience_score
ggplot(data=movies_filtered, aes(x = audience_rating, y = audience_score, fill = audience_rating)) +
  geom_boxplot() +
  scale_fill_manual(values=c("coral1","cadetblue3"), guide = guide_legend("Audience Rating")) +
  labs(x = "Audience Rating", y = "Audience Score")
```

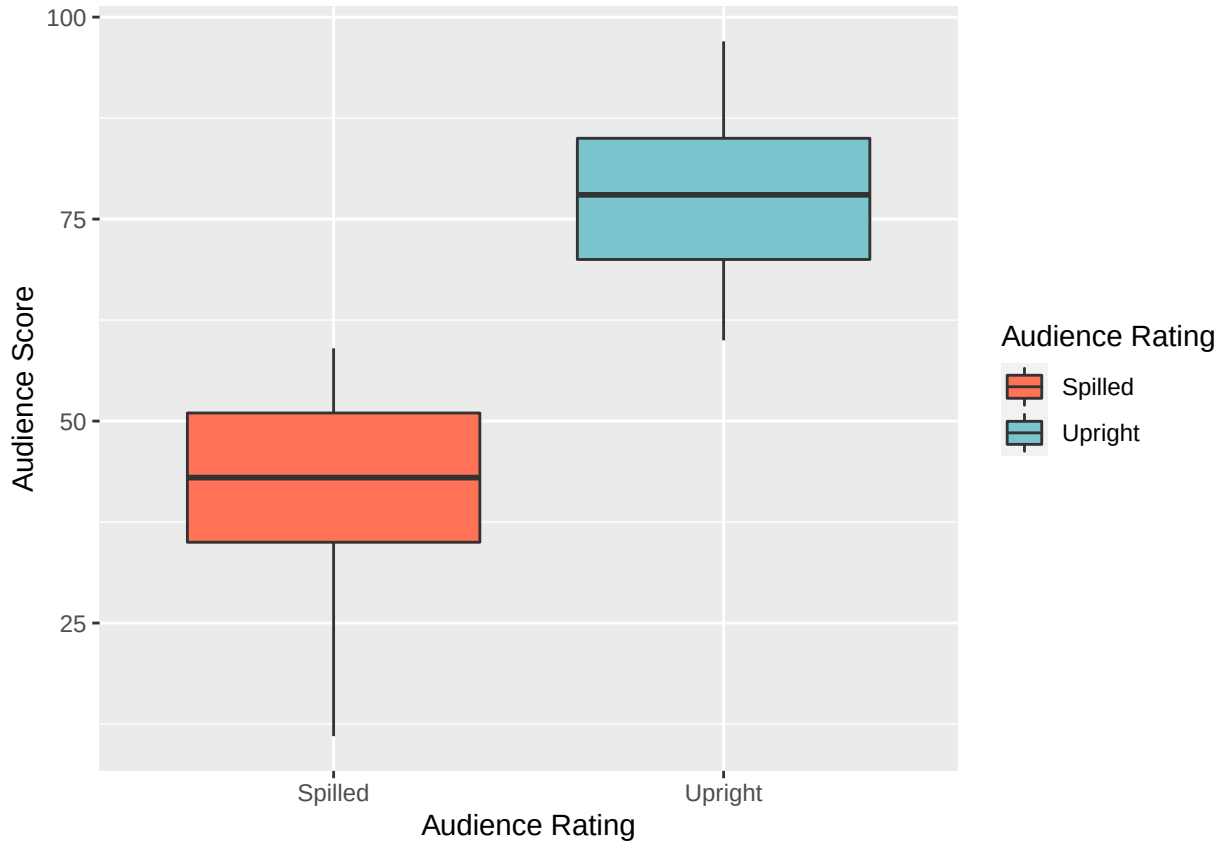

Figure 6: Box plot for Audience Score and Audience Rating

From figure 5, it can be observed that the median of scores of categories "Certified Fresh" and "Fresh" are higher than the one belonging to the category "Rotten" as is the case for the Audience Rating category "Upright" compared to the category "Spilled" shown in figure 6. There is no strong skewness in the data. The results are coherent to what is expected.

Part 4: Modeling

Two common strategies for adding or removing variables in a multiple regression model are called backward elimination and forward selection.

In order to develop a multiple linear regression model to predict a numerical variable in the movies dataset, a forward selection strategy will be performed. The forward selection strategy starts with an empty model followed by adding variables one-at-a-time until there are not any variables that improve the model (as measured by adjusted R-squared). This strategy is suited for this data set because of the large number of explanatory variables and herence is preferred over Backward elimination strategy.

Firstly, the full model is considered to find the first predictor, based on the lowest p-value:

```
#This code gives the summary statistics for the full model considering all variables at once
fullmodel_movies <- lm(audience_score ~ imdb_rating + imdb_num_votes + critics_rating +
                    critics_score + audience_rating, data = movies_filtered)
summary(fullmodel_movies)
```

```
##
## Call:
## lm(formula = audience_score ~ imdb_rating + imdb_num_votes +
##     critics_rating + critics_score + audience_rating, data = movies_filtered)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.8461  -4.6326   0.7755   4.3839  24.2043
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -8.718e+00  2.664e+00  -3.272  0.00112 **
## imdb_rating             9.050e+00  4.662e-01  19.413  < 2e-16 ***
## imdb_num_votes          1.110e-06  2.790e-06   0.398  0.69083
## critics_ratingFresh    -2.745e-01  8.593e-01  -0.319  0.74950
## critics_ratingRotten   -8.603e-01  1.397e+00  -0.616  0.53823
## critics_score           1.435e-02  2.494e-02   0.575  0.56522
## audience_ratingUpright  2.063e+01  7.803e-01  26.443  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.954 on 644 degrees of freedom
## Multiple R-squared:  0.8828, Adjusted R-squared:  0.8817
## F-statistic: 808.8 on 6 and 644 DF,  p-value: < 2.2e-16
```

According to the results shown above, the predictor imdb_rating is chosen as the first variable to be added to the empty model due to its very low p-value close to 0 (2e-16).

```
#This code find the summary statistics including p-values
#and Multiple R-squared, Adjusted R-squared for the empty model + imdb_rating
movies1 <- lm(audience_score ~ imdb_rating, data = movies_filtered)
summary(movies1)
```

```
##
## Call:
## lm(formula = audience_score ~ imdb_rating, data = movies_filtered)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.800  -6.567   0.649   5.689  52.896
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -42.3284     2.4183  -17.50   <2e-16 ***
## imdb_rating  16.1234     0.3674   43.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.16 on 649 degrees of freedom
## Multiple R-squared:  0.748,  Adjusted R-squared:  0.7476
## F-statistic:  1926 on 1 and 649 DF,  p-value: < 2.2e-16
```

The results show a value of multiple R-squared and Adjusted R-squared of 0.75 and p-value 2.2e-16 (almost 0), herence, imdb_rating is a significant predictor of the response variable audience_score.

Followed by imdb_rating, a second predictor to be chosen according to its low p-value would be audience_rating. The results of adding this second variable to the empty model are shown below:

```
#This code find the summary statistics including p-values
#and Multiple R-squared, Adjusted R-squared for the empty model
#+ imdb_rating + audience_rating
movies2 <- lm(audience_score ~ imdb_rating + audience_rating, data = movies_filtered)
summary(movies2)
```

```
##
## Call:
## lm(formula = audience_score ~ imdb_rating + audience_rating,
##     data = movies_filtered)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.1570  -4.7630   0.6209   4.3572  24.3224
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -11.4864     2.0068  -5.724 1.59e-08 ***
## imdb_rating                9.5191     0.3497  27.220  < 2e-16 ***
## audience_ratingUpright    20.8473     0.7674  27.166  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.952 on 648 degrees of freedom
## Multiple R-squared:  0.8822, Adjusted R-squared:  0.8818
## F-statistic:  2426 on 2 and 648 DF,  p-value: < 2.2e-16
```

After adding the second predictor audience_rating, it has been found that the model Multiple R-Squared and Adjusted R-squared have improved notably.

The third variable to be added to the existing model will be critics_rating, according to the next lowest p-value obtained as criteria that is being followed in this process:

```
#This code find the summary statistics including p-values
#and Multiple R-squared, Adjusted R-squared for the empty model
#+ imdb_rating + audience_rating + critics_rating
movies3 <- lm(audience_score ~ imdb_rating + audience_rating +
                critics_rating, data = movies_filtered)
summary(movies3)
```

```
##
## Call:
## lm(formula = audience_score ~ imdb_rating + audience_rating +
##     critics_rating, data = movies_filtered)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.117  -4.709   0.700   4.422  24.165
##
## Coefficients:
```

```
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -8.5173     2.6160  -3.256  0.00119 **
## imdb_rating               9.2160     0.3888  23.704  < 2e-16 ***
## audience_ratingUpright   20.6195     0.7785  26.486  < 2e-16 ***
## critics_ratingFresh      -0.4708     0.7854  -0.599  0.54905
## critics_ratingRotten     -1.5235     0.9030  -1.687  0.09207 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.946 on 646 degrees of freedom
## Multiple R-squared:  0.8828, Adjusted R-squared:  0.882
## F-statistic:  1216 on 4 and 646 DF,  p-value: < 2.2e-16
```

After adding the third predictor critics_rating, it has been found that the model Multiple R-Squared value barely improved from 0.8822 to 0.8828 and Adjusted R-squared has almost the same value of 0.882.

The fourth variable to be added to the existing model will be critics_score, according to the next lowest p-value obtained as criteria that is being followed in this process:

```
#This code find the summary statistics including p-values
#and Multiple R-squared, Adjusted R-squared for the empty model
#+ imdb_rating + audience_rating + critics_rating + critics_score
movies4 <- lm(audience_score ~ imdb_rating + audience_rating +
                critics_rating + critics_score, data = movies_filtered)
summary(movies4)
```

```
##
## Call:
## lm(formula = audience_score ~ imdb_rating + audience_rating +
##     critics_rating + critics_score, data = movies_filtered)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.8831  -4.6944   0.7152   4.4042  24.2502
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -8.76323    2.66024  -3.294  0.00104 **
## imdb_rating              9.09982    0.44910  20.262  < 2e-16 ***
## audience_ratingUpright  20.61913    0.77895  26.471  < 2e-16 ***
## critics_ratingFresh     -0.40176    0.79707  -0.504  0.61440
## critics_ratingRotten    -1.00641    1.34695  -0.747  0.45523
## critics_score            0.01273    0.02459   0.518  0.60490
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.95 on 645 degrees of freedom
## Multiple R-squared:  0.8828, Adjusted R-squared:  0.8819
## F-statistic: 971.8 on 5 and 645 DF,  p-value: < 2.2e-16
```

The results shown above that both Multiple R-squared and Adjusted R-squared did not improve the existing model as the values almost remain the same and stuck at 0.8828 for Multiple R-squared and 0.882 for Adjusted R-squared.

The forward selection strategy has been performed until a Parsimonious model has been found. A Parsimonious model is a simple model with great explanatory predictive power. It explains data with a minimum number of predictor variables, so it uses the right amount of predictors needed to explain the model well. In this case, the Parsimonious model has been reached after adding the third predictor critics_rating. The final model is therefore as follows:

```
#This code find the summary statistics including p-values
#and Multiple R-squared, Adjusted R-squared for the empty model
#+ imdb_rating + audience_rating + critics_rating
movies3 <- lm(audience_score ~ imdb_rating + audience_rating +
                critics_rating, data = movies_filtered)
summary(movies3)
```

```
##
## Call:
## lm(formula = audience_score ~ imdb_rating + audience_rating +
##     critics_rating, data = movies_filtered)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.117  -4.709   0.700   4.422  24.165
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              -8.5173     2.6160  -3.256  0.00119 **
## imdb_rating               9.2160     0.3888  23.704  < 2e-16 ***
## audience_ratingUpright   20.6195     0.7785  26.486  < 2e-16 ***
## critics_ratingFresh      -0.4708     0.7854  -0.599  0.54905
## critics_ratingRotten     -1.5235     0.9030  -1.687  0.09207 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.946 on 646 degrees of freedom
## Multiple R-squared:  0.8828, Adjusted R-squared:  0.882
## F-statistic:  1216 on 4 and 646 DF,  p-value: < 2.2e-16
```

Interpretation of model coefficients

The model coefficients interpretation is shown as follows:

1. Intercept: -8.5173. This is the estimated audience_score value for a movie considering the predictors imdb_rating, audience_rating and critics_rating having a value of 0.
2. imdb_rating coefficient: 9.2160. For every one unit increase in imdb_rating (predictor), the audience_score (response variable) increase on average 9.2160 times, all else hold constant.
3. audience_rating coefficient: 20.6195. For every one unit increase in audience_rating (predictor), the audience_score (response variable) increase on average 20.6195 times, all else hold constant.
4. critics_ratingFresh coefficient: -0.4708. For every one unit increase in critics_ratingFresh (predictor), the audience_score (response variable) decrease on average 0.4708 times, all else hold constant.
5. critics_ratingRotten coefficient: -1.5235. For every one unit increase in critics_ratingRotten (predictor), the audience_score (response variable) decrease on average 1.5235 times, all else hold constant.
6. Model Multiple R-squared: 0.8828. The model explains 88.28% of all the variability of audience_score (response variable).

7. Model Adjusted R-squared: 0.882. The adjusted R-squared is a modified version of R-squared that has been adjusted for the number of predictors in the model. The adjusted R-squared increases only if the new term improves the model more than would be expected by chance. It decreases when a predictor improves the model less than expected by chance. In this case, Adjusted R-squared remains almost constant because adjusted R-squared applies a penalty for including an extra predictor in the model.
8. Degrees of Freedom (DF): 646. In multiple regression the degrees of freedom is calculated as follow: df = n - k - 1 where n is the number of observations, in this movies data set n = 651, k is the number of predictors used, in this case k = 4. The result then is: df = 651 - 4 - 1 = 646.
9. Overall F-test of significance: p-value < 2.2e-16. The value obtained for the p-value for the overall F-test of significance < 2.2e-16 (almost 0) indicates that the sample data provide sufficient evidence to conclude that the regression model fits the data better than the model with no independent variables whether the p-value is less than the significance level, which is the case.

Model diagnostics

Linear condition: Figure 7 shows the residual plot in order to check the linearity condition:

```
#This code plots the residual plot of the model selected
ggplot(data = movies3, aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Fitted values") +
  ylab("Residuals") + ggtitle("Residual Plot")
```

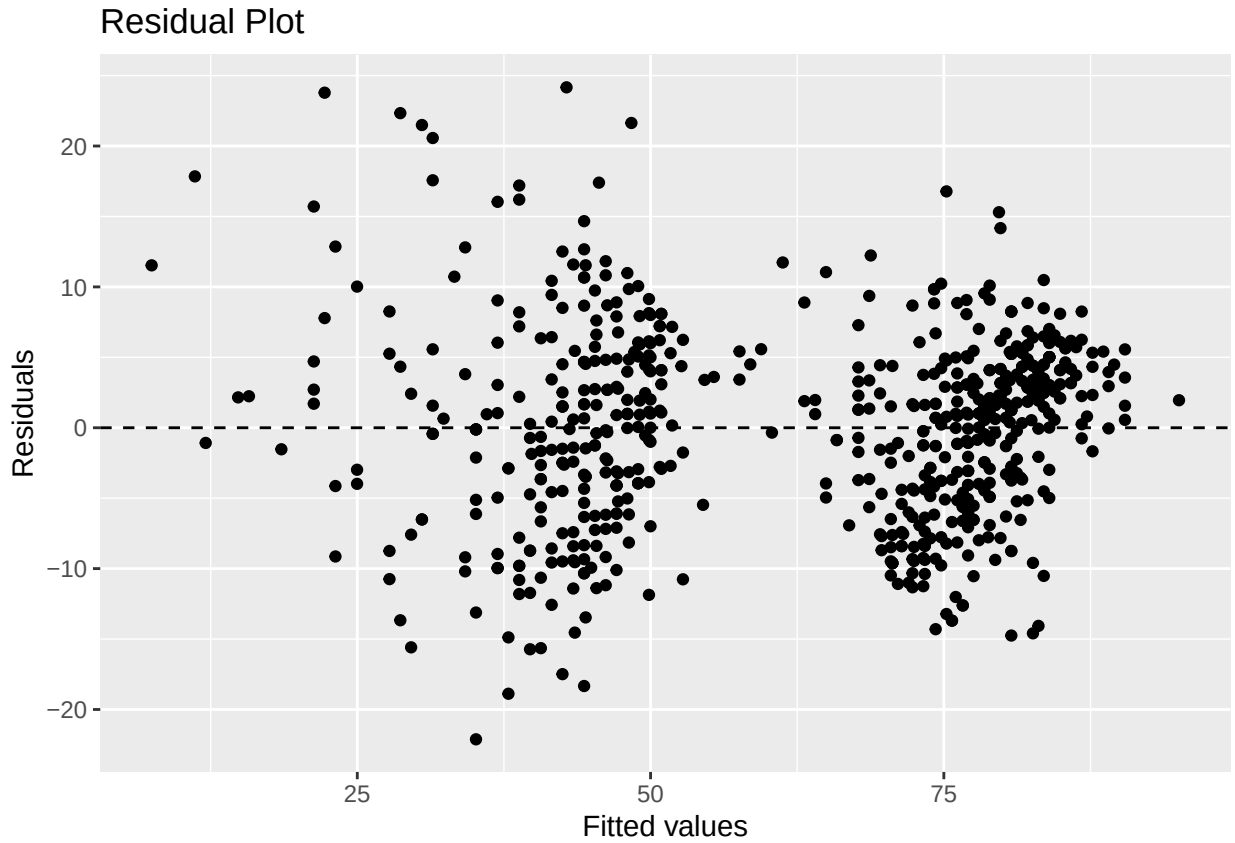

Figure 7: Residual plot of the final model

As can be observed in Figure 7, the linear condition is met as there is a complete random scatter around zero and no fan shape has been found.

Normalily condition: Figure 8 shows the normal probability plot to check the normality condition of the model.

```
#This code plots the normal probability plot of the model selected
qqnorm(movies3$residuals, main="Normality Condition")
qqline(movies3$residuals, main="Normality Condition")
```

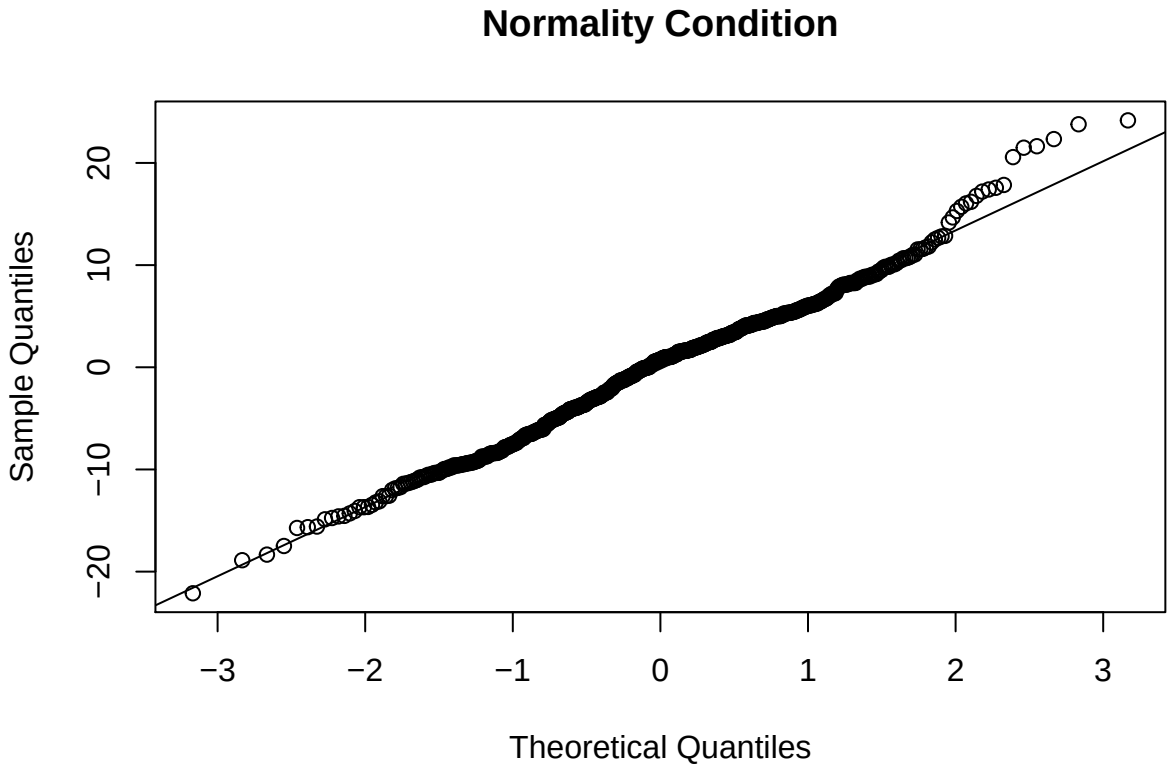

Figure 8: Normal probability plot of the final model

Normality condition is met as the majority of the points lie on the line with the exception of a few points causing some skweness.

Figure 9 shows an histogram to check the nearly normal condition:

```
#This code plots the histogram of the final model
hist(movies3$residuals, prob=TRUE, main="Normality Condition", xlab = "Residuals",
     border = "black", col = "darkseagreen4")
lines(density(movies3$residuals), col="darkblue", lwd=3)
```

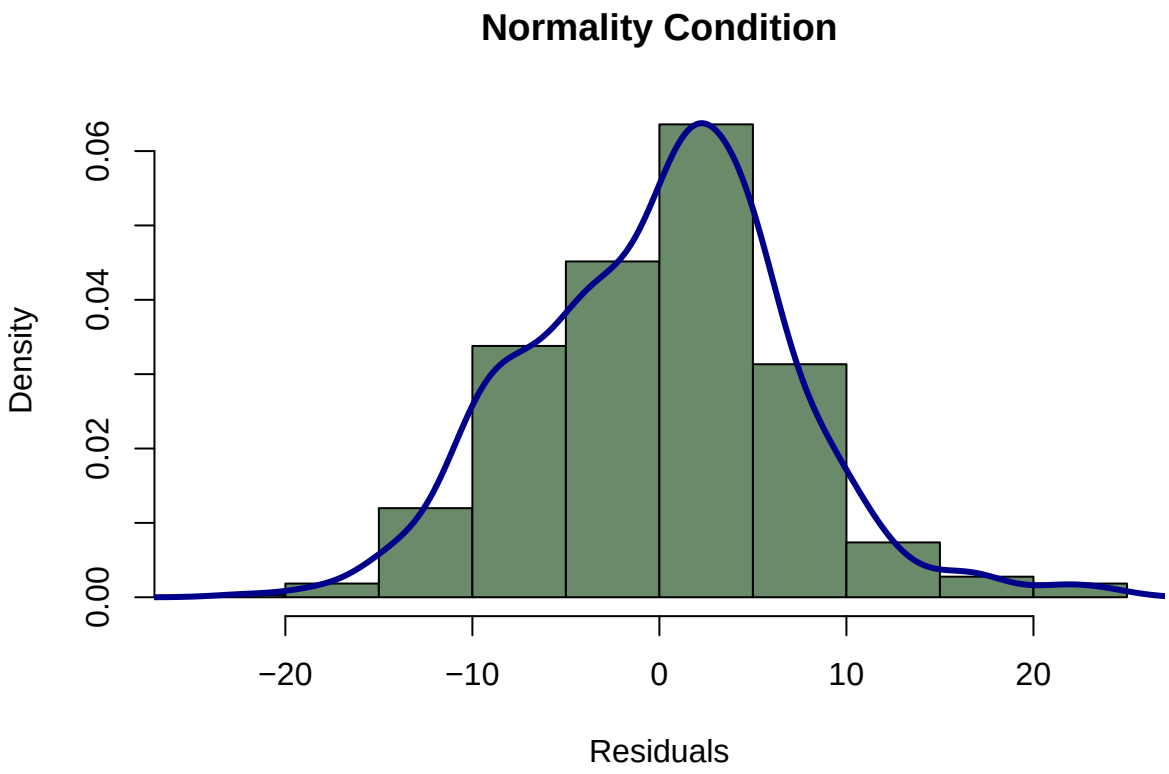

Figure 9: Histogram plot of the final model

Nearly normal condition is met as the distribution is fairly symmetric with some right skweness.

Independence condition: Figure 10 shows a scatter plot to find the independence of the model.

```
#This code plots a scatter plot to seek for independence
plot(movies3$residuals, main="Independence Conditions", ylab = "Residuals")
```

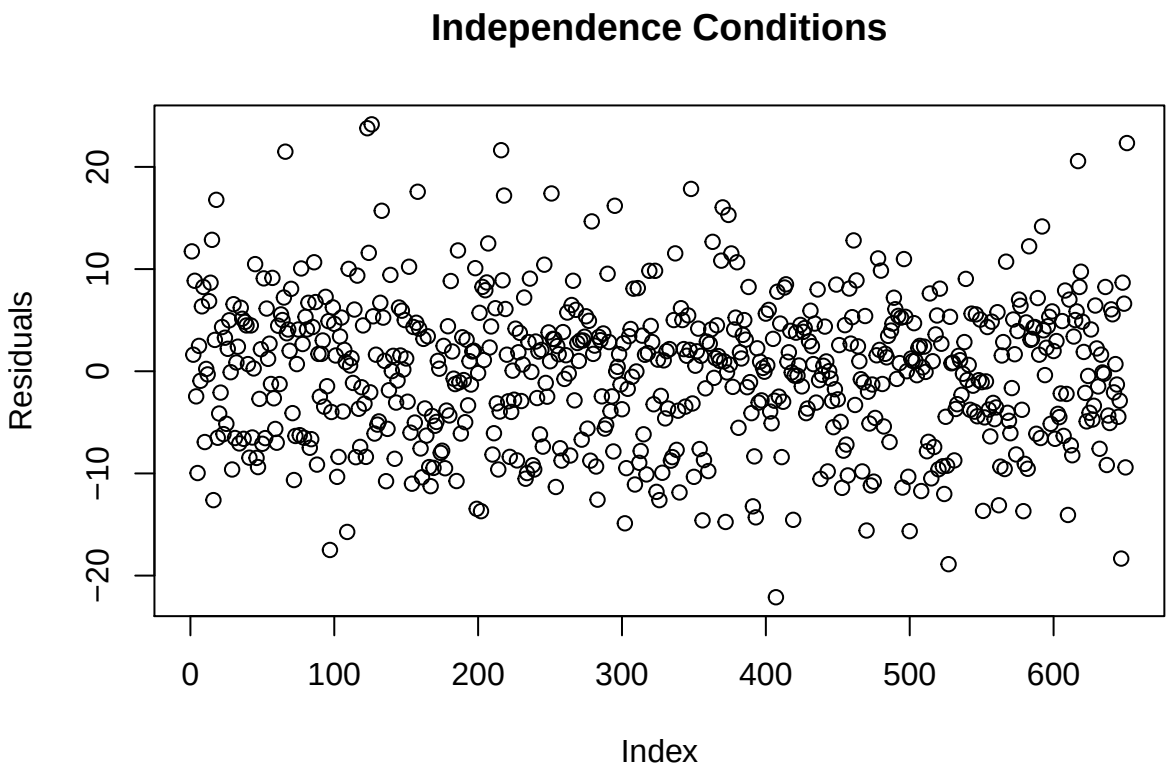

Figure 10: Scatter plot of the final model

As observed in Figure 10, independence condition is met as there is a constant variability and residuals are randomly scattered around 0.

Part 5: Prediction

The final model movies3 is used to predict the audience score for the movie "In the Name of the Father". Firstly, the data is stored in a new variable Test as follows:

```
#This code stores the data in a new variable "Test"
genre <- "Drama"
imdb_rating <- 8.1
audience_rating <- "Upright"
critics_rating <- "Certified Fresh"

Test <- data.frame (genre, imdb_rating, audience_rating, critics_rating)
```

Following the audience score is predicted in the final model using the new variable "Test":

```
#This code predicts the movie selected following the criteria above.
Prediction <- Test %>%
  select(genre, imdb_rating, audience_rating, critics_rating)
predict(movies3, Prediction)
```

```
##        1
## 86.75163
```

The model predicts the movie "In the Name of the Father" in the test set will have an audience score at approximate 87. Following, a confidence interval for a Confidence level of 95% is calculated:

```
#This code makes an estimation of the confidence interval
#under a confidence level of 95%
predict(movies3, Prediction, interval = "prediction", level = 0.95)
```

```
##        fit     lwr      upr
## 1 86.75163 73.0532 100.4501
```

The value predicted is 86.75 with a confidence interval: (73.05, 100.45). We are 95% confident that, all else being equal, the predicted audience score for the movie "In the Name of the Father" will be between 73.05 and 100.45 on average.

The actual audience score for the movie "In the Name of the Father" is the following:

```
#This code gives the audience score for the movie "In the Name of the Father"
In_the_Name_of_the_Father <- movies %>%
  filter(title == "In the Name of the Father") %>%
  select(audience_score)
In_the_Name_of_the_Father
```

```
## # A tibble: 1 x 1
##   audience_score
##            <dbl>
## 1             95
```

Since the actual audience score for the movie "In the Name of the Father" is 95, the confidence interval contains this value.

Part 6: Conclusion

The exploratory data analysis performed allowed finding the appropriate predictors to include and build the model. In addition, as Moghaddam, F. et al. 2019 found in their study "predicting Movie Popularity and Ratings with Visual Features", an hybrid approach selecting categorical and numerical variables achieves good results by predicting the popularity of a movie.

In the present study, the model obtained was able to roughly predict the audience score of the movie "In the Name of the Father" using a confidence level of 95%, getting a value of 87 compared to the actual 95 with a confidence interval (73, 100) which contains the actual value of the audience score for this movie. Nevertheless, the predictive power of this model is limited because the sampling is not large enough to allow the model to achieve a higher accuracy. Furthermore, the data is biased towards drama movies so that the model is rather more accurate predicting the popularity of drama movies and not so robust to predict movies belonging to other genres.

For a further research, there are some recommendations to be taken into account to achieve better results looking for improving Multiple R-square and Adjusted R-square: (1) Perform a larger sampling to collect data in order to get more variability in the population, (2) Build different models for each genre, (3) use a stratified sampling technique to reflect the true proportion of movie genres in the population.