Modeling and prediction for movies

Setup

Load packages

```
library(ggplot2)
library(dplyr)
library(statsr)
library(GGally)
library(knitr)
library(gridExtra)
```

Load data

Make sure your data and R Markdown files are in the same directory. When loaded your data file will be called `movies`. Delete this note when before you submit your work.

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

Part 1: Data

What is this data about?

- The dataset contains a list of randomly sampled movies from the IMDB and Rotten Tomatoe movie databases.
- It contains information about both audience and critic scores about how much they like/dislike movies. As well as other variables associated with the movie.
- The dataset contains 651 rows and 32 variables.
- The codebook can be viewed from this link: Code Book

Sampling:

- **Random Sampling** was used in the selection of the movies from the IMDB and Rotten Tomatoe databases.

Generalizability and Association vs Causation

Generalizability :

- The survey uses data from the Rotten Tomatoes and IMDB databases.
- These are some of the biggest databases in the world, that has data on a huge repository of movies from around the world.
- However, on closer inspection. The data about the movies in the sample contain only English language movies.

Answer :

- So we can safely say that the sample data is generalizable **only** to English language movies from the IMDB and Rotten Tomato database.

Causality :

- The data is **Associative** at the moment.
- However, controlled random assignment of new movies to any model generated, can be used to establish **causal** relationships between **response** and **explanatory** variables.

Answer :

- The data is **Associative** at the moment.
- **However** ,there is a possibilty of drawing conclusions on **causality** only after conducting many simulations on generated models to figure out there is strong evidence of **causality** between the **response** and **explanatory** variables used to create the model.

Part 2: Research question

What are the best predictor variables to create an accurate model to predict an IMDB score ?

- Real life modelling involves,looking at lots of variables from different sources and piecing them together to make a robust model that can predict an acurate outcome at a specified confidence interval.
- I am interested in creating a predictor with a 95% confidence interval to predict the IMDB score of a movie, based on various predictor variables.
- I am planning to create the most efficient parsimonious model, which can explain most of the variability in the IMDB ratings.
- **In order to achieve this , I will be using a forward selection approach, with emphasis on getting the best adjusted R-squared value.**

Part 3: Exploratory data analysis

Step1 :

We begin by having an overview of the model:

```
str(movies)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    651 obs. of  32 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 ...
##  $ mpaa_rating     : Factor w/ 6 levels "G","NC-17","PG",..: 5 4 5 3 5 6 4 5 6 6 ...
##  $ studio          : Factor w/ 211 levels "20th Century Fox",..: 91 202 167 34 13 163 147 118 88 84
##  $ thtr_rel_year   : num  2013 2001 1996 1993 2004 ...
##  $ thtr_rel_month  : num  4 3 8 10 9 1 1 11 9 3 ...
##  $ thtr_rel_day    : num  19 14 21 1 10 15 1 8 7 2 ...
##  $ dvd_rel_year    : num  2013 2001 2001 2001 2005 ...
##  $ dvd_rel_month   : num  7 8 8 11 4 4 2 3 1 8 ...
##  $ dvd_rel_day     : num  30 28 21 6 19 20 18 2 21 14 ...
##  $ 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_nom    : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ 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 ...
##  $ top200_box      : Factor w/ 2 levels "no","yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ director        : chr  "Michael D. Olmos" "Rob Sitch" "Christopher Guest" "Martin Scorsese" ...
##  $ actor1          : chr  "Gina Rodriguez" "Sam Neill" "Christopher Guest" "Daniel Day-Lewis" ...
##  $ actor2          : chr  "Jenni Rivera" "Kevin Harrington" "Catherine O'Hara" "Michelle Pfeiffer" .
##  $ actor3          : chr  "Lou Diamond Phillips" "Patrick Warburton" "Parker Posey" "Winona Ryder" .
##  $ actor4          : chr  "Emilio Rivera" "Tom Long" "Eugene Levy" "Richard E. Grant" ...
##  $ actor5          : chr  "Joseph Julian Soria" "Genevieve Mooy" "Bob Balaban" "Alec McCowen" ...
##  $ imdb_url        : chr  "http://www.imdb.com/title/tt1869425/" "http://www.imdb.com/title/tt020587
##  $ rt_url          : chr  "//www.rottentomatoes.com/m/filly_brown_2012/" "//www.rottentomatoes.com/m/
```

Result:

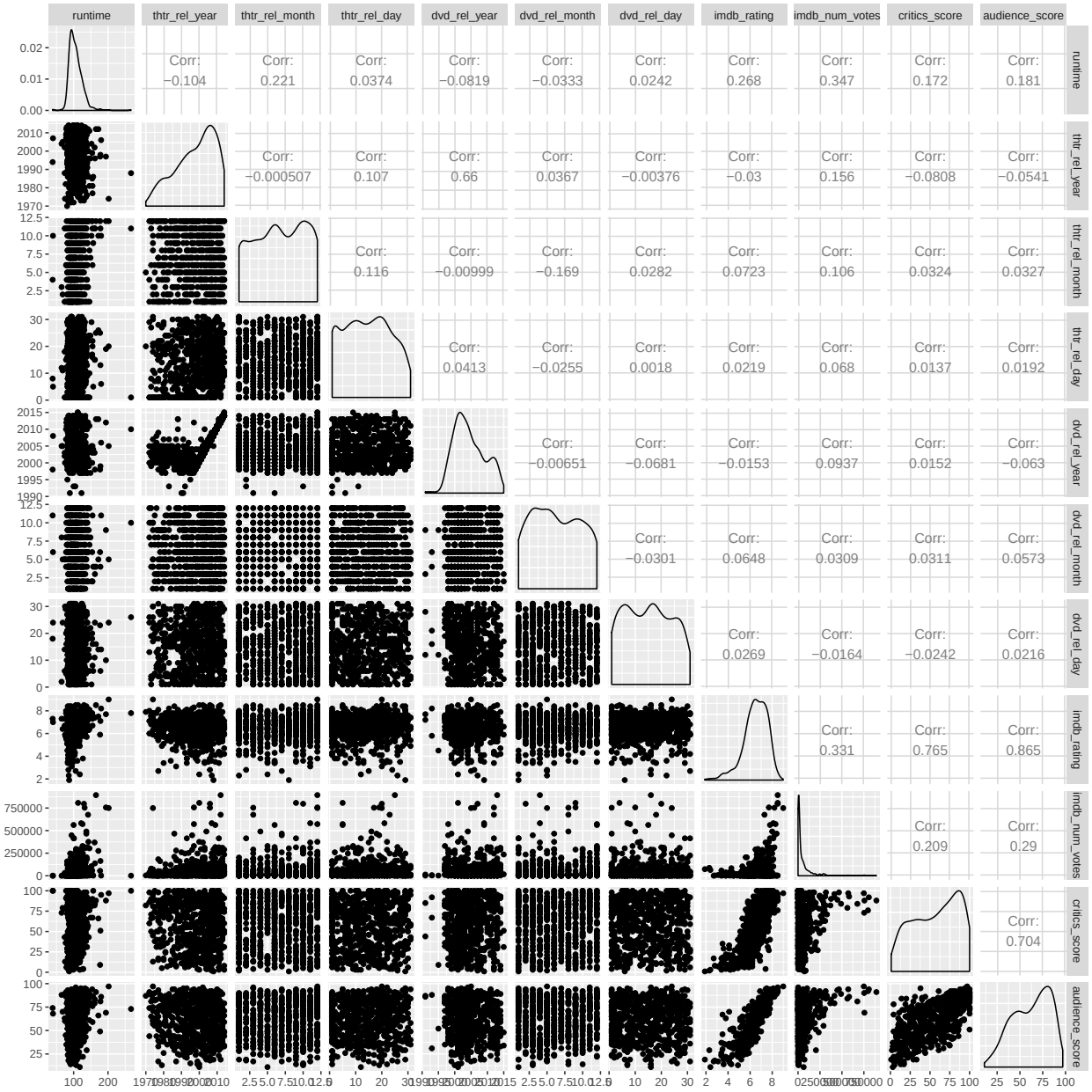
- We see there are **10 numeric variables** in the dataset:
- title,actor1-actor5 , imdb_url, rt_url are **9 character variables**.
- there are **12 factor variables** in the dataset.
- there is **1 integer variable** imdb_num_votes.

Step2 :

Let us now plot a ggpairs plot of all the numeric variables in our dataset.

```
m1 <- movies%>%select(c(runtime,thtr_rel_year,thtr_rel_month,thtr_rel_day,dvd_rel_year,dvd_rel_month,dv

ggpairs(data = m1)
```



ANOVA

```
AnoV <- aov(imdb_rating~imdb_num_votes+critics_score+audience_score+runtime+thtr_rel_year+thtr_rel_month
summary(AnoV)
```

```
##                  Df Sum Sq Mean Sq  F value   Pr(>F)
## imdb_num_votes    1   82.8    82.8  368.243  < 2e-16 ***
## critics_score     1  370.2   370.2 1647.482  < 2e-16 ***
## audience_score    1  139.1   139.1  619.149  < 2e-16 ***
## runtime           1    4.6     4.6   20.292 7.92e-06 ***
## thtr_rel_year     1    0.5     0.5    2.247    0.134
## thtr_rel_month    1    0.3     0.3    1.124    0.289
## thtr_rel_day      1    0.0     0.0    0.059    0.808
## dvd_rel_year      1    0.0     0.0    0.048    0.827
## dvd_rel_month     1    0.5     0.5    2.018    0.156
## dvd_rel_day       1    0.3     0.3    1.327    0.250
## Residuals       631  141.8     0.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 9 observations deleted due to missingness
```

Multiple R-squared: 0.8083808

Adjusted R-squared: 0.805344

- We observe from ANOVA: critics_score, audience_score, imdb_num_votes, runtime. Contribute maximum to the variability, due to ther high values.
- We will rebuild a model with only these 4, to see what the R values look like.

```
AnoV <- aov(imdb_rating~imdb_num_votes+critics_score+audience_score+runtime, data = m1)
summary(AnoV)
```

```
##                  Df Sum Sq Mean Sq F value   Pr(>F)
## imdb_num_votes    1   84.3    84.3  372.74  < 2e-16 ***
## critics_score     1  386.0   386.0 1707.26  < 2e-16 ***
## audience_score    1  143.1   143.1  633.04  < 2e-16 ***
## runtime           1    4.6     4.6   20.33 7.74e-06 ***
## Residuals       645  145.8     0.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 1 observation deleted due to missingness
```

Multiple R-squared: 0.8090797

Adjusted R-squared: 0.8078957

Result: Why did we pick only these 4 numeric variables

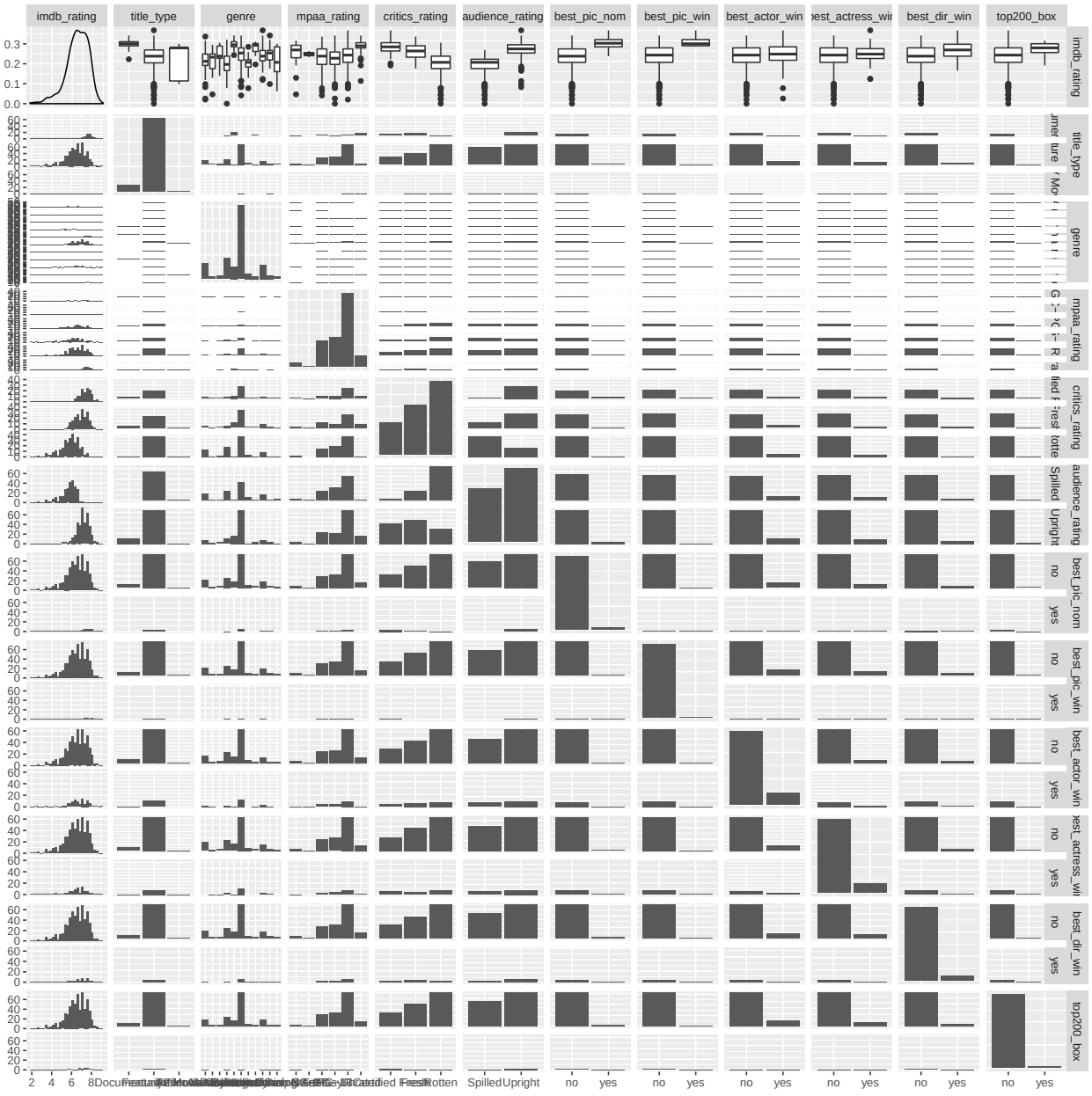
- We see that this parsimonious model has a better overall adjusted R-squared
- Therefore, these 4 numeric variables will be analyzed in depth, to build out model.
- Also these 4 numeric variables have the highest correlation coefficients, which indicated they have a more linear relationship. The other numeric variables can be dropped from consideration since they do not have strong linear relationsips.
- I am **NOT** taking the critics_score and audience_score collinearity. Because there has been times when critics and audiences have diverging scores. So even though the correlation seems high, based on experience. I am not going to use collineartiy to drop one of them. Besides there must be some really important information that seems to superceded the collinearity condition, as the adjusted_R squared value has actually gone up substantially.

Step3 :

plot IMDB rating against the 11 factor variables(excluding studio) to see if we can learn something:

```
m2 <- movies%>%select(c(imdb_rating,title_type,genre,mpaa_rating,critics_rating,audience_rating,best_pi

ggpairs(data = m2)
```



Result:

- Looking at the first row, we can see that most categorical variables exhibit a high degree of variance against the imdb_rating. This is something interesting that needs to be noted. the 3 that seem to

have the least variance are: best_actor_win, best_actress_win, best_director_win.

ANOVA and adjusted-R-Squared: imdb_rating vs title_type

```
AnoV <- aov(imdb_rating~title_type, data = m2)
summary(AnoV)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## title_type    2   83.7   41.84    39.8 <2e-16 ***
## Residuals   648  681.2    1.05
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Result:

Multiple R-squared: 0.1094114

Adjusted R-squared: 0.1066627

Note adjusted R-squared has not dropped significantly, this seems like a good candidate for our model

---

ANOVA and adjusted-R-Squared: imdb_rating vs genre

```
AnoV <- aov(imdb_rating~genre, data = m2)
summary(AnoV)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## genre        10  174.5  17.446   18.91 <2e-16 ***
## Residuals   640  590.4   0.922
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Result:

Multiple R-squared: 0.2281009

Adjusted R-squared: 0.21604

Note adjusted R-squared has not dropped significantly, this seems like a good candidate for our model

---

ANOVA and adjusted-R-Squared: imdb_rating vs mpaa_rating

```
AnoV <- aov(imdb_rating~mpaa_rating, data = m2)
summary(AnoV)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## mpaa_rating   5   55.8  11.157   10.15 2.26e-09 ***
## Residuals   645  709.1   1.099
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Result:

Multiple R-squared: 0.0729363

Adjusted R-squared: 0.0657497

Note mpaa_rating seems to have a verly low R-squared and an even lower adjusted R-squared. We may ignore this while buiilding the model.

ANOVA and adjusted-R-Squared: imdb_rating vs critics_rating

```
AnoV <- aov(imdb_rating~critics_rating, data = m2)
summary(AnoV)
```

```
##                 Df Sum Sq Mean Sq F value Pr(>F)
## critics_rating   2  309.3   154.7     220 <2e-16 ***
## Residuals      648  455.5     0.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Result:

Multiple R-squared: 0.4044055

Adjusted R-squared: 0.4025673

Note adjusted R-squared has not dropped significantly, this seems like a good candidate for our model

---

ANOVA and adjusted-R-Squared: imdb_rating vs audience_rating

```
AnoV <- aov(imdb_rating~audience_rating, data = m2)
summary(AnoV)
```

```
##                  Df Sum Sq Mean Sq F value Pr(>F)
## audience_rating   1  369.6   369.6     607 <2e-16 ***
## Residuals       649  395.2     0.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Result:

Multiple R-squared: 0.4832746

Adjusted R-squared: 0.4824785

Note adjusted R-squared has not dropped significantly, this seems like a good candidate for our model

---

ANOVA and adjusted-R-Squared: imdb_rating vs best_pic_nom

```
AnoV <- aov(imdb_rating~best_pic_nom, data = m2)
summary(AnoV)
```

```
##               Df Sum Sq Mean Sq F value   Pr(>F)
## best_pic_nom   1   36.0   35.97   32.03 2.28e-08 ***
## Residuals    649  728.9    1.12
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lm2_1 <- lm(imdb_rating~best_pic_nom, data = m2)
summary(lm2_1)
```

```
##
## Call:
## lm(formula = imdb_rating ~ best_pic_nom, data = m2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5491 -0.5491  0.0509  0.7509  2.0509
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      6.44913    0.04225  152.62  < 2e-16 ***
## best_pic_nomyes  1.30087    0.22986    5.66 2.28e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.06 on 649 degrees of freedom
## Multiple R-squared:  0.04703,    Adjusted R-squared:  0.04556
## F-statistic: 32.03 on 1 and 649 DF,  p-value: 2.279e-08
```

Note adjusted R-squared has not dropped significantly, this seems like a good candidate for our model

---

ANOVA and adjusted-R-Squared: imdb_rating vs best_pic_win

```
AnoV <- aov(imdb_rating~best_pic_win, data = m2)
summary(AnoV)
```

```
##               Df Sum Sq Mean Sq F value   Pr(>F)
## best_pic_win   1   14.0  14.006   12.11 0.000536 ***
## Residuals    649  750.8   1.157
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lm2_1 <- lm(imdb_rating~best_pic_win, data = m2)
summary(lm2_1)
```

```
##
## Call:
## lm(formula = imdb_rating ~ best_pic_win, data = m2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5778 -0.5778  0.1222  0.8222  2.0222
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      6.47780    0.04238 152.834  < 2e-16 ***
## best_pic_winyes  1.42220    0.40874   3.479 0.000536 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.076 on 649 degrees of freedom
## Multiple R-squared:  0.01831,    Adjusted R-squared:  0.0168
## F-statistic: 12.11 on 1 and 649 DF,  p-value: 0.000536
```

Note This seems like a decent candidate, but might be dropped as adjusted R-squared is quite small

---

ANOVA and adjusted-R-Squared: imdb_rating vs best_actor_win

```
AnoV <- aov(imdb_rating~best_actor_win, data = m2)
summary(AnoV)
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)
## best_actor_win   1    3.2   3.189   2.717 0.0998 .
## Residuals      649  761.7   1.174
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lm2_1 <- lm(imdb_rating~best_actor_win, data = m2)
summary(lm2_1)
```

```
##
## Call:
## lm(formula = imdb_rating ~ best_actor_win, data = m2)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -4.5645 -0.5645  0.0355  0.8355  2.3355
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        6.46452    0.04586 140.961   <2e-16 ***
## best_actor_winyes  0.20000    0.12134   1.648   0.0998 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.083 on 649 degrees of freedom
## Multiple R-squared:  0.004169,   Adjusted R-squared:  0.002635
## F-statistic: 2.717 on 1 and 649 DF,  p-value: 0.09977
```

Note adjusted R-squared is too small to be of significance, this looks to be definitely dropped. This corresponds to the lack of noticable variance in the box-plots.

ANOVA and adjusted-R-Squared: imdb_rating vs best_actress_win

```
AnoV <- aov(imdb_rating~best_actress_win, data = m2)
summary(AnoV)
```

```
##                   Df Sum Sq Mean Sq F value Pr(>F)
## best_actress_win   1    3.9   3.897   3.324 0.0687 .
## Residuals        649  760.9   1.172
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lm2_1 <- lm(imdb_rating~best_actress_win, data = m2)
summary(lm2_1)
```

```
##
## Call:
## lm(formula = imdb_rating ~ best_actress_win, data = m2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5658 -0.5658  0.1342  0.8342  2.2875
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           6.4658     0.0450 143.684   <2e-16 ***
## best_actress_winyes   0.2467     0.1353   1.823   0.0687 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.083 on 649 degrees of freedom
## Multiple R-squared:  0.005096,   Adjusted R-squared:  0.003563
## F-statistic: 3.324 on 1 and 649 DF,  p-value: 0.06874
```

Note adjusted R-squared is too small to be of any value, we will ignore this while building the model

---

ANOVA and adjusted-R-Squared: imdb_rating vs best_dir_win

```
AnoV <- aov(imdb_rating~best_dir_win, data = m2)
summary(AnoV)
```

```
##               Df Sum Sq Mean Sq F value   Pr(>F)
## best_dir_win   1   13.9  13.865   11.98 0.000572 ***
## Residuals    649  751.0   1.157
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lm2_1 <- lm(imdb_rating~best_dir_win, data = m2)
summary(lm2_1)
```

```
##
## Call:
## lm(formula = imdb_rating ~ best_dir_win, data = m2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5543 -0.5543  0.0457  0.7519  2.0457
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      6.45428    0.04363 147.948  < 2e-16 ***
## best_dir_winyes  0.58758    0.16974   3.462 0.000572 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.076 on 649 degrees of freedom
## Multiple R-squared:  0.01813,    Adjusted R-squared:  0.01662
## F-statistic: 11.98 on 1 and 649 DF,  p-value: 0.0005722
```

Note adjusted R-squared is really small, we will be dropping this while building the model

---

ANOVA and adjusted-R-Squared: imdb_rating vs top200_box

```
AnoV <- aov(imdb_rating~top200_box, data = m2)
summary(AnoV)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## top200_box    1    6.4   6.425   5.499 0.0193 *
## Residuals   649  758.4   1.169
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lm2_1 <- lm(imdb_rating~top200_box, data = m2)
summary(lm2_1)
```

```
##
## Call:
## lm(formula = imdb_rating ~ top200_box, data = m2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5778 -0.5778  0.1222  0.8222  2.5222
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.47783    0.04286 151.122   <2e-16 ***
## top200_boxyes  0.66217    0.28239   2.345   0.0193 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.081 on 649 degrees of freedom
## Multiple R-squared:  0.008401,   Adjusted R-squared:  0.006873
## F-statistic: 5.499 on 1 and 649 DF,  p-value: 0.01933
```

Note adjusted R-squared os really small. We won't be including this in our model

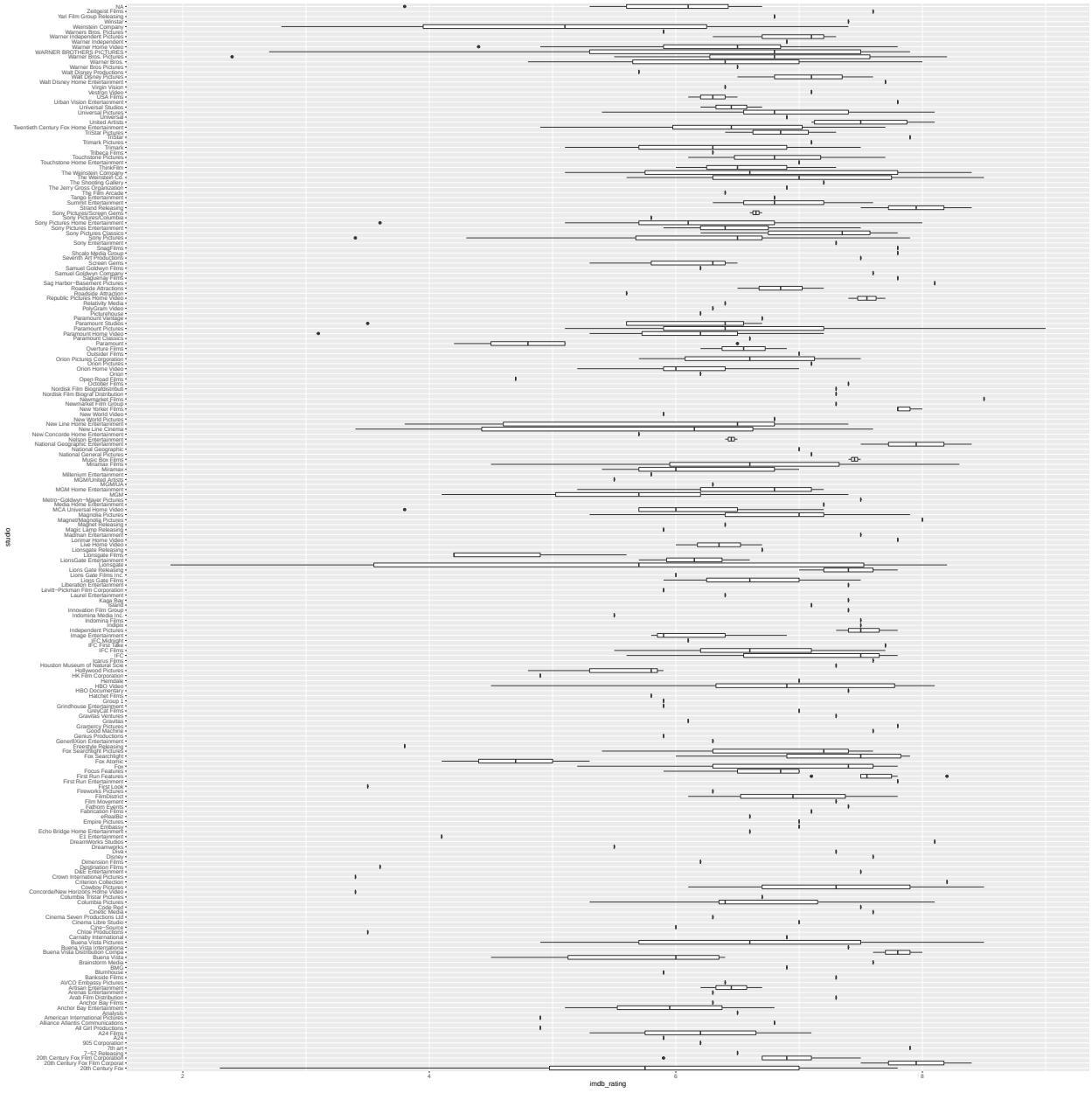
---

Step4 :

Now we can look at the relationship between IMDB_ratings and studios next:

```
m3 <- movies%>%select(c(imdb_rating,studio))
```

```
#m4 <- movies%>%select(c(imdb_rating,title))
ggplot(data = m3,aes(x = studio, y = imdb_rating))+geom_boxplot()+coord_flip()
```




```
##ggplot(data = m4[1:10,], aes(x = title, y = imdb_rating))+geom_boxplot()
```

Let us do an ANOVA to check if there is any relationship, as well as check the adjusted R-squared via a linear model.

```
AnoV <- aov(imdb_rating~studio, data = m3)
summary(AnoV)
```

```
##              Df Sum Sq Mean Sq F value  Pr(>F)
## studio      210  306.5   1.460   1.406 0.00173 **
## Residuals   432  448.6   1.038
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## 8 observations deleted due to missingness
```

```
lm4 <- lm(imdb_rating~studio, data = m3)
summary(lm4)$r.squared
```

```
## [1] 0.4059134
```

```
summary(lm4)$adj.r.squared
```

```
## [1] 0.1171212
```

- boxplot of imdb_rating versus studio shows an interesting variation among the different studios. Certain studios show a skewed distribution with hign medians, with long left or right skewes, certain other studions have a median high imdb_rating, compared to the rest. This box plot has lots of interesting information. And can be it's own topic of research.
- **NOTE** The difference between R-squared of: 0.4059 and adjusted R-squared: 0.1171. Which is very steep. This seems to indicate that adding studio to our model, might actually reduce the overall adjusted-R squared of our model. And it also seems to show that studio is not necessarily a good predictor variable.

Result:

- We see a high degree of variance between the studios with respect to the IMDB_ratings. So it looks like studios might have a strong influence in the imdb_ratings.
- We need to keep this in mind when we build our model.

~~I am interested if the title of a movie, has any influence on audience, affecting the imdb_rating of a movie.~~

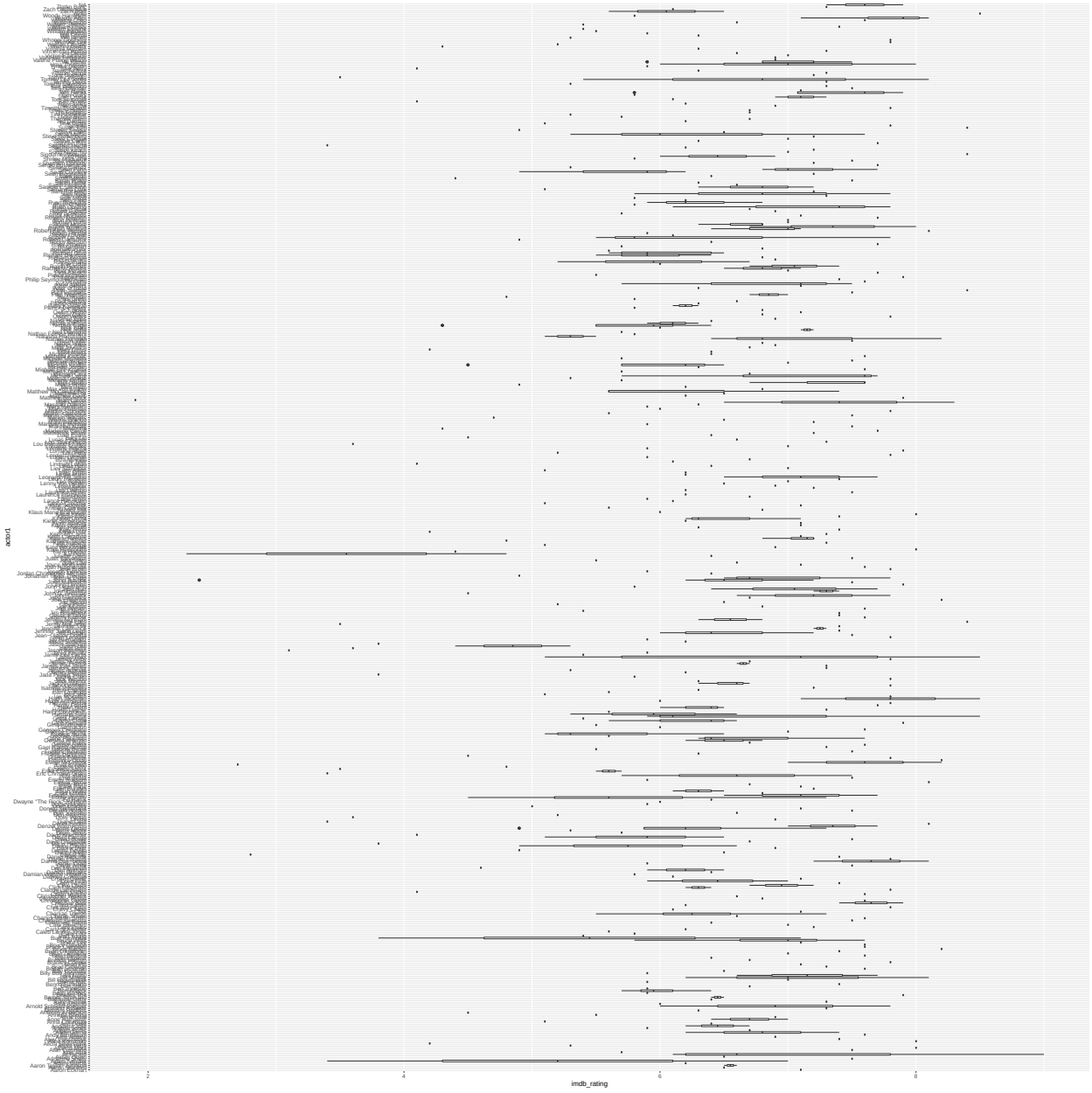
- imdb_url and rt_url, offer no real value, as they merely are locations in the internet of where information about the movies is found.

Step5 :

Influence on lead actor on IMDB_ratings:

- Another factor of interest, is to see if actors have any influence on movie quality thereby influencing the IMDB_rating
- Lead actors can sometimes have a major influence on ratings of movies. I will be exploring the relationship between the **actor1** variable and the **IMDB_ratings** next.

```
m5 <- movies%>%select(c(imdb_rating,actor1))
ggplot(data = m5,aes(x = actor1, y = imdb_rating))+geom_boxplot()+coord_flip()
```



Let us do a more formal check with an ANOVA analysis as well as fit to a linear model just to make sure. Before we get the R squared value.

```
AnoV <- aov(imdb_rating~actor1, data = m5)
summary(AnoV)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## actor1      484  635.9  1.3138   1.706 3.71e-05 ***
## Residuals   164  126.3  0.7703
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 2 observations deleted due to missingness
```

```
lm5 <- lm(imdb_rating~actor1, data = m5)
summary(lm5)$r.squared
```

```
## [1] 0.8342664
```

```
summary(lm5)$adj.r.squared
```

```
## [1] 0.3451501
```

- the boxplot presents lots of interesting information on lead actors and their movies.
- It says quite a bit about what kind of movies that the actors seem to fall into. for example: **Adam Sandler** although famous seems to be starring in movies with really low IMDB ratings. **Arnold Schwazneggar** seems to star in movies with overall higher median IMDB rating.
- Although we see variance, can it be completely attributed to just the actor or to other factors such as the director.
- Nevertheless, this is an interestng variable to consider too.
- **NOTE** One really interesting trend is the difference between R-squared of: 0.8342 and adjusted R-squared: 0.3452. From this we see that adding in more actors into the predictors actually results in a higher penality. So from this it looks like, actor1 is not necessarily a good predictor variable

---

Part 4: Modeling

- We will now begin creating the model by using the forward selection R-squared approach.
- We will add one predictor variable at a time, and check if the R-squared value increases or decreases. Predictor variables that lower R-squared will be dropped.
- Our strategy for adding the predictor variables will follow this strategy:-
- First we will be adding the numeric predictor variables, we will add predictors that result in increased adjustd R-squared. We will then run tests on them to make sure that they satisfy the conditions needed to be added in a MLR model. If they fail to satisfy the conditions, we will remove them from the model.
- We will follow the same approach and add in the categorical variables.

Let us list out their R-squared and Adjusted R-square values:

```
kable(RsquaredSummary)
```

predictorVariable	VariableType	rsquared	adjrsquared
audience_rating	Categorical	0.4832746	0.4824785
critics_rating	Categorical	0.4044055	0.4025673
actor1	Categorical	0.8342664	0.3451501
genre	Categorical	0.2281009	0.2160400
studio	Categorical	0.4059134	0.1171212

predictorVariable	VariableType	rsquared	adjrsquared
title_type	Categorical	0.1094114	0.1066627
mpaa_rating	Categorical	0.0729363	0.0657497
best_pic_nom	Categorical	0.0470320	0.0455636
best_pic_win	Categorical	0.0183129	0.0168003
best_dir_win	Categorical	0.0181285	0.0166156
top200_box	Categorical	0.0084011	0.0068732
best_actress_win	Categorical	0.0050955	0.0035625
best_actor_win	Categorical	0.0041689	0.0026345
audience_score	Numeric	0.7479917	0.7476034
critics_score	Numeric	0.5852793	0.5846403
imdb_num_votes	Numeric	0.1096620	0.1082901
runtime	Numeric	0.0719530	0.0705208

Section 1: Build model with numeric variables

- We will be considering the following numeric_variables, since they have the largest correleation coefficients: **audience_score, critics_score, imdb_num_votes, runtime**

```
model <- lm(imdb_rating~audience_score+critics_score, data = movies)

summary(model)$r.squared
```

```
## [1] 0.7962356
```

```
summary(model)$adj.r.squared
```

```
## [1] 0.7956067
```

```
# p1 <- ggplot(data = model, aes(x = audience_score, y = .resid))+geom_point()+geom_hline(yintercept=0,
#
#
# p2 <- ggplot(data = model, aes(x = .resid)) + geom_histogram() + xlab("Residuals")
# #QQ-plot
# p2_1 <- ggplot(data = model, aes(sample = .resid)) + stat_qq()+stat_qq_line(size=1.25, color="red")
#
# p3 <- ggplot(data = model, aes(x = .fitted, y = .resid))+geom_point()+geom_hline(yintercept=0, linety
# p3_1 <- ggplot(data = model, aes(x = abs(.fitted), y = .resid))+geom_point()+geom_hline(yintercept=0,
#
#
# print(p1)
# print(p2)
# print(p2_1)
# print(p3)
# print(p3_1)


#hist(model$residuals)
#qqnorm(model$residuals)
#qqline(model$residuals)
```

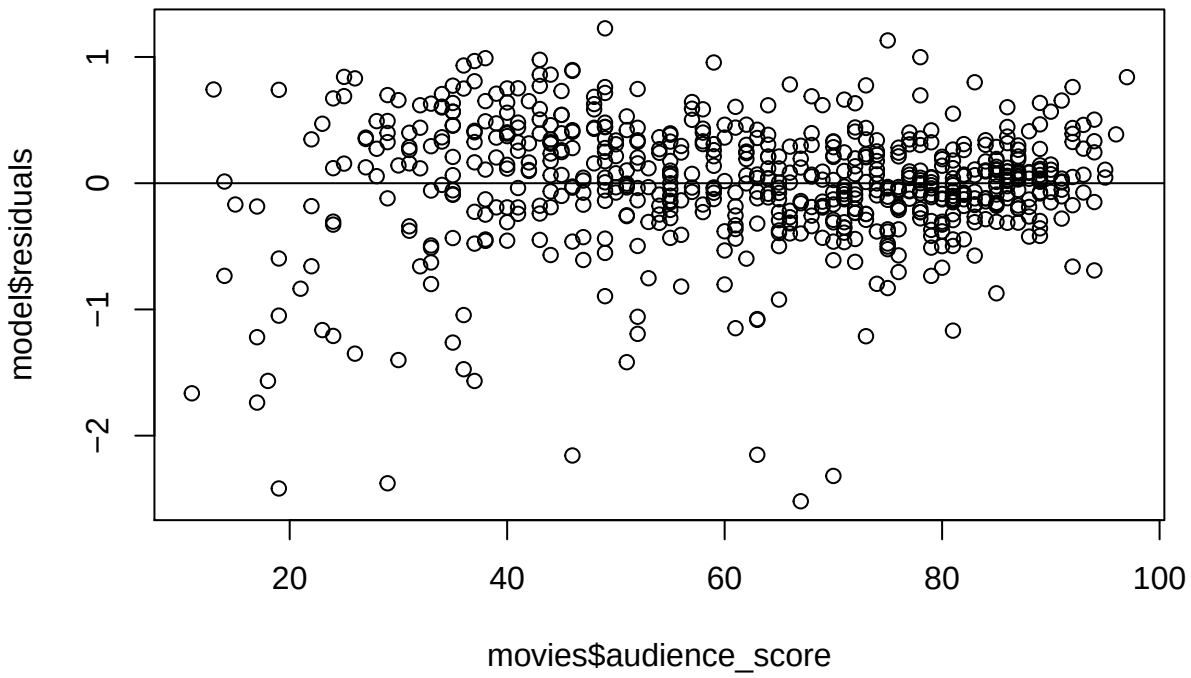
```r
#plot(model$residuals~model$fitted)+abline(h = 0)
#plot(abs(model$residuals)~model$fitted)+abline(h = 0)
#plot(model$residuals)+abline(h = 0)
#modelSummary <- data.frame( modelPredVar = modelPredVar,modelRsquare = modelRsquare ,modeladjRsquare =

summary(model)
```

```
##
## Call:
## lm(formula = imdb_rating ~ audience_score + critics_score, data = movies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.51964 -0.19767  0.03466  0.30671  1.22691
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.647241   0.062471   58.38   <2e-16 ***
## audience_score 0.034703   0.001340   25.90   <2e-16 ***
## critics_score  0.011816   0.000954   12.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4904 on 648 degrees of freedom
## Multiple R-squared:  0.7962, Adjusted R-squared:  0.7956
## F-statistic:  1266 on 2 and 648 DF,  p-value: < 2.2e-16
```

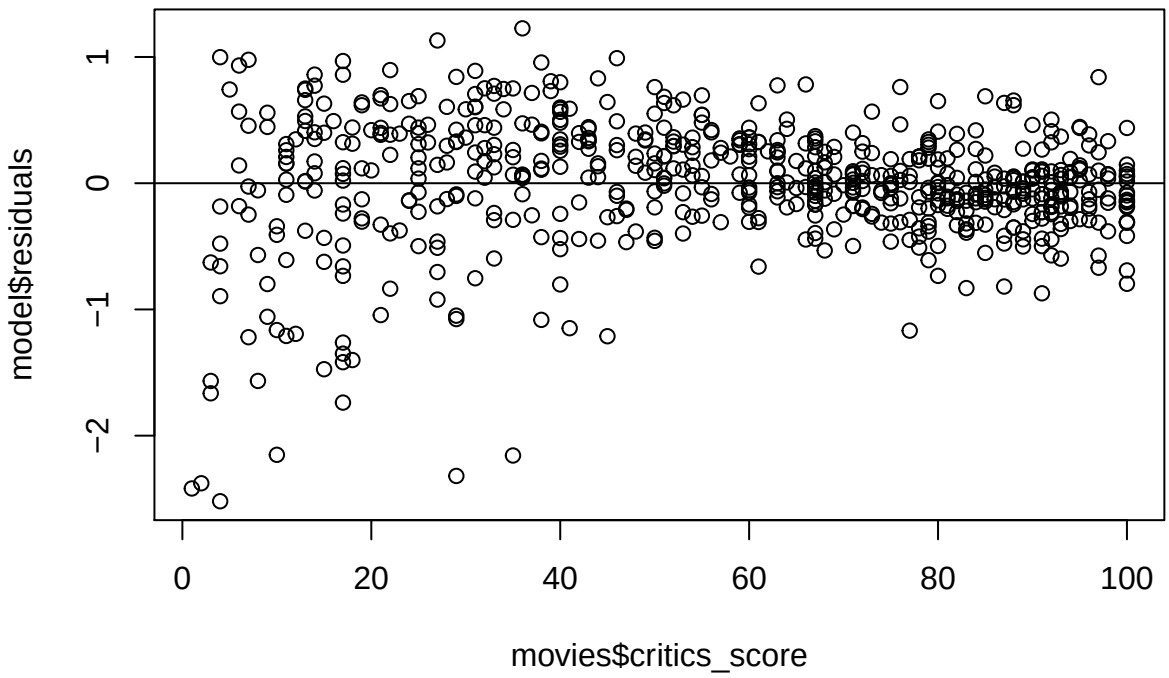
Let us plot all the numeric variables to check for costant variability vs residuals

```r
plot(model$residuals~movies$audience_score)+abline(h = 0)
```



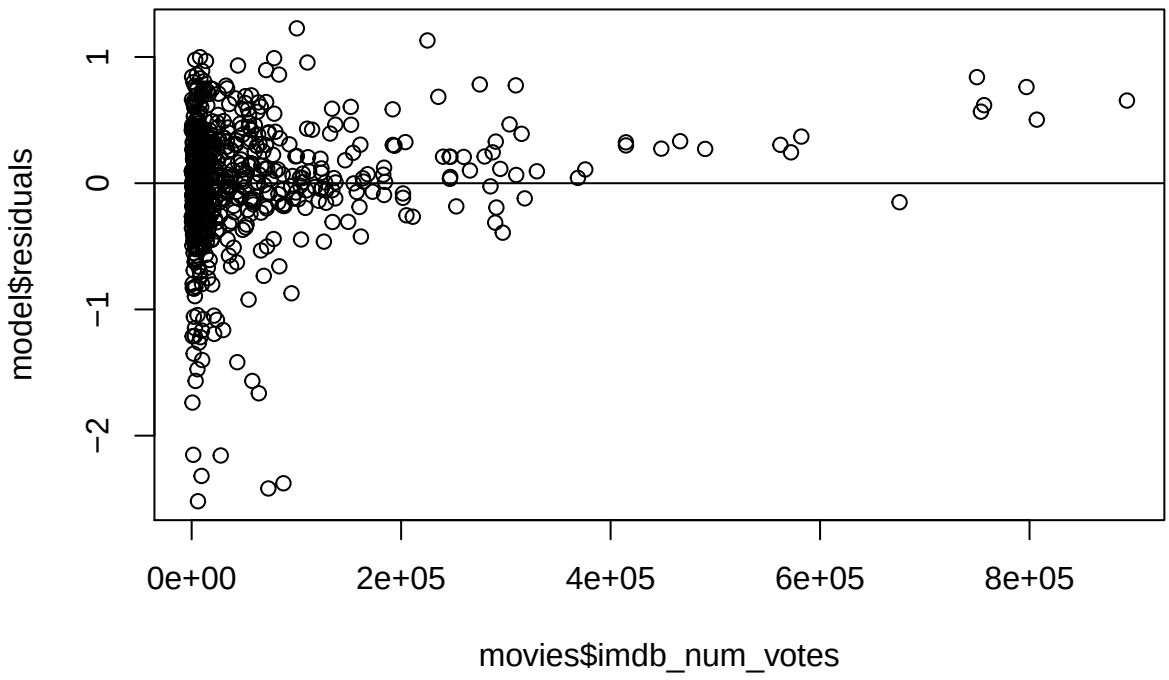

```
## integer(0)
```

```
plot(model$residuals~movies$critics_score)+abline(h = 0)
```



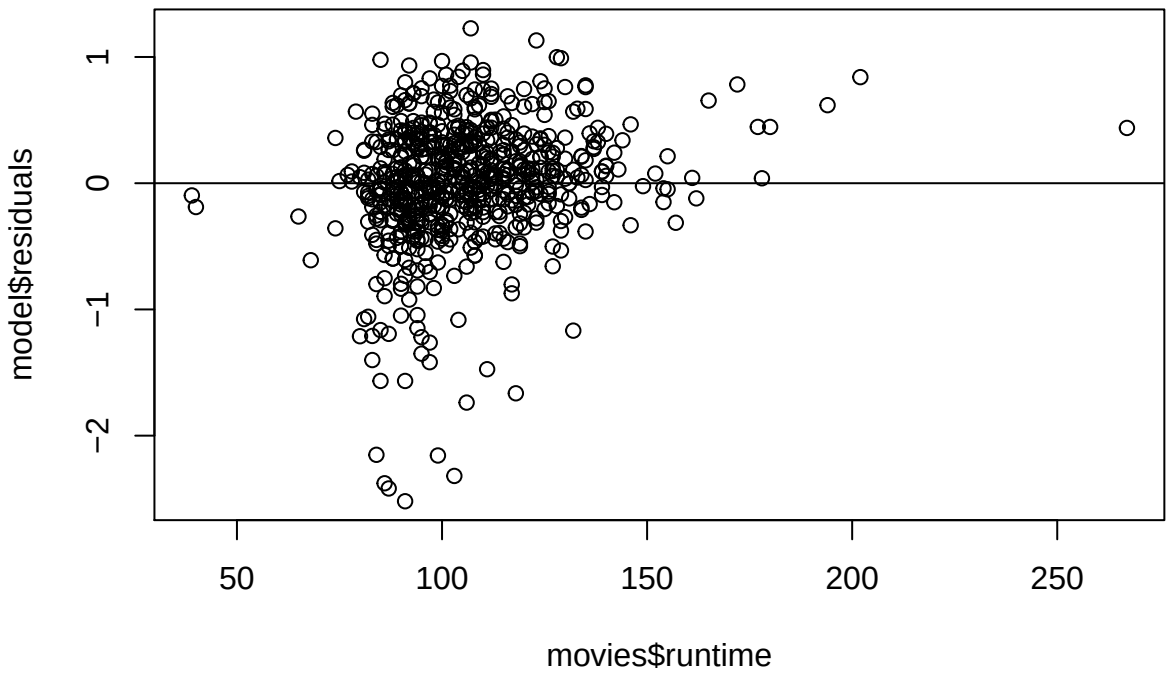

```
## integer(0)
```

```
plot(model$residuals~movies$imdb_num_votes)+abline(h = 0)
```




```
## integer(0)
```

```
plot(model$residuals~movies$runtime)+abline(h = 0)
```




```
## integer(0)
```

- **audience_score** is constatnly variable , so we keep it.
- **critics_score** is constantly variable so we keep it.
- **imdb_num_votes** is not constantly variable so we drop it.
- **runtime** is not constantly variable so we drop it.

```
model <- lm(imdb_rating~audience_score, data = movies)
```

After adding: **audience_score**

the R-squared: 0.7479917

the adjusted R-squared: 0.7476034

```
model <- lm(imdb_rating~audience_score+critics_score, data = movies)
```

After adding: **critics_score**

the R-squared: 0.7962356

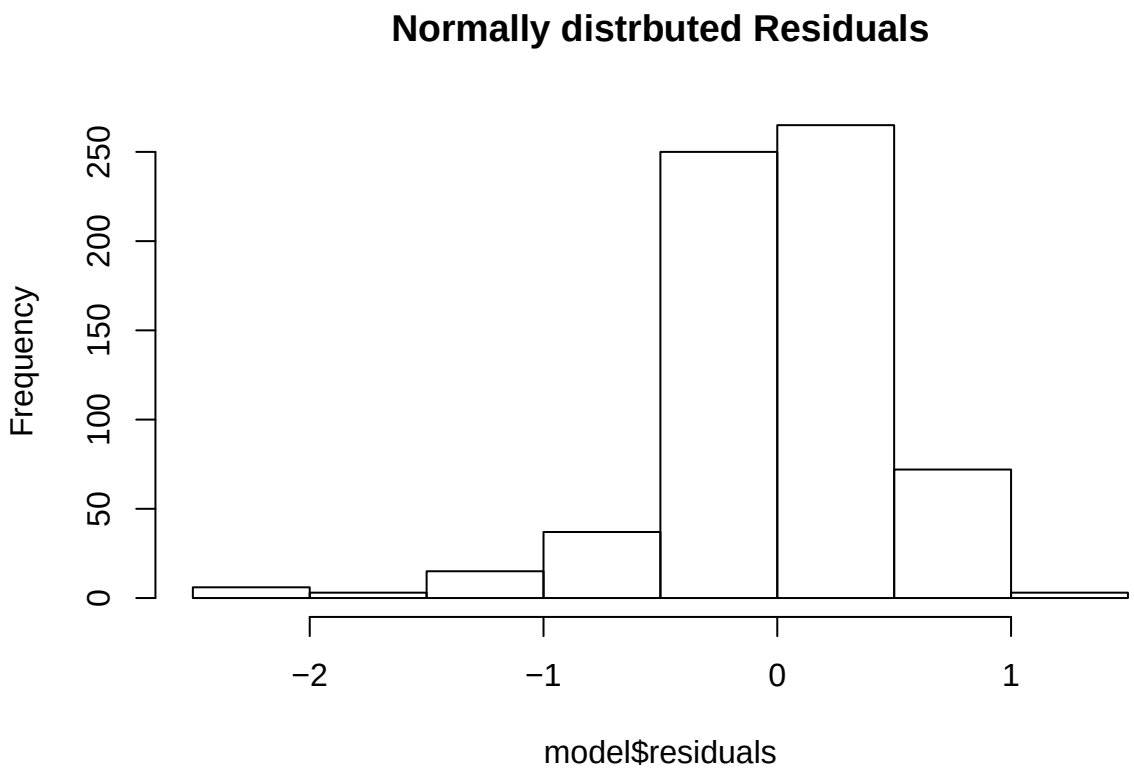
the adjusted R-squared: 0.7956067

The adjusted R-squared has increased, so critics score is a useful predictor variable, along with audience_score

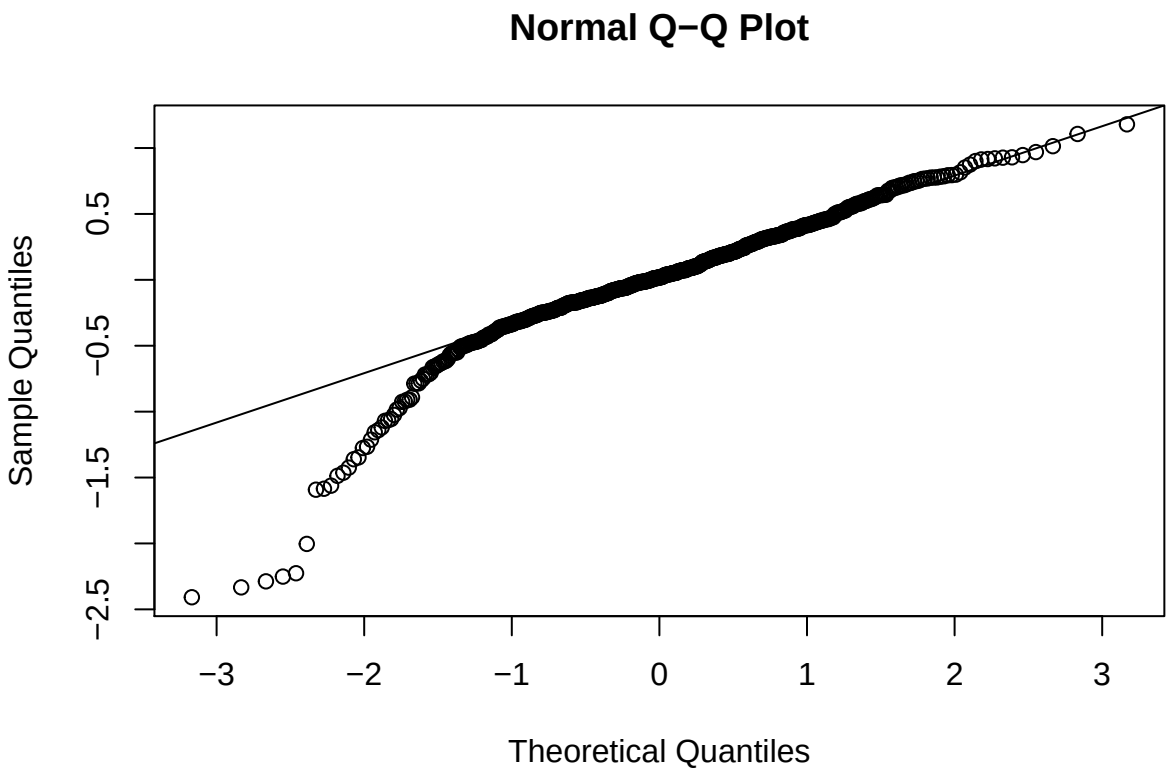
Section 2: Build model with categorical variables

audience_rating

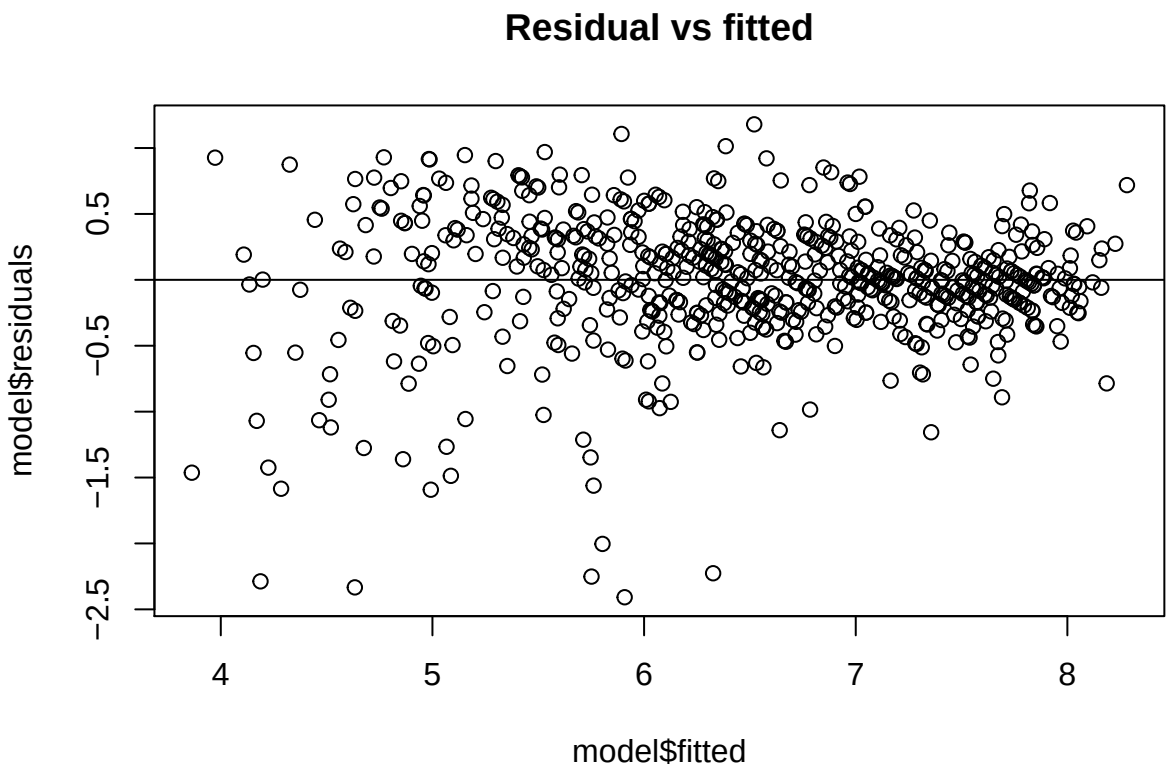
```
model <- lm(imdb_rating~audience_score+critics_score+audience_rating, data = movies)
hist(model$residuals, main = "Normally distrbuted Residuals")
```




```
qqnorm(model$residuals)
qqline(model$residuals)
```

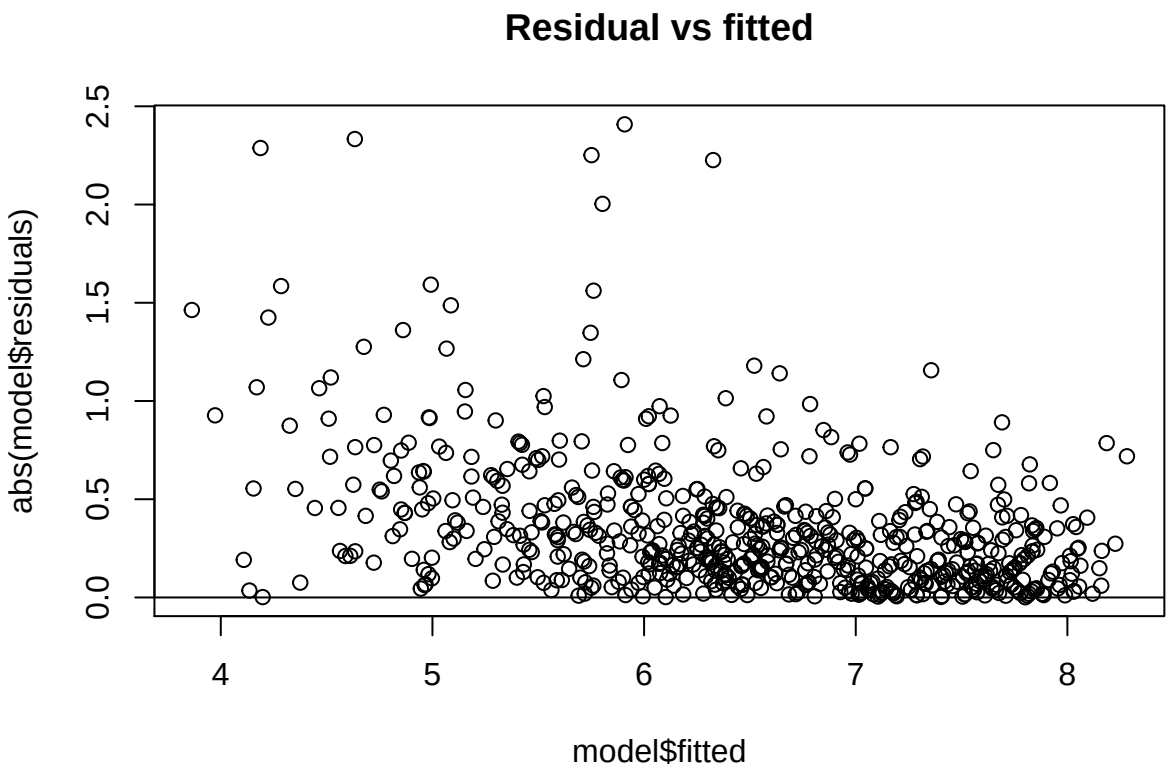



```
plot(model$residuals~model$fitted,main = "Residual vs fitted")+abline(h = 0)
```



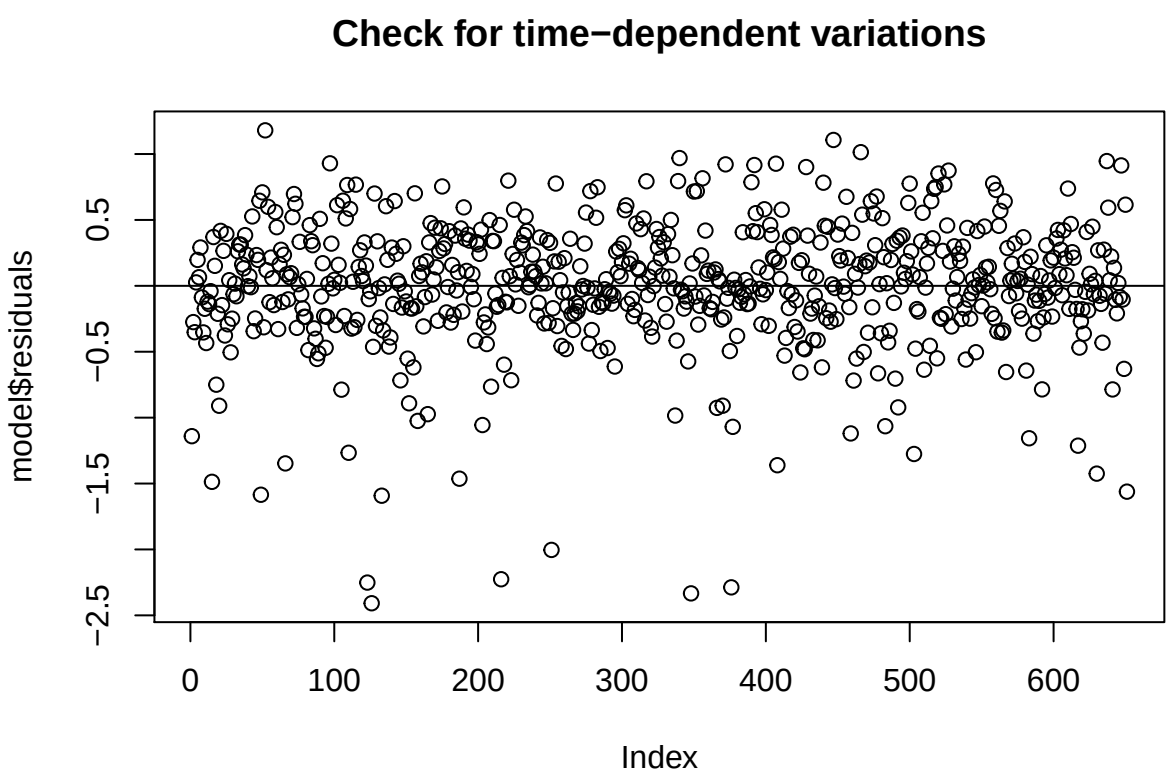

```
## integer(0)
```

```
plot(abs(model$residuals)~model$fitted,main = "Residual vs fitted")+abline(h = 0)
```




```
## integer(0)
```

```
plot(model$residuals,main = "Check for time-dependent variations")+abline(h = 0)
```




```
## integer(0)
```

- From the plots: audience_rating shows: Constant variability. Nearly Normal residuals.

After adding: **audience_rating**

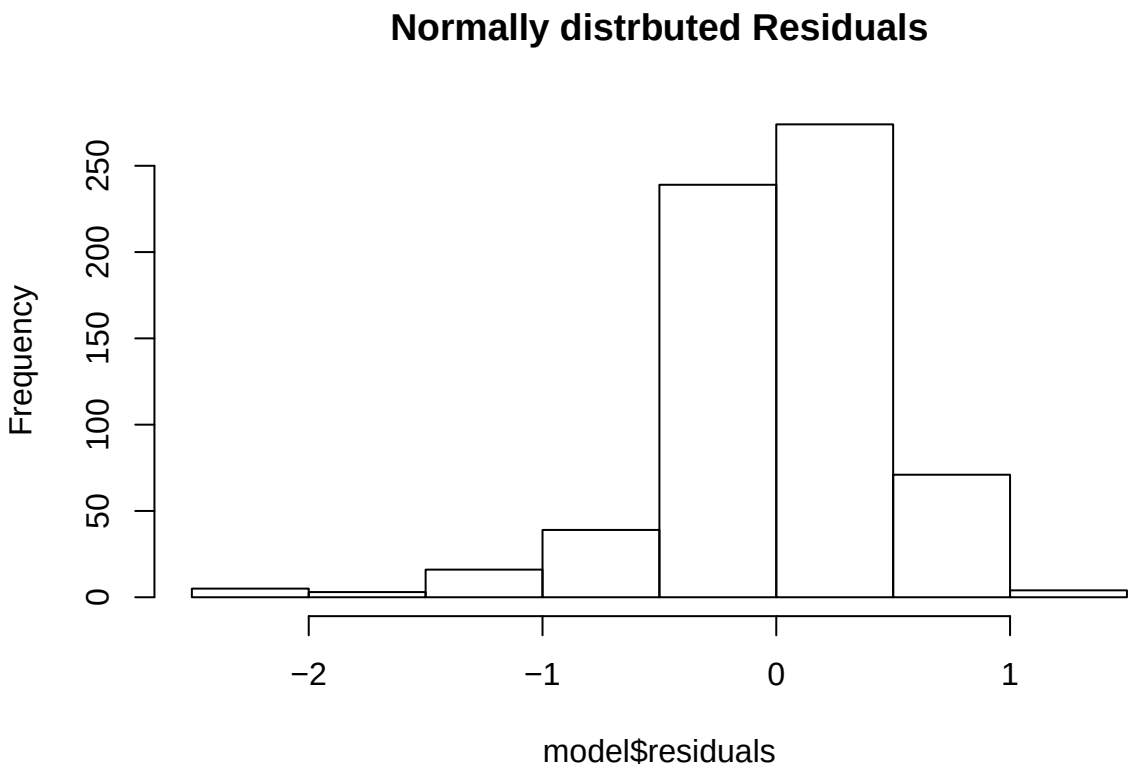
the R-squared: 0.8045298

the adjusted R-squared: 0.8036235

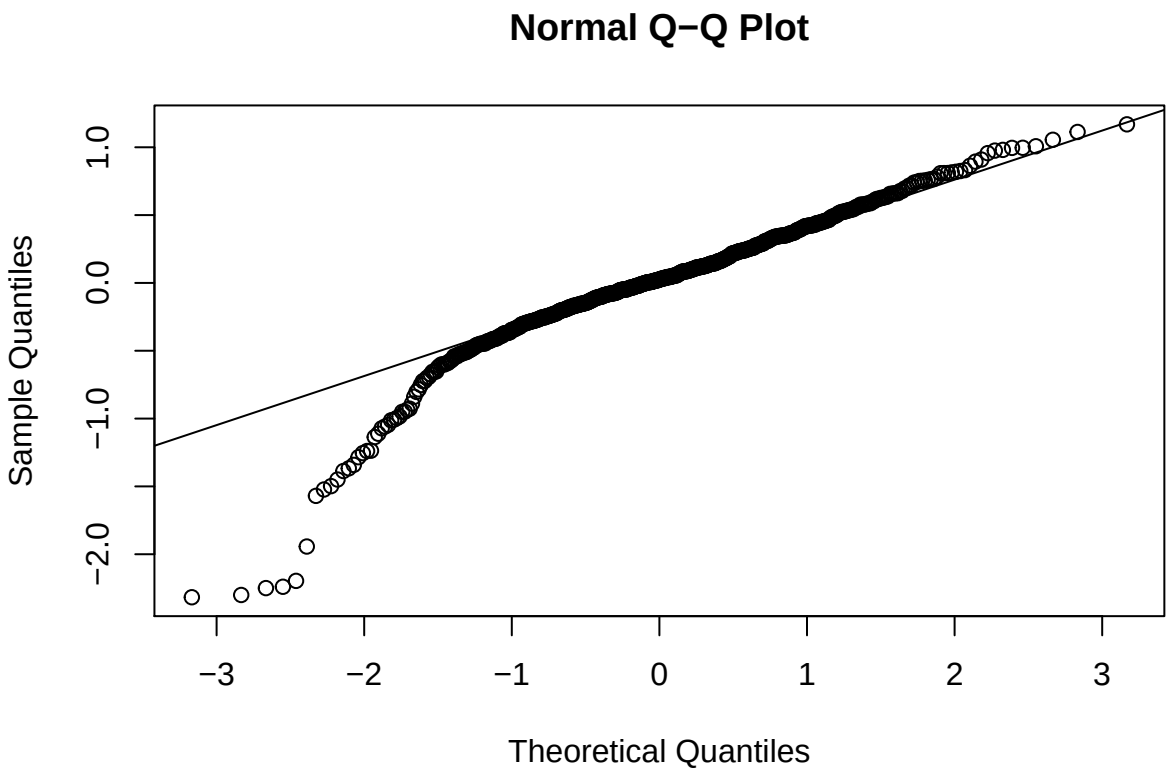
The adjusted- R squared increased so we keep this predictor

critics_rating

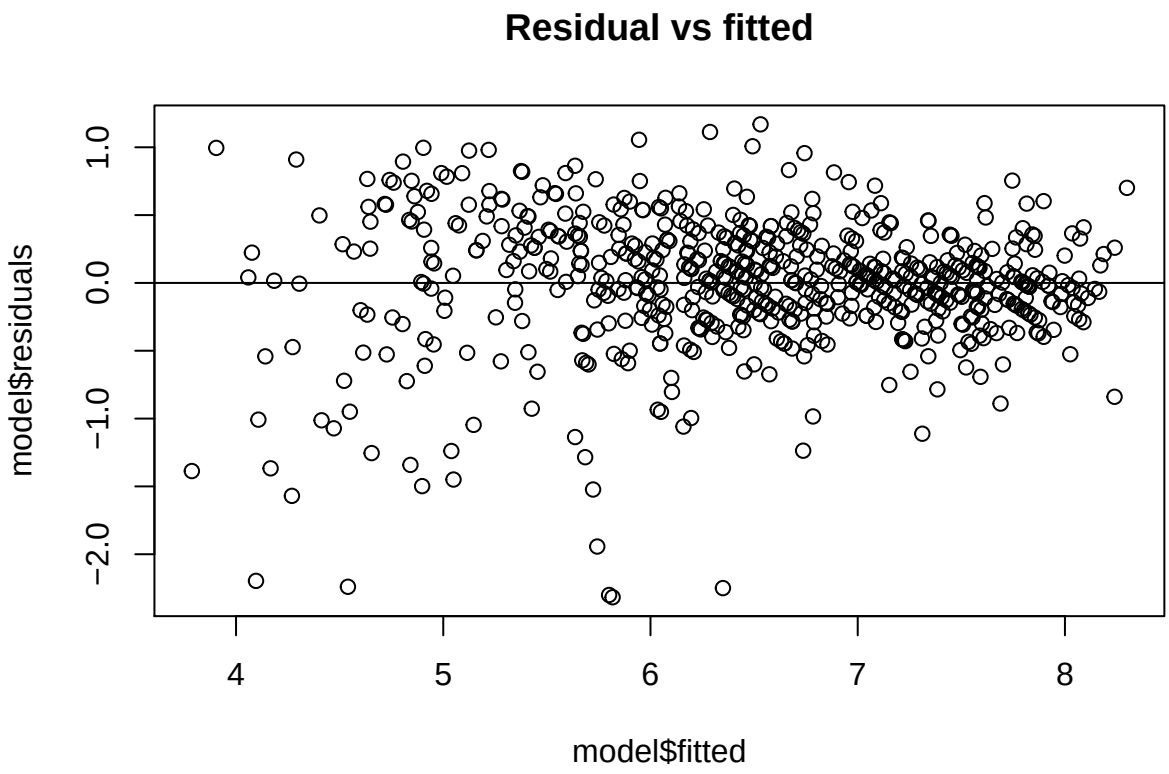
```
model <- lm(imdb_rating~audience_score+critics_score+audience_rating+critics_rating, data = movies)
hist(model$residuals, main = "Normally distrbuted Residuals")
```




```
qqnorm(model$residuals)
qqline(model$residuals)
```

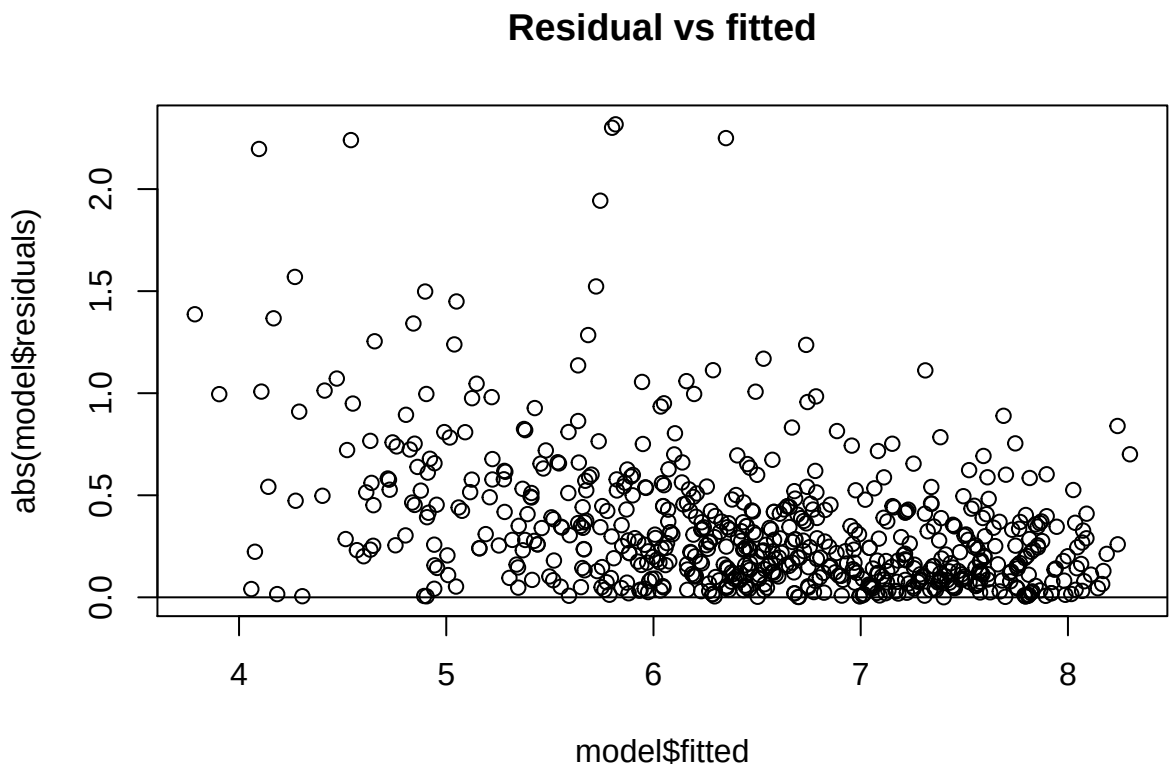



```
plot(model$residuals~model$fitted,main = "Residual vs fitted")+abline(h = 0)
```



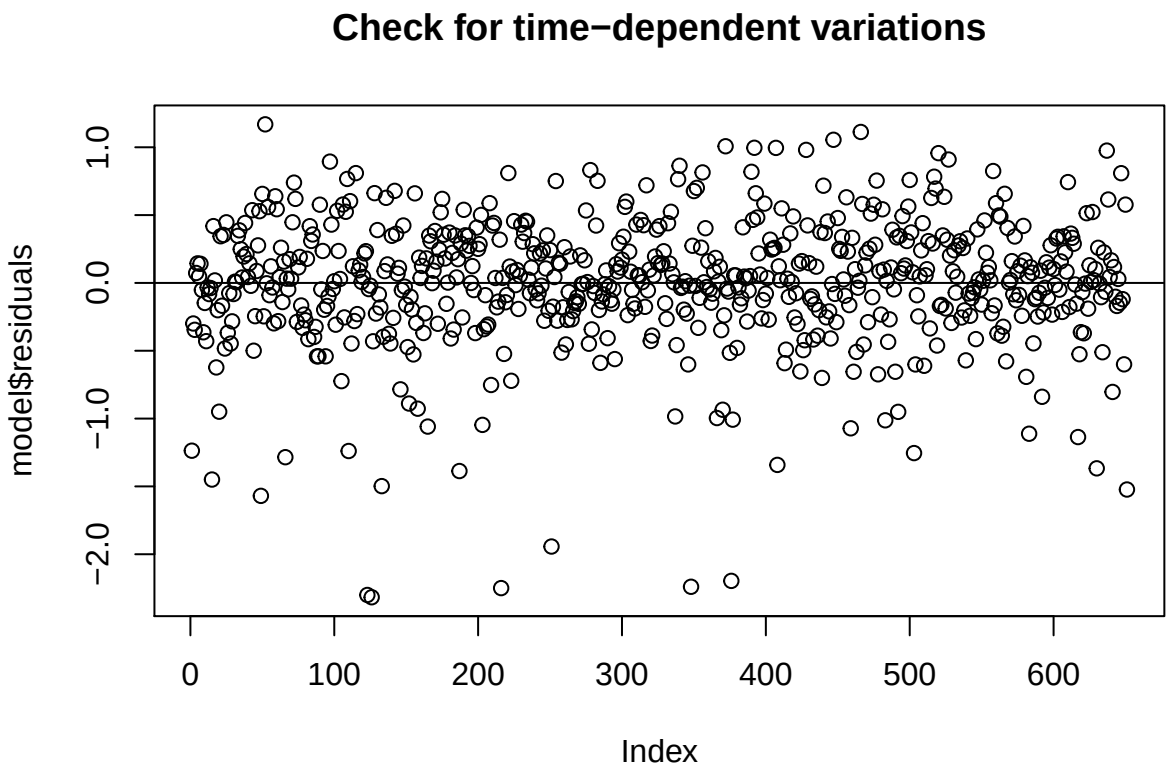

```
## integer(0)
```

```
plot(abs(model$residuals)~model$fitted,main = "Residual vs fitted")+abline(h = 0)
```




```
## integer(0)
```

```
plot(model$residuals,main = "Check for time-dependent variations")+abline(h = 0)
```




```
## integer(0)
```

- From the plots: audience_rating shows: Constant variability. Nearly Normal residuals.

After adding: **critics_rating**

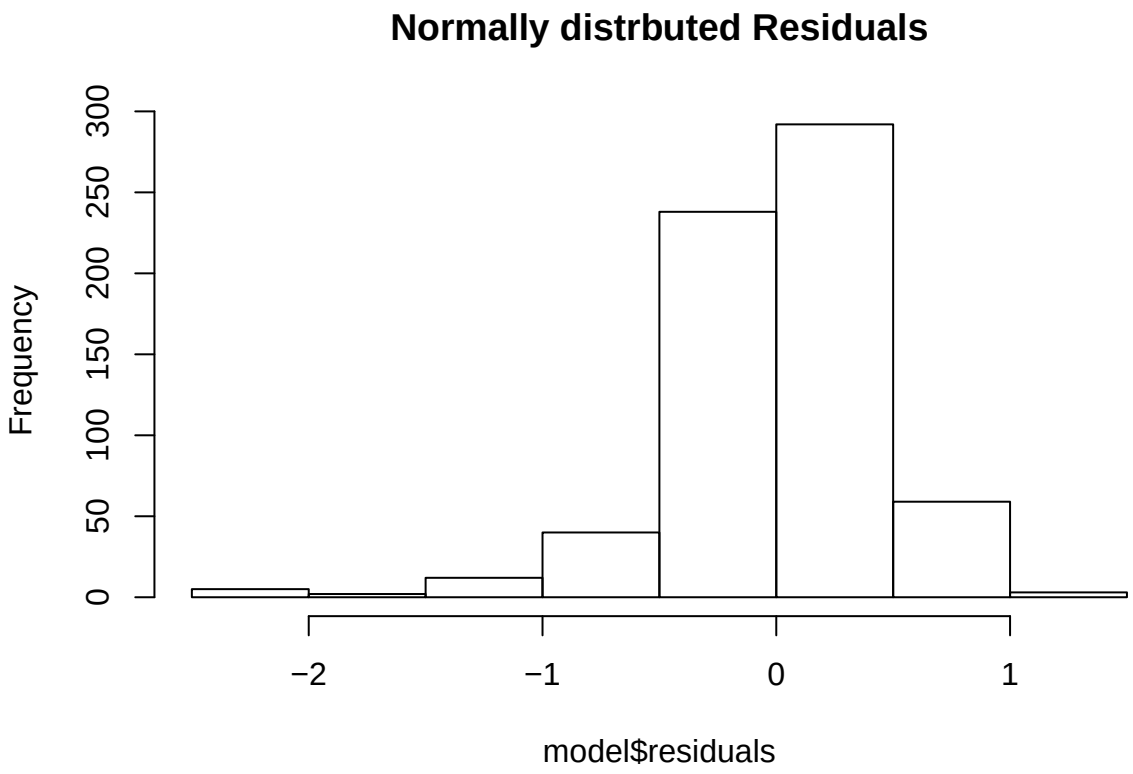
the R-squared: 0.808689

the adjusted R-squared: 0.8072059

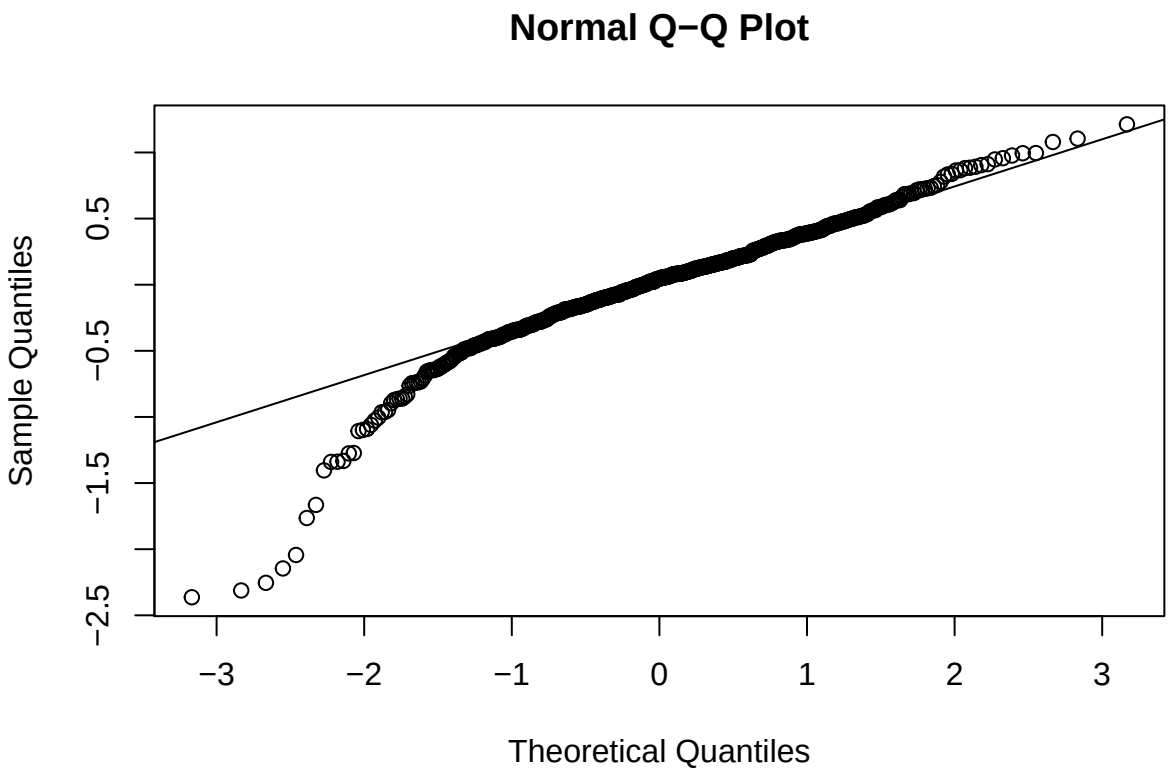
The adjusted- R squared increased so we keep this predictor

genre

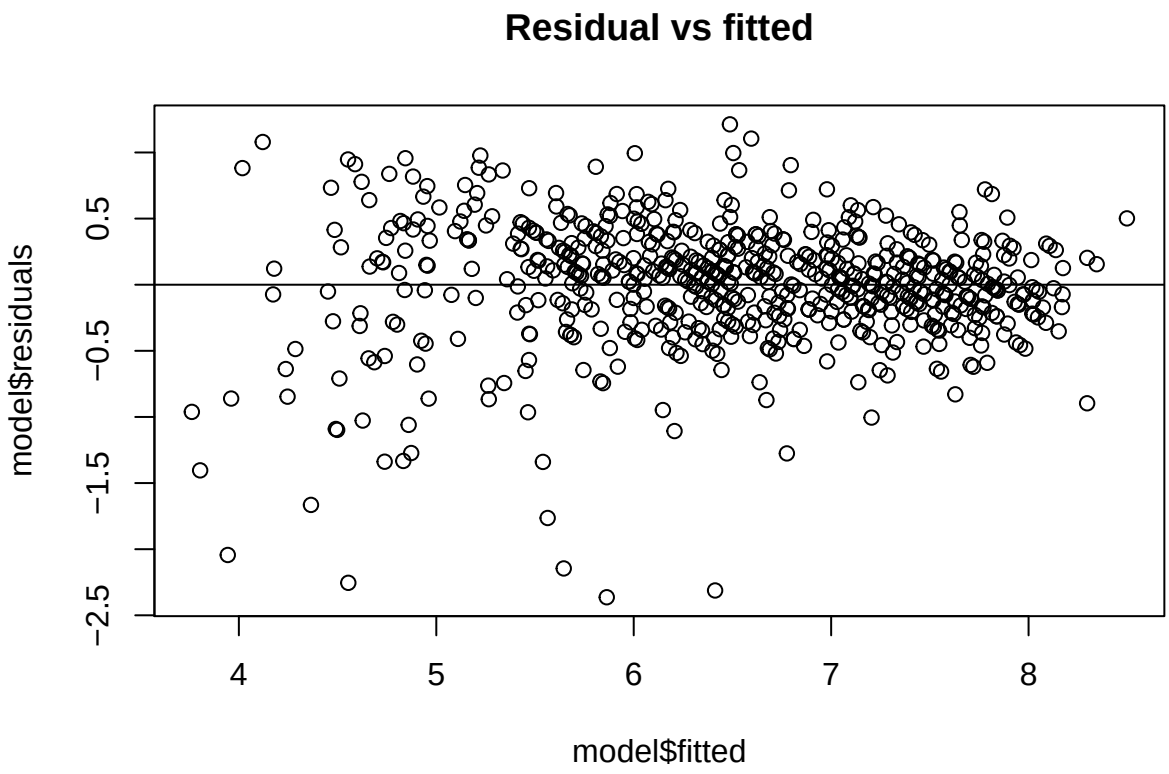
```
model <- lm(imdb_rating~audience_score+critics_score+audience_rating+critics_rating+genre, data = movies
hist(model$residuals, main = "Normally distrbuted Residuals")
```




```
qqnorm(model$residuals)
qqline(model$residuals)
```

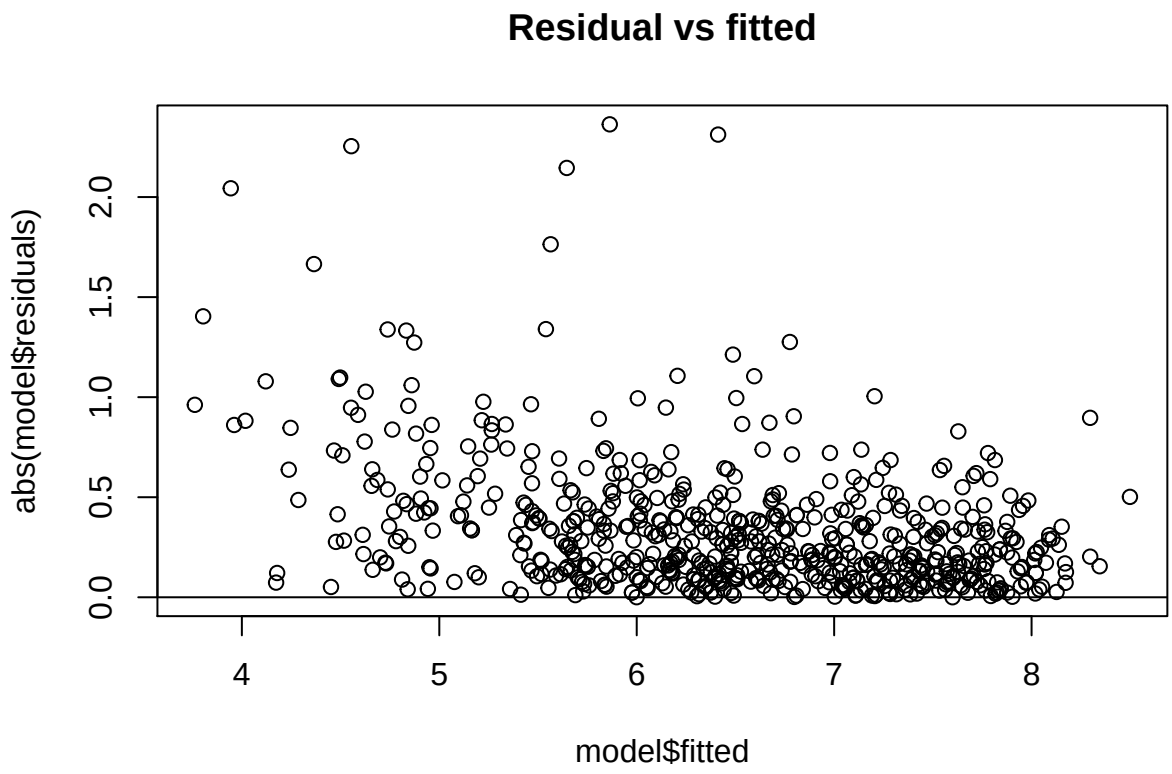



```
plot(model$residuals~model$fitted,main = "Residual vs fitted")+abline(h = 0)
```



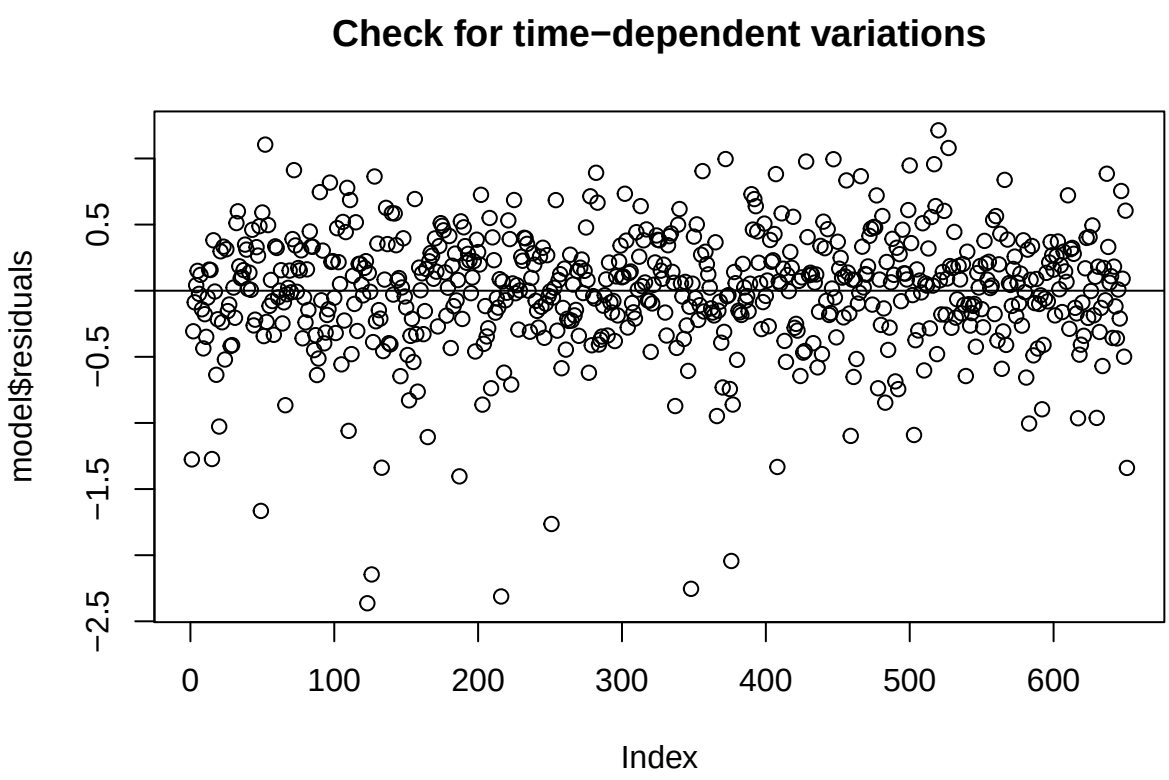

```
## integer(0)
```

```
plot(abs(model$residuals)~model$fitted,main = "Residual vs fitted")+abline(h = 0)
```




```
## integer(0)
```

```
plot(model$residuals,main = "Check for time-dependent variations")+abline(h = 0)
```




```
## integer(0)
```

- From the plots: audience_rating shows: Constant variability. Nearly Normal residuals.

After adding: **genre**

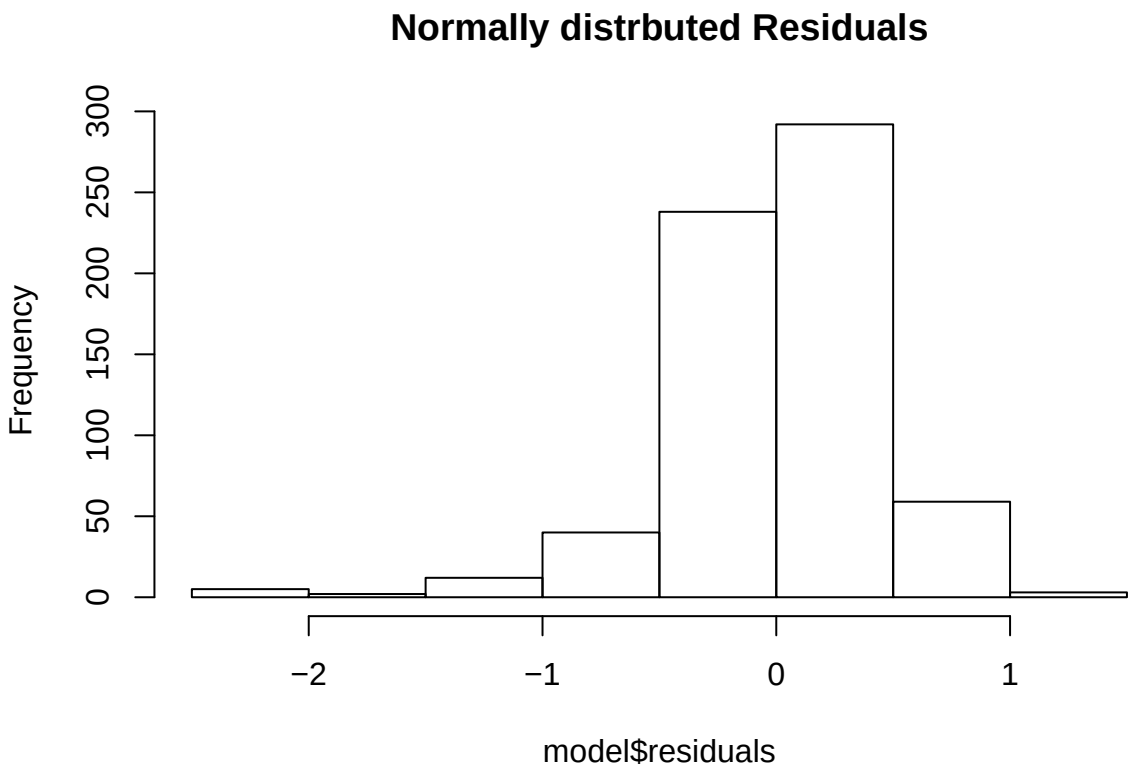
the R-squared: 0.8240338

the adjusted R-squared: 0.8198771

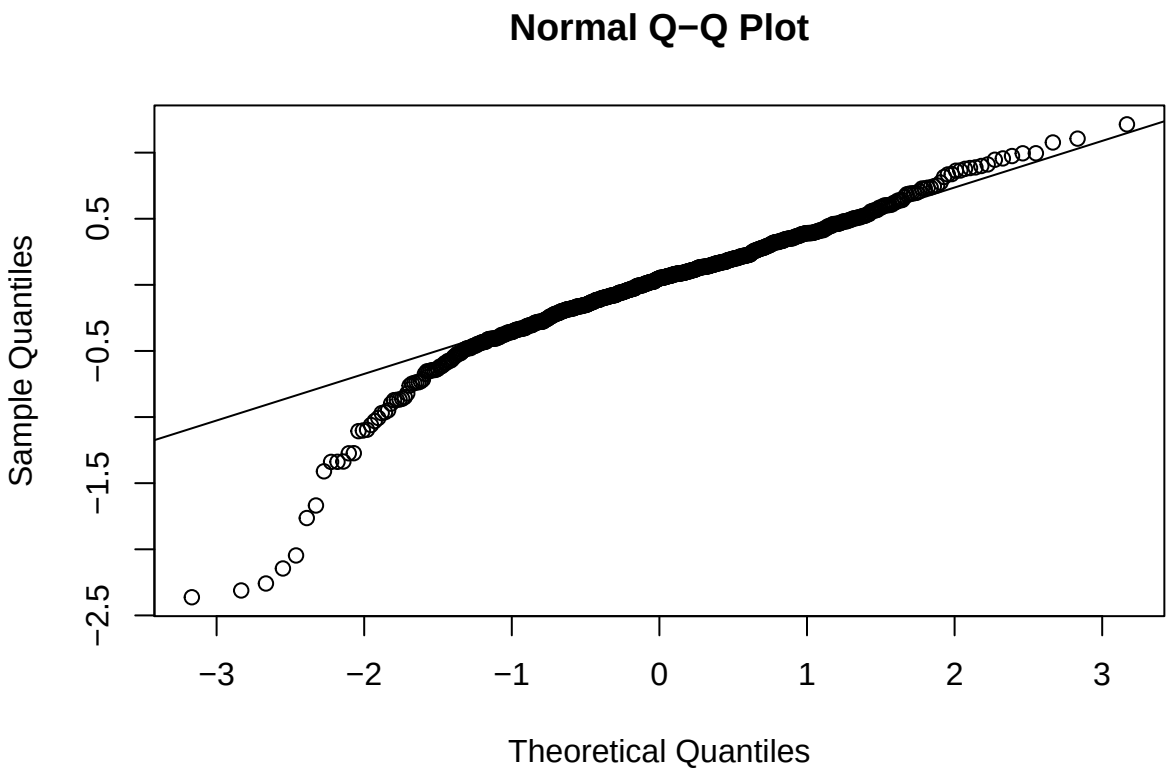
The adjusted- R squared increased so we keep this predictor

best_pic_win

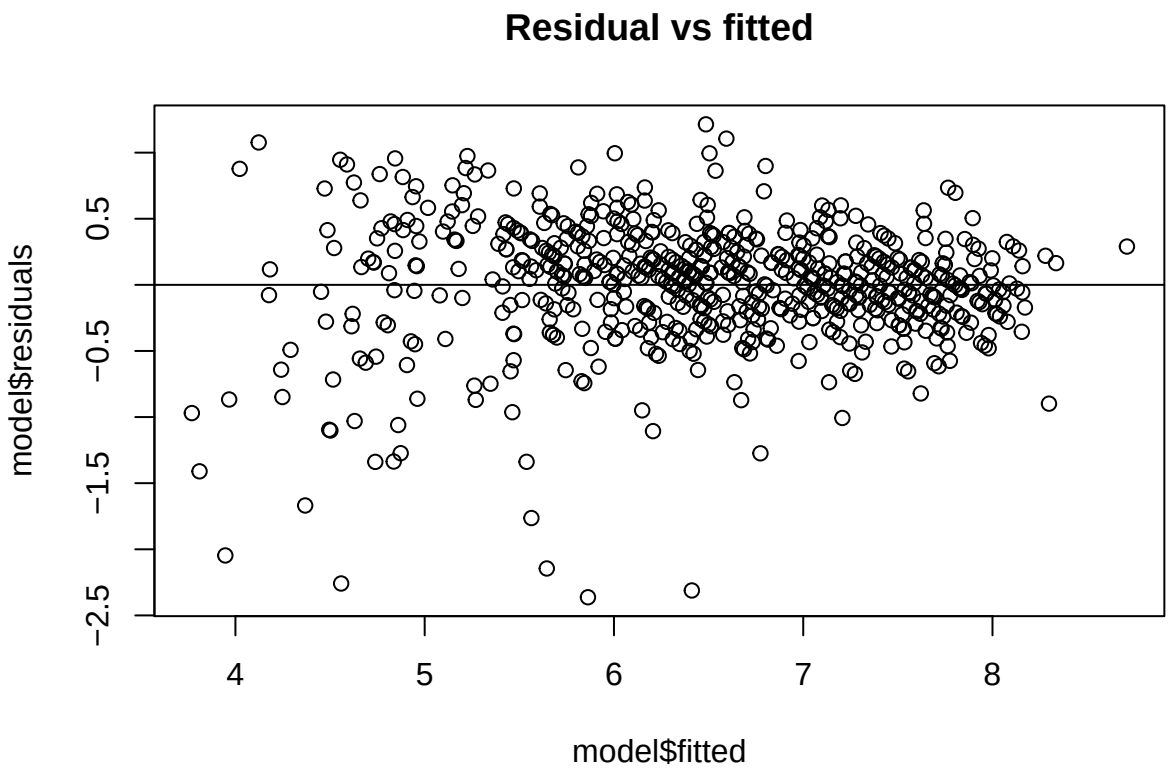
```
model <- lm(imdb_rating~audience_score+critics_score+audience_rating+critics_rating+genre+best_pic_win,
hist(model$residuals, main = "Normally distrbuted Residuals")
```




```
qqnorm(model$residuals)
qqline(model$residuals)
```

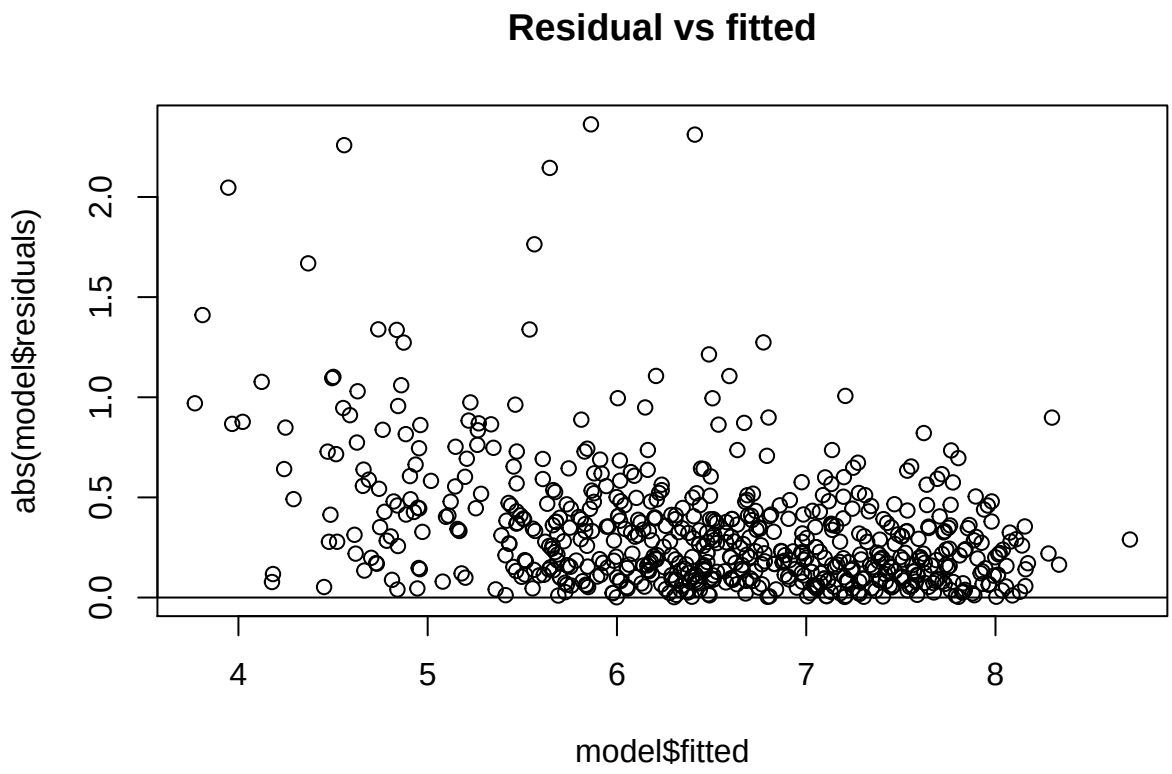



```
plot(model$residuals~model$fitted,main = "Residual vs fitted")+abline(h = 0)
```



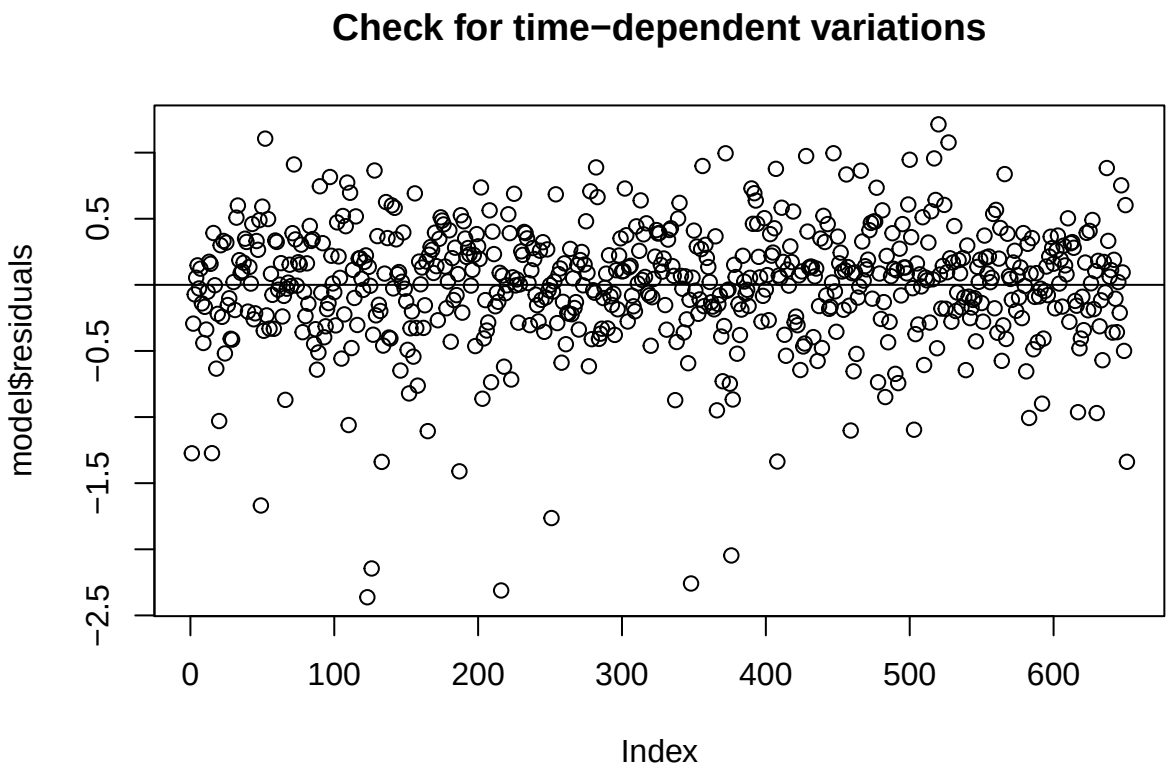

```
## integer(0)
```

```
plot(abs(model$residuals)~model$fitted,main = "Residual vs fitted")+abline(h = 0)
```




```
## integer(0)
```

```
plot(model$residuals,main = "Check for time-dependent variations")+abline(h = 0)
```




```
## integer(0)
```

- From the plots: audience_rating shows: Constant variability. Nearly Normal residuals.

After adding: **best_pic_win**

the R-squared: 0.8244864

the adjusted R-squared: 0.820057

The adjusted- R squared increased so we keep this predictor

- The remaining categorical variables were ignored as they caused a decrease in adjuster - r squared.
- Also studio and actor1 are not really categorical variables, as they do not have categories , but just a whole bunch of actor and studio names.
- There fore in our model they are not used.

Part 5: Prediction

Wonder Woman(2017)

```
#Wonder Woman rating
newprof <- data.frame(audience_score = 87 ,critics_score = 93 ,audience_rating = "Upright" ,critics_rati
p1 <- predict(model, newprof, interval = "prediction", level = 0.95)
kable(p1)
```

fit	lwr	upr
7.705417	6.79029	8.620543

predicted value: 7.7054169

original IMDB rating: 7.4

prediction off by(percentage): 3.0541693

ORIGINAL IMDB VALUE falls inside PREDICTED Confidence interval of 95%

Star Wars: The Last Jedi (2018)

```
#SW The Last Jedi
newprof <- data.frame(audience_score = 43 ,critics_score = 91 ,audience_rating = "Spilled" ,critics_rati
predict(model, newprof, interval = "prediction", level = 0.95)
```

```
##        fit      lwr      upr
## 1 6.191144 5.271704 7.110583
```

```
p1 <- predict(model, newprof, interval = "prediction", level = 0.95)
kable(p1)
```

fit	lwr	upr
6.191144	5.271704	7.110583

predicted value: 6.1911436

original IMDB rating: 7.0

prediction off by(percentage): -8.0885645

ORIGINAL IMDB VALUE falls inside PREDICTED Confidence interval of 95%

Star Wars: The Rise of Skywalker (2019)

```
#SW The rise of skywalker
newprof <- data.frame(audience_score = 86 ,critics_score = 52 ,audience_rating = "Upright" ,critics_rati
predict(model, newprof, interval = "prediction", level = 0.95)
```

```
##        fit      lwr      upr
## 1 7.329901 6.414344 8.245458
```

```
p1 <- predict(model, newprof, interval = "prediction", level = 0.95)
kable(p1)
```

fit	lwr	upr
7.329901	6.414344	8.245458

predicted value: 7.3299008

original IMDB rating: 6.7

prediction off by(percentage): 6.2990085

ORIGINAL IMDB VALUE falls inside PREDICTED Confidence interval of 95%

James Bond 007: Casino Royale 007

```
#James Bond 007: Casino Royale 007
newprof <- data.frame(audience_score = 89 ,critics_score = 95 ,audience_rating = "Upright" ,critics_rati
predict(model, newprof, interval = "prediction", level = 0.95)
```

```
##        fit      lwr      upr
## 1 7.819608 6.904118 8.735097
```

```
p1 <- predict(model, newprof, interval = "prediction", level = 0.95)
kable(p1)
```

fit	lwr	upr
7.819608	6.904119	8.735097

predicted value: 7.8196079

original IMDB rating: 8

prediction off by(percentage): -1.8039209

ORIGINAL IMDB VALUE falls inside PREDICTED Confidence interval of 95%

Part 6: Conclusion

- Our model is **really accurate in the 95% confidence interval** . It is able to predicted IMDB rating values accurately such that they fall into this confidence interval.
- The mean loss of precision of our model is: **6.3480396**
- Our model is **well designed** as it takes into account, diveregence of opinion between Critics and Audiences.‘
- This is a robust well designed model.