Modeling and prediction for movies

Michelle Lin

July 15, 2022

Setup

Load packages

```
library(ggplot2)
library(dplyr)
library(statsr)
library(knitr)
library(formatR)
opts_chunk$set(tidy.opts=list(width.cutoff=80),tidy=TRUE)
```

Load data

```
load("/cloud/project/movies.rdata")
```

Data Introduction

This study is an experimental study given that there was random sample selection. The data consists of 651 randomly sampled movies that were produced and released prior to 2016. Although randomly sampled, I would hesitate in generalizing the trends found in this analysis to movies past 2016, since movie trends and production methods can vary year to year. Moreover, a potential source of bias could arise given that movie information was sampled from Rotten Tomatoes and IMDB, which excludes foreign films or smaller indie films that do not publish information on those sites. However, overall, these biases will have negligible impact since the data and the findings of the study will be published in the context of Western films in the United States.

Research Questions Posed

What characteristics of a movie correlate with higher audience scores for a movie on Rotten Tomatoes? Specifically, which of the following variables are correlated with higher audience scores?

- Genre
- Critics Score on Rotten Tomatoes
- IMDB rating
- Whether or not a movie is in the Top 200 Box Office List
- MPAA Rating

##Exploratory data analysis

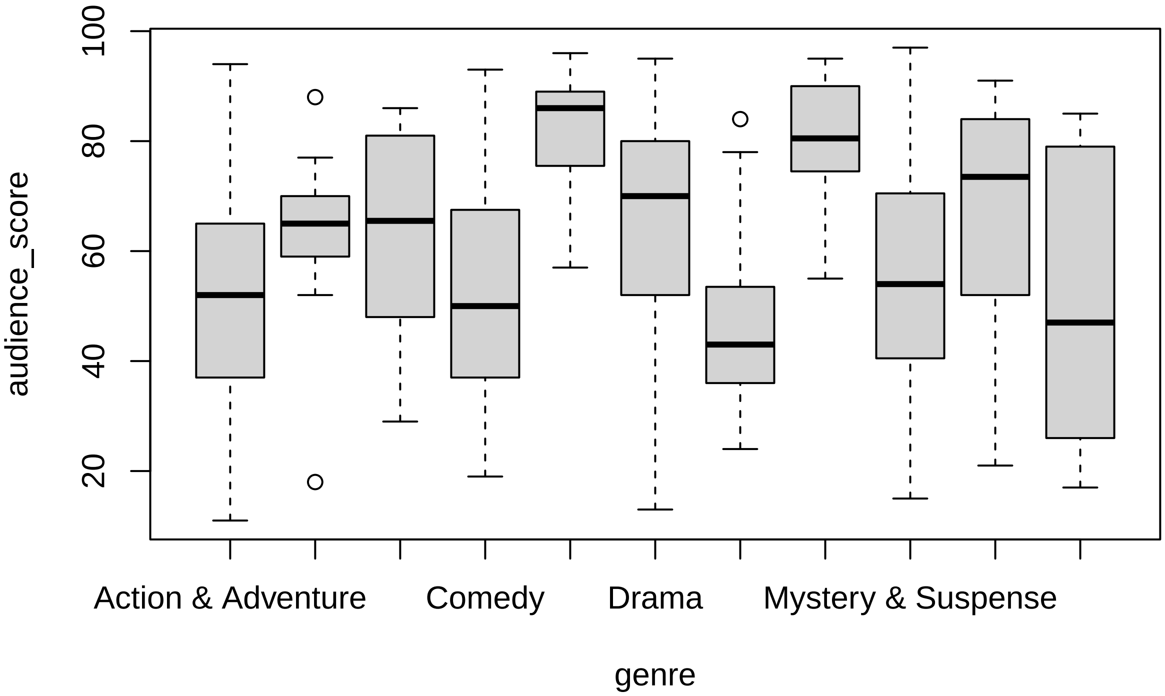
I'd like to look at the variables of interest to see whether or not they are feasible for statistical analysis. Statistical analysis was conducted to ensure that the variables satisfy the conditions for Multiple Linear Regression modeling, which are: - Linear relationships between the explanatory and response variable (x and y) - Nearly normal residuals - Constant variability of residuals

```
# First 3 variables are categorical, will wait until multiple linear regression
# fit to conduct statistical analysis. For now, just focused on visualizing
# distribution of categorical data.

# distribution of movie genre in data set
summary(movies$genre)
```

```
##         Action & Adventure                   Animation Art House & International
##                         65                           9                        14
##                     Comedy                 Documentary                     Drama
##                         87                          52                       305
##                     Horror Musical & Performing Arts         Mystery & Suspense
##                         23                          12                        59
##                      Other Science Fiction & Fantasy
##                         16                           9
```

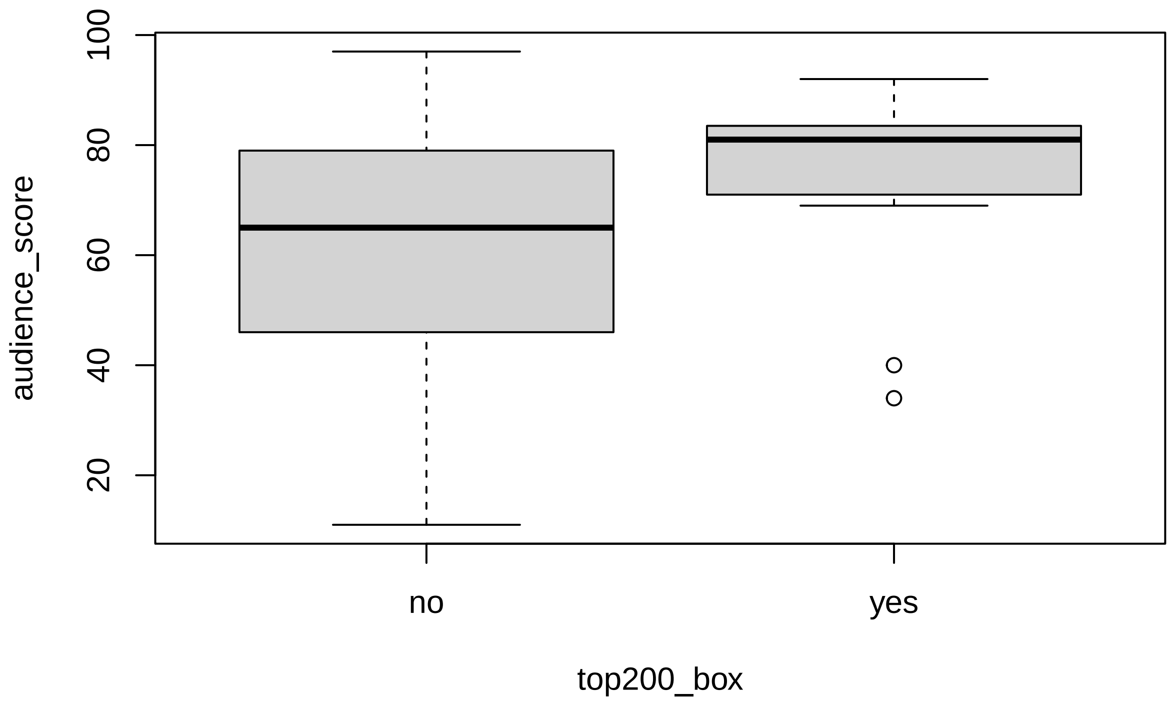
```
boxplot(audience_score ~ genre, data = movies)
```



```
# distribution of whether or not a movie is in the Top 200 Box
summary(movies$top200_box)
```

```
##  no yes
## 636  15
```

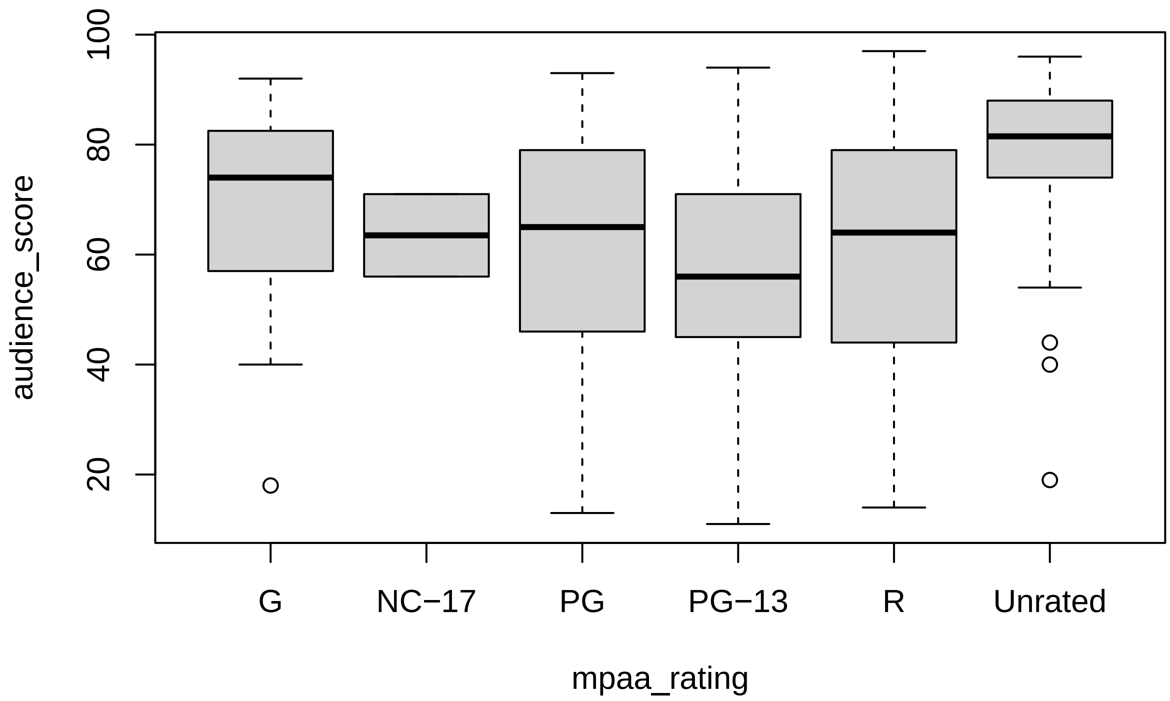
```
boxplot(audience_score ~ top200_box, data = movies)
```



```
# distribution of movie MPAA rating
summary(movies$mpaa_rating)
```

```
##       G   NC-17      PG   PG-13       R Unrated 
##      19       2     118     133     329      50
```

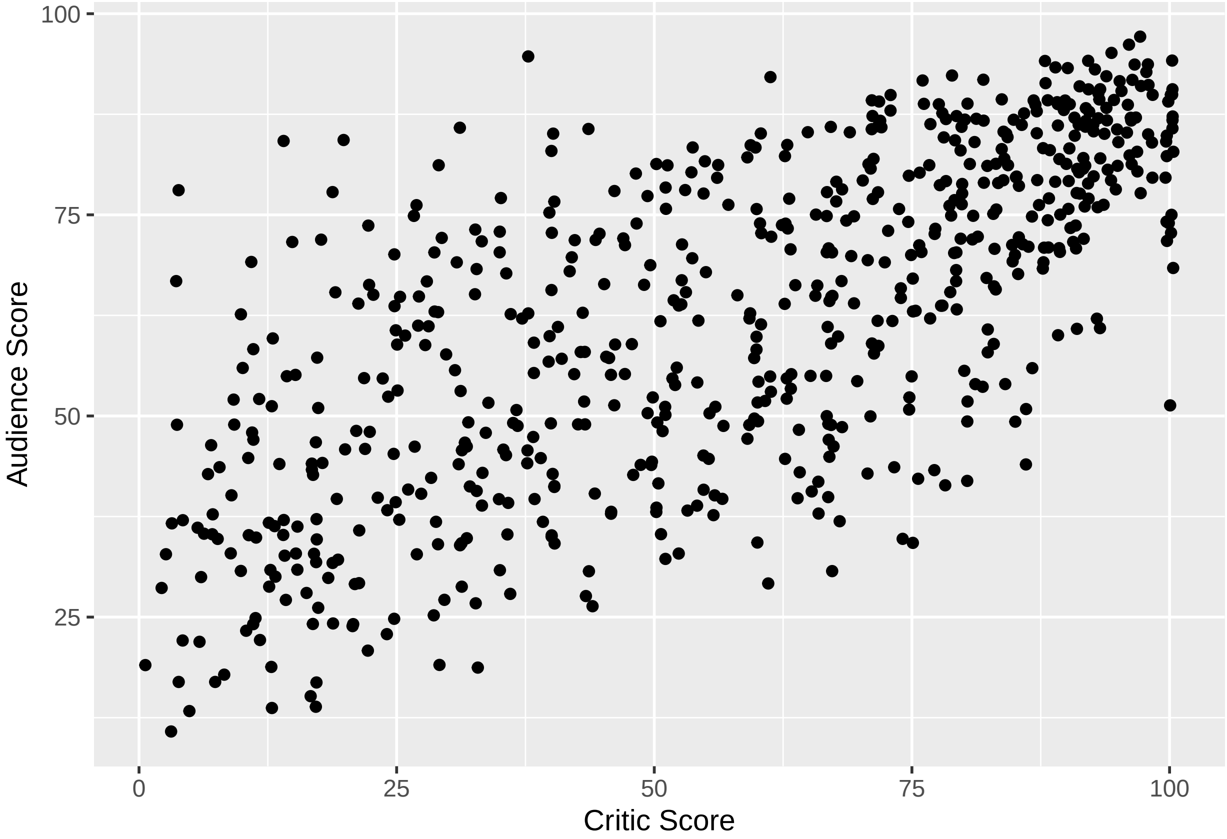
```
boxplot(audience_score ~ mpaa_rating, data = movies)
```



```
# Last 2 variables are numerical, will conduct preliminary linear modeling to
# determine whether multiple linear regression conditions are satisfied before
# combining into multiple linear regression.
# distribution of critics score on Rotten Tomatoes
summary(movies$critics_score)
```

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

```
ggplot(data = movies, aes(x = critics_score, y = audience_score)) + geom_jitter() +
    xlab("Critic Score") + ylab("Audience Score")
```

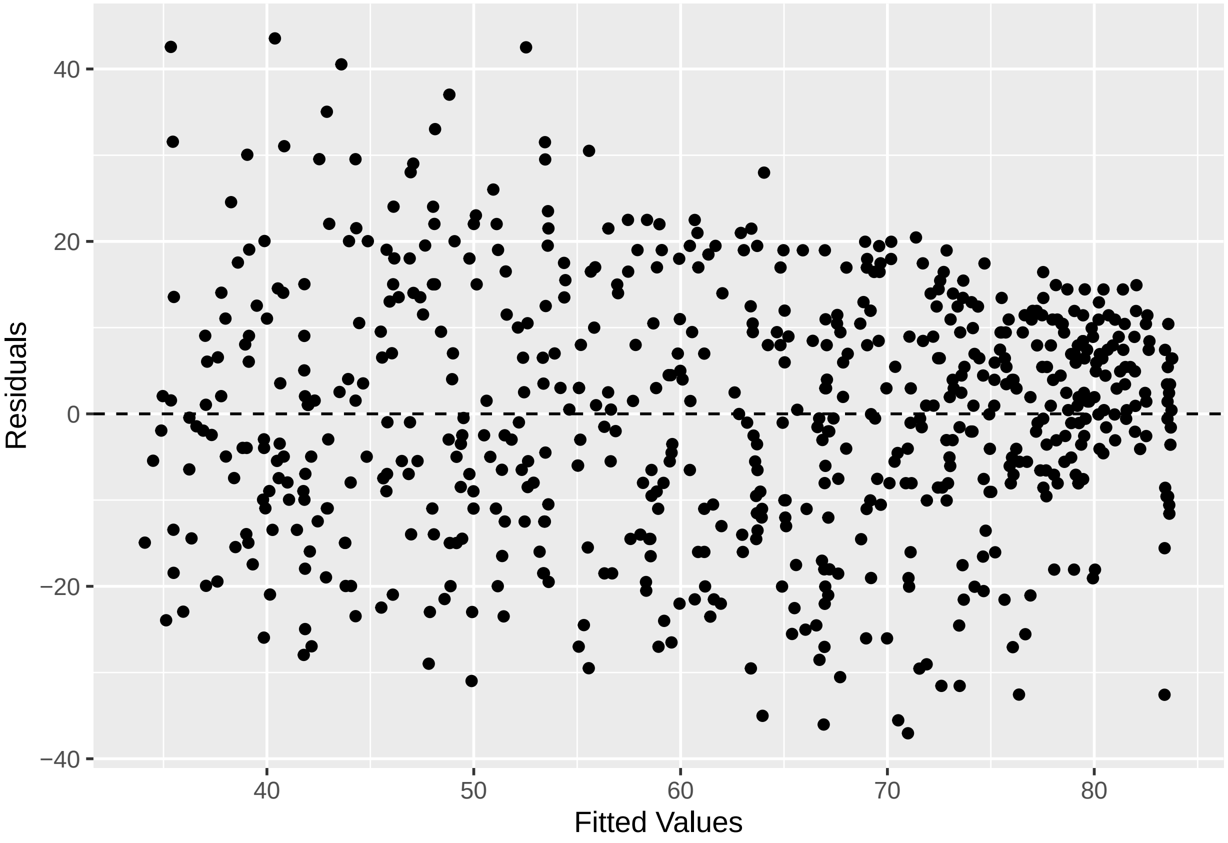


```
cs1 <- lm(audience_score ~ critics_score, data = movies)
summary(cs1)
```

```
##
## Call:
## lm(formula = audience_score ~ critics_score, data = movies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.043  -9.571   0.504  10.422  43.544
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   33.43551    1.27561   26.21   <2e-16 ***
## critics_score  0.50144    0.01984   25.27   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

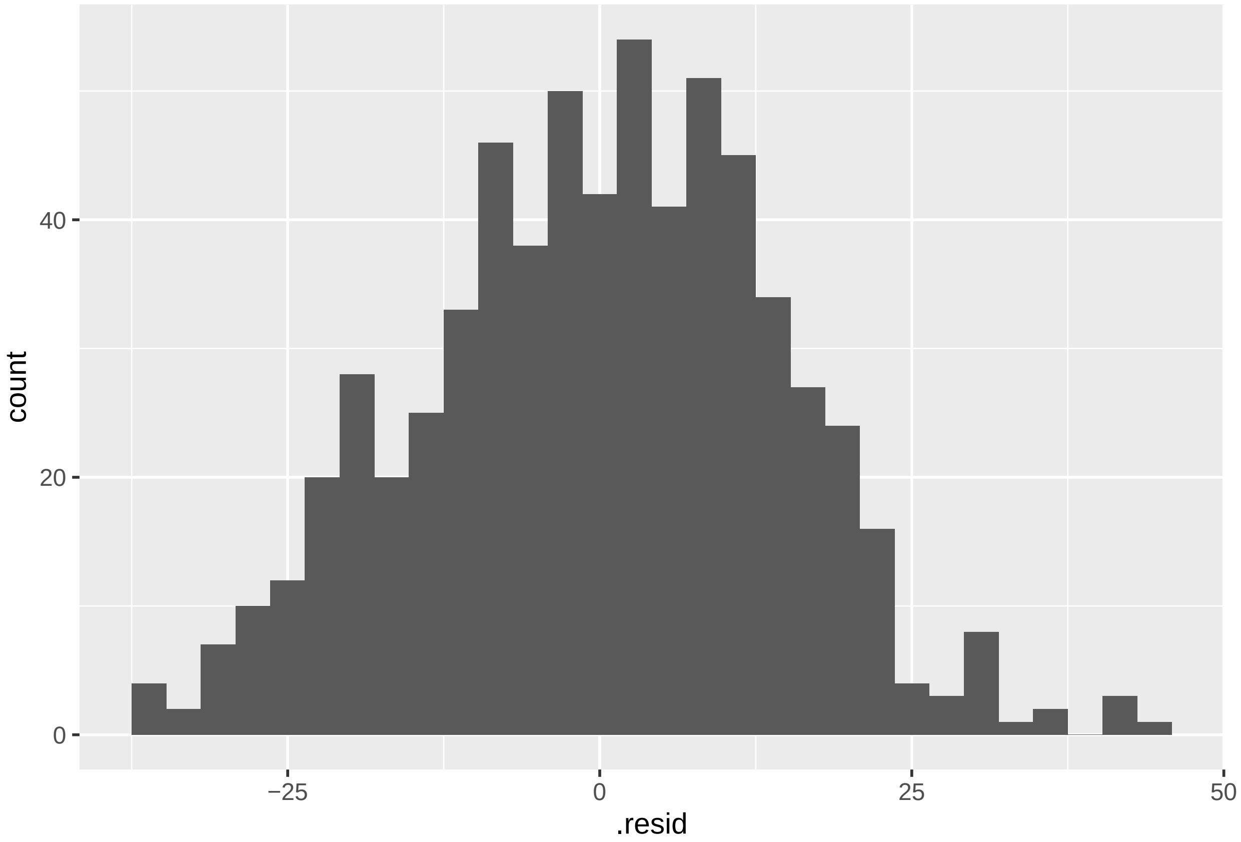
```
## Residual standard error: 14.37 on 649 degrees of freedom
## Multiple R-squared:  0.496,  Adjusted R-squared:  0.4952
## F-statistic: 638.7 on 1 and 649 DF,  p-value: < 2.2e-16
```

```
# check for linearity and constant variability
ggplot(data = cs1, aes(x = .fitted, y = .resid)) + geom_jitter() + geom_hline(yintercept = 0,
    linetype = "dashed") + xlab("Fitted Values") + ylab("Residuals")
```

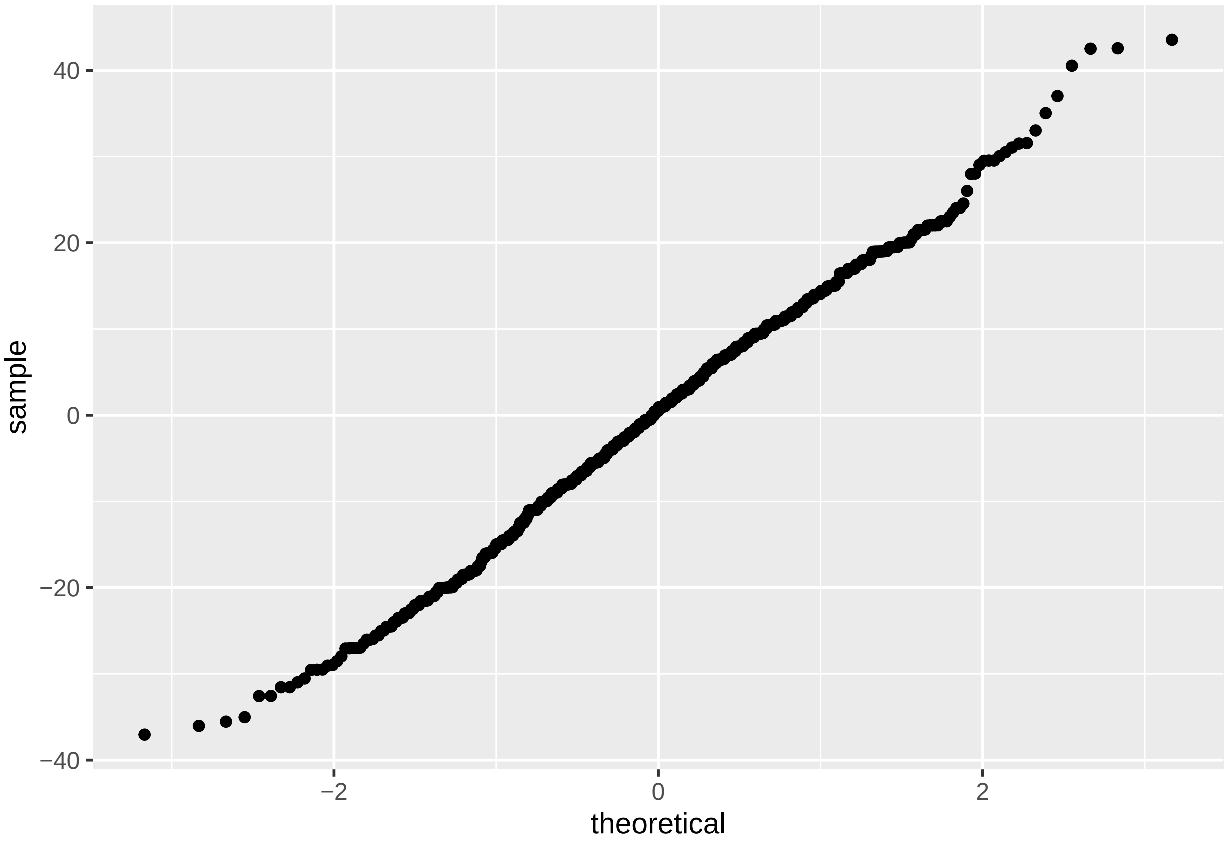


```
# check for nearly normal residuals
ggplot(data = cs1, aes(x = .resid)) + geom_histogram()
```

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



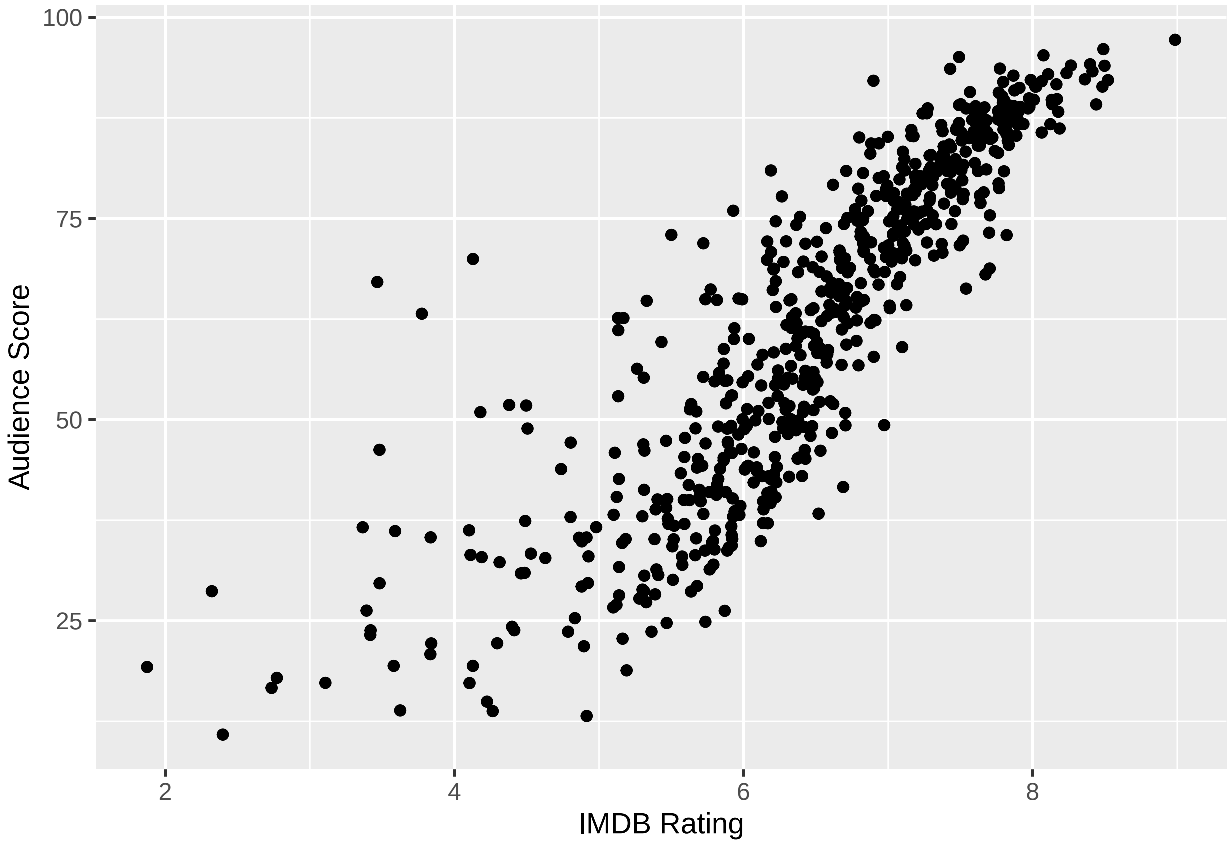
```
ggplot(data = cs1, aes(sample = .resid)) + stat_qq()
```



```
# distribution of IMDB rating
summary(movies$imdb_rating)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   1.900   5.900   6.600   6.493   7.300   9.000
```

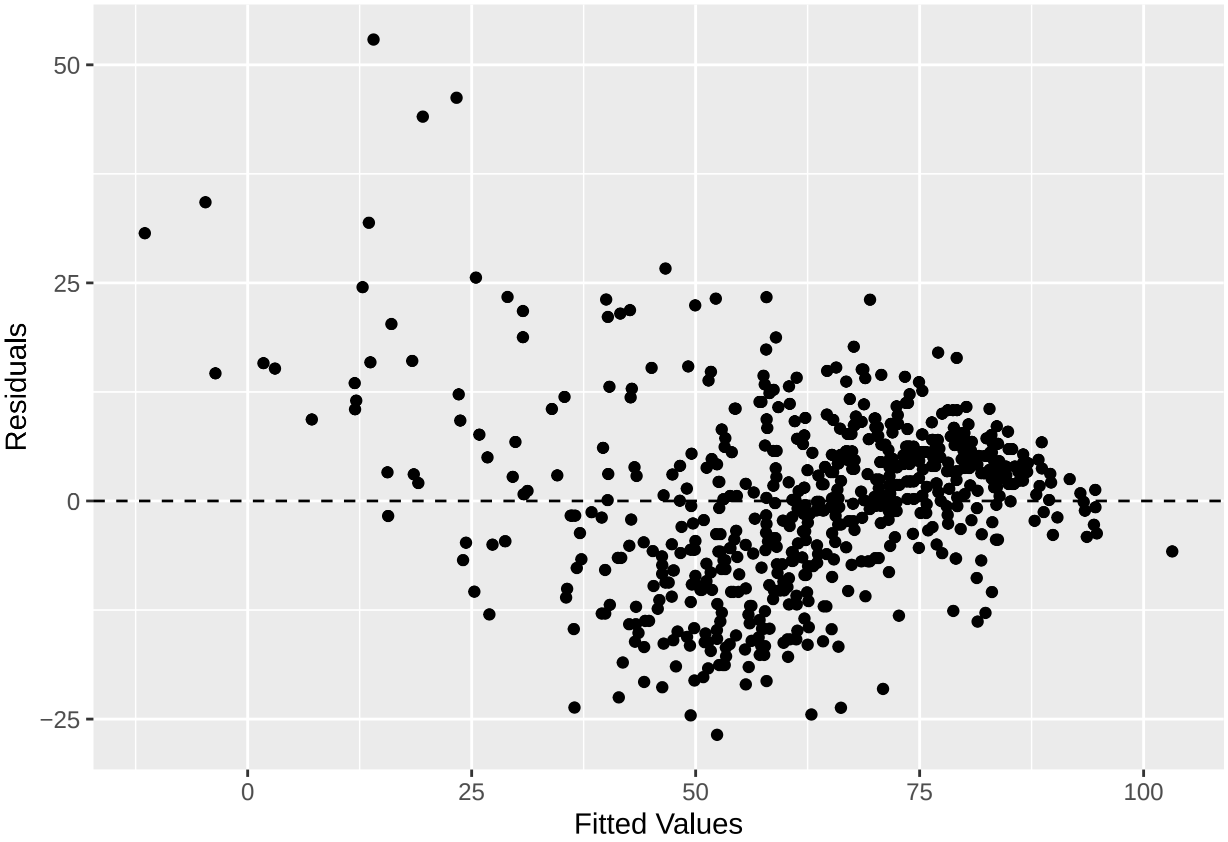
```
ggplot(data = movies, aes(x = imdb_rating, y = audience_score)) + geom_jitter() +
    xlab("IMDB Rating") + ylab("Audience Score")
```



```r
imdb1 <- lm(audience_score ~ imdb_rating, data = movies)
summary(imdb1)
```

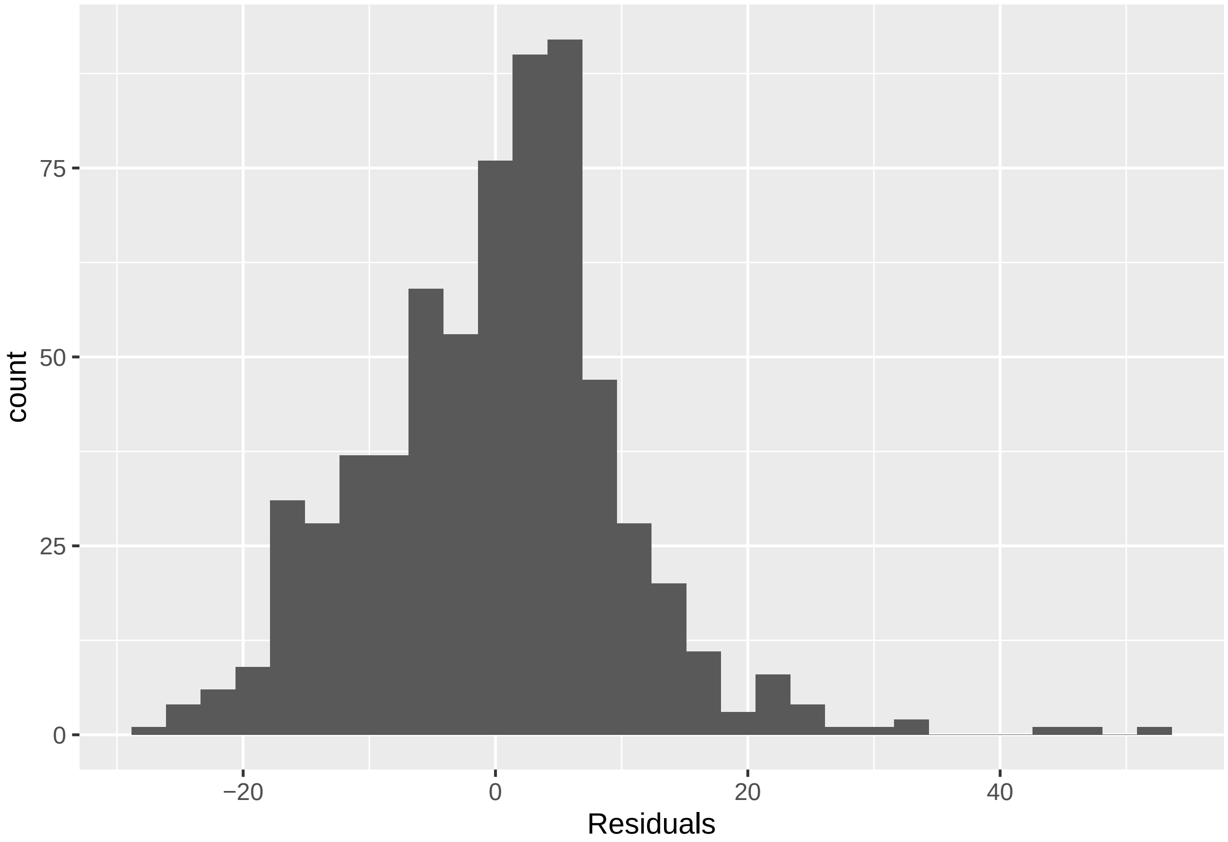
```
##
## Call:
## lm(formula = audience_score ~ imdb_rating, data = movies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.800  -6.567   0.649   5.689  52.896
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -42.3284     2.4183  -17.50   <2e-16 ***
## 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
```

```r
# check for linearity and constant variability
ggplot(data = imdb1, aes(x = .fitted, y = .resid)) + geom_jitter() + geom_hline(yintercept = 0,
    linetype = "dashed") + xlab("Fitted Values") + ylab("Residuals")
```

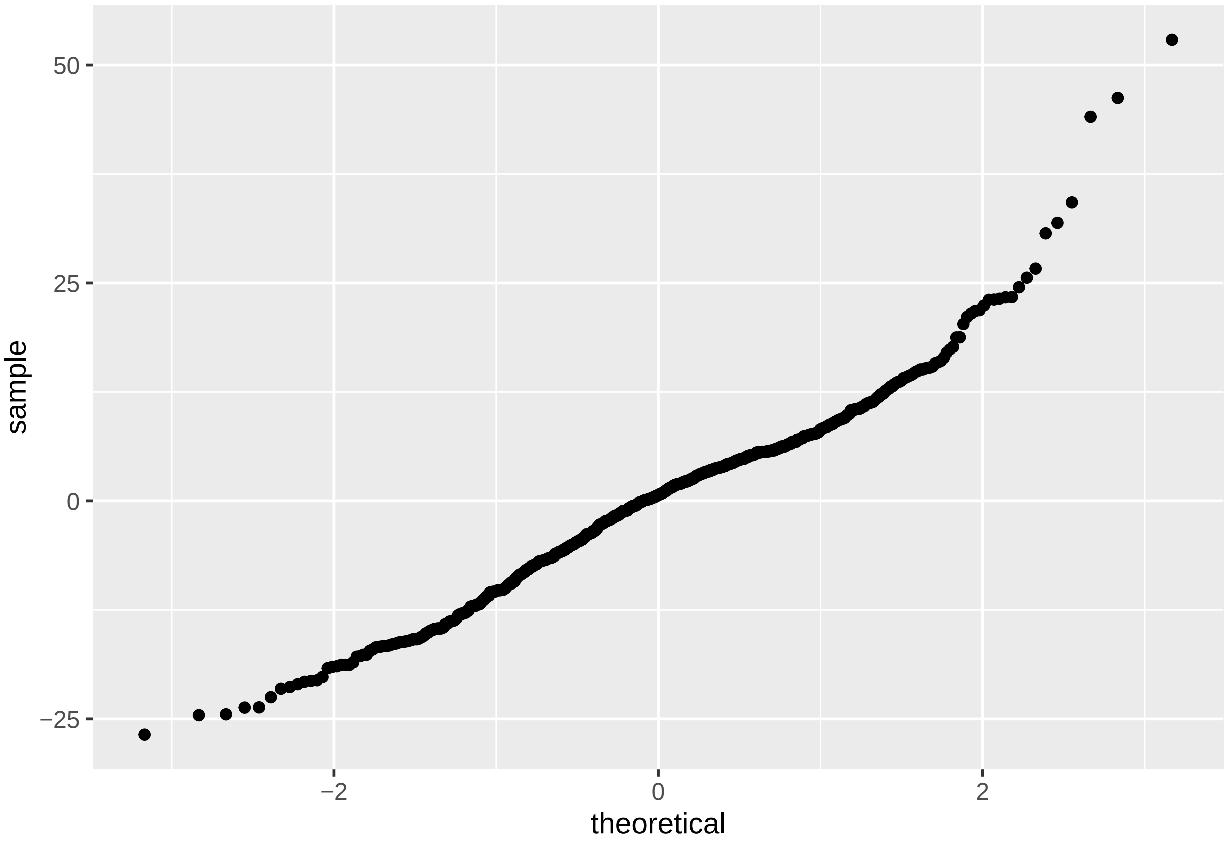


```
# check for nearly normal residuals
ggplot(data = imdb1, aes(x = .resid)) + geom_histogram() + xlab("Residuals")
```

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



```
ggplot(data = imdb1, aes(sample = .resid)) + stat_qq()
```



Can conclude that there is a linear relationship between audience score and critics score from Rotten Tomatoes. However, the same can not be said for the relationship between audience score on Rotten Tomatoes and IMDB ratings. There is a slight right skew in the data, and the residuals do not seem to have constant variance around mean of 0. However, it will be considered in the full model analysis for educational purposes and may be removed when selecting the parsimonious model.

Full Model and Parsimonious Model Selection

The full model will consider the following variables in relation to audience score on Rotten Tomatoes: - Critic score on Rotten Tomatoes - IMDB rating - Genre - Whether or not a movie is in the Top 200 Box Office List - MPAA Rating

```
fullmodel <- lm(audience_score ~ critics_score + imdb_rating + genre + top200_box +
    mpaa_rating, data = movies)
summary(fullmodel)$adj.r.squared
```

```
## [1] 0.7628812
```

```
# initial full model gave high p value for mpaa_rating. removed and
# re-evaluated model.
m1 <- lm(audience_score ~ critics_score + imdb_rating + genre + top200_box, data = movies)
summary(m1)$adj.r.squared
```

```
## [1] 0.7640965
```

```
# removing mpaa-rating gave a higher R^2 value.  Proceeded to remove top200_box
# variable to determine effect on R^2
```

```
m2 <- lm(audience_score ~ critics_score + imdb_rating + genre + mpaa_rating, data = movies)
summary(m2)$adj.r.squared
```

```
## [1] 0.7631403
```

```
# Removed genre to determine effect on R^2 value
m3 <- lm(audience_score ~ critics_score + imdb_rating + mpaa_rating + top200_box,
    data = movies)
summary(m3)$adj.r.squared
```

```
## [1] 0.7522014
```

```
# model with critic score, imdb rating, genre and top 200 box gave highest R^2
# value. further refined to see if R^2 will increase.

# removetop200box
m1r <- lm(audience_score ~ critics_score + imdb_rating + genre, data = movies)
summary(m1r)$adj.r.squared
```

```
## [1] 0.7643319
```

```
# remove genre
m1r2 <- lm(audience_score ~ critics_score + imdb_rating + top200_box, data = movies)
summary(m1r2)$adj.r.squared
```

```
## [1] 0.7513364
```

```
# remove imdb rating
m1r3 <- lm(audience_score ~ critics_score + genre + top200_box, data = movies)
summary(m1r3)$adj.r.squared
```

```
## [1] 0.5204078
```

```
# remove critics score
m1r4 <- lm(audience_score ~ imdb_rating + genre + top200_box, data = movies)
summary(m1r4)$adj.r.squared
```

```
## [1] 0.7609622
```

```
# model with critic score, imdb rating and genre gave highest R^2 value conduct
# final backwards selection to see if removing any variable will increase R^2

summary(lm(audience_score ~ critics_score + imdb_rating, data = movies))$adj.r.squared
```

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

```
summary(lm(audience_score ~ critics_score + genre, data = movies))$adj.r.squared
```

```
## [1] 0.5196506
```

```
summary(lm(audience_score ~ genre + imdb_rating, data = movies))$adj.r.squared
```

```
## [1] 0.7611065
```

Final model for predicting audience score included the following variables: - IMDB rating - Critic score on Rotten Tomatoes - Genre

```
summary(m1r)
```

```
##
## Call:
```

```
## lm(formula = audience_score ~ critics_score + imdb_rating + genre,
##     data = movies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.708  -6.446   0.614   5.479  50.111
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -37.15275    3.10900 -11.950  < 2e-16 ***
## critics_score                   0.06687    0.02142   3.122 0.001879 **
## imdb_rating                    14.76644    0.57286  25.777  < 2e-16 ***
## genreAnimation                  9.11698    3.49847   2.606 0.009375 **
## genreArt House & International  0.03008    2.90452   0.010 0.991740
## genreComedy                     2.09167    1.61412   1.296 0.195493
## genreDocumentary                1.19414    1.96833   0.607 0.544281
## genreDrama                     -0.20316    1.38018  -0.147 0.883022
## genreHorror                    -5.02795    2.38740  -2.106 0.035591 *
## genreMusical & Performing Arts  4.39791    3.13828   1.401 0.161588
## genreMystery & Suspense        -6.25279    1.77914  -3.515 0.000472 ***
## genreOther                      1.58228    2.76266   0.573 0.567024
## genreScience Fiction & Fantasy -0.29079    3.50319  -0.083 0.933872
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.817 on 638 degrees of freedom
## Multiple R-squared:  0.7687, Adjusted R-squared:  0.7643
## F-statistic: 176.7 on 12 and 638 DF,  p-value: < 2.2e-16
```

Prediction of audience score for a 2016 movie

With the model, I will predict the audience score of a movie that was released in 2016. To do so, a new data set was created to determine if and what movies from 2016 are already included in the original data set.

```
movies2016 <- movies %>%
    filter(thtr_rel_year == 2016)
sum(movies2016)
```

```
## [1] 0
```

No movies from 2016 were found. As a result, any movie released in 2016 can be used. Finding Dory was chosen, which has the following attributes: - IMDB rating = 7.3 - Critic score on Rotten Tomatoes = 94 - Genre = Animation, Adventure, Comedy

```
findingdory <- data.frame(imdb_rating = 7.3, critics_score = 94, genre = "Animation")
predict(m1r, findingdory, interval = "prediction", level = 0.95)
```

```
##        fit      lwr      upr
## 1 86.04477 65.68079 106.4087
```

The audience score on Rotten Tomatoes for Finding Dory was of 15 July 2022 was 84%. The predicted value of an audience score of 86% is well within the 95% confidence interval calculated.

Something worth noting is that movies can be classified under multiple genres, which can bias the predicted audience score. The way that movies were categorized in the original data set may have also biased the resulting multiple linear model produced in this analysis.

A worthwhile exercise was conducted to see how the predicted audience score changed when the genre of Finding Dory changed as well.

```
# Prediction if Finding Dory is an adventure film
findingdoryadv <- data.frame(imdb_rating = 7.3, critics_score = 94, genre = "Action & Adventure")
predict(m1r, findingdoryadv, interval = "prediction", level = 0.95)
```

```
##       fit     lwr     upr
## 1 76.9278 57.4395 96.4161
```

```
# Prediction if Finding Dory is a comedy film
findingdorycom <- data.frame(imdb_rating = 7.3, critics_score = 94, genre = "Comedy")
predict(m1r, findingdorycom, interval = "prediction", level = 0.95)
```

```
##        fit      lwr      upr
## 1 79.01946 59.56649 98.47244
```

The calculated audience score values when Finding Dory is labelled as an Action & Adventure or a Comedy both produced lower values than the actual audience score. However, the observed audience score on Rotten Tomatoes is still within the 95% confidence interval calculated.

Conclusion

In conclusion, a multiple linear regression model was created to predict audience scores on Rotten Tomatoes for a given movie produced and released before 2016. The model should not be generalized to movies past 2016 since data was not extensively analyzed for movies released after 2016. However, the model does show capacity to predict audience scores on Rotten Tomatoes close to the observed values, as seen with the Finding Dory example.

In doing the analysis, a source of bias that was not initially obvious emerged. Specifically, when characterizing a movie by its genre, the data set was limited in the fact that a movie could only take 1 genre, when in reality, movies may be characterized by more than 1 genre.

Overall, the model showed that the IMDB rating and the critic score on Rotten Tomatoes were the most influential predictors in determining what the audience score on Rotten Tomatoes would be.