Modeling and prediction for movies

Amin Ghaderi

February 15, 2018

Setup

Load packages

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.3
```

```
library(statsr)
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 3.4.3
```

Load data

We use load command to import the movies data.

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

Part 1: Data

Acquisition: This data is randomly selected from IMDB and Rotten Tompato APIs from movies produced before 2016.

Population: To be included in this data set, the movie needs to be (1) in the Rotten Tomatoes and IMDB databases, (2) produced before 2016.

Causality/Generalization: Since the data is randomly sampled from the discussed population and no ***random assignment*** is performed, the resultSs of this study does not demonstrate any causality. Any results could be merely used to demonstrate correlation. The results is also only generalizable to the poplation discussed above, which are movies in IMDB and RT databases, produced before 2016. * * *

Part 2: Research question

In this study, we are choosing our target (dependant variable) as the IMDB score (`imdb_rating`). We would like to see how this score is affected by different factors. We are mostly interested in the effect of `critics_score`, `audience_score`, and movie type (`title_type`). We will also study the effect of rest of the parameters and finally will build a model that can predict the IMDB popularity of the movies based on the available parameters.

Part 3: Exploratory data analysis

3.1 EDA on `critics_score` vs `imdb_rating`:

This is the first pair that we are studying. Let's first see the data type of these features to decide the required statistics.

3.1.1 Scatter Plot and Fitting Linear Model

```
lapply(movies, class)$critics_score
```

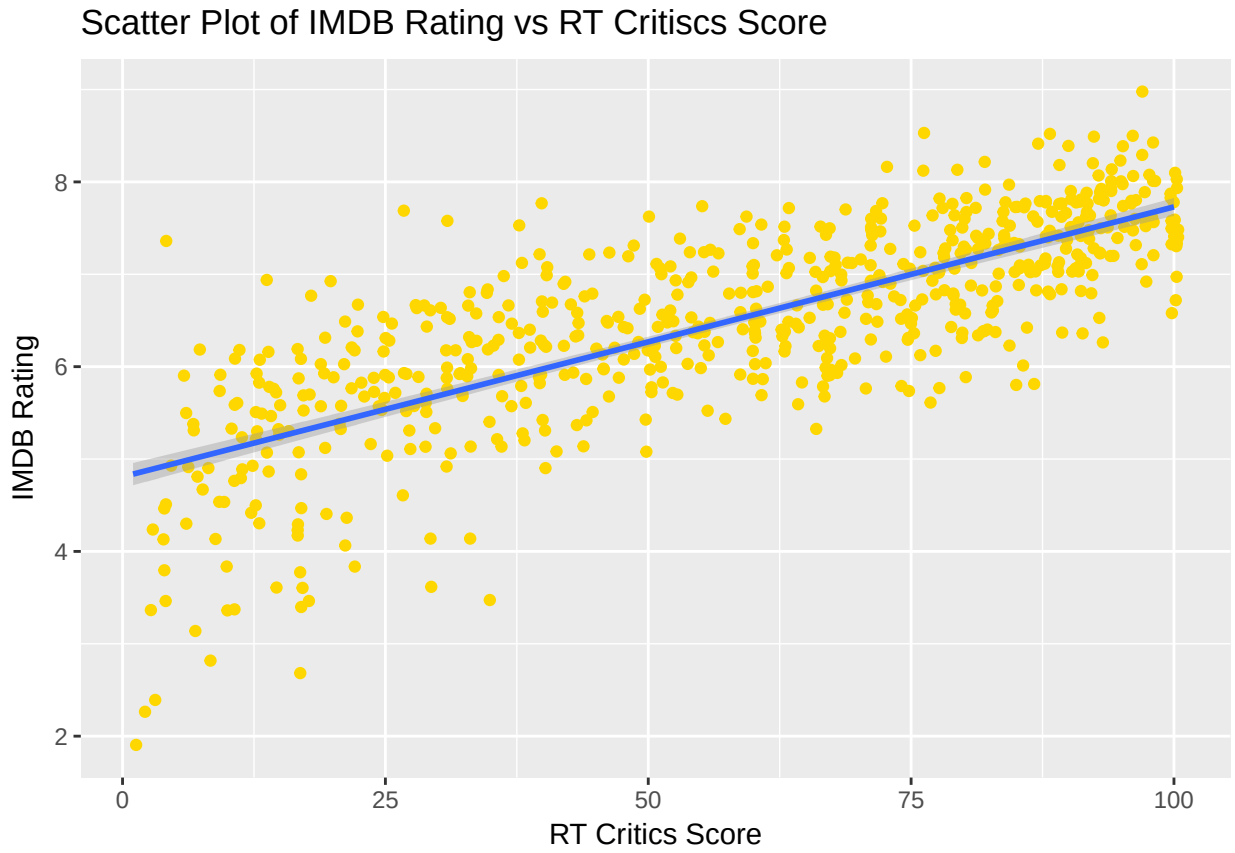
```
## [1] "numeric"
```

```
lapply(movies, class)$imdb_rating
```

```
## [1] "numeric"
```

Both variables are numeric. Therefore, we go with a scatter plot to first get an understanding of the data. Explanatory variable (`critics_score`) will be on the x-axis and the target (`imdb_rating`) on the y-axis.

```
ggplot(movies, aes(x=critics_score, y=imdb_rating))+geom_jitter(color='gold')+
  labs(x="RT Critics Score", y="IMDB Rating",
       title = 'Scatter Plot of IMDB Rating vs RT Critiscs Score')+
  stat_smooth(method = "lm", se = TRUE)
```



There is a very obvious linear relationship between these two values. The relationship is positive, linear, and strong. Let's quantify this relationship. we are going to fit a linear model. Before that however, we need to see the correlation between these two variables.

```
movies%>%summarise(cor(critics_score, imdb_rating))
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.3
```

```
## # A tibble: 1 x 1
##   `cor(critics_score, imdb_rating)`
##                               <dbl>
## 1                             0.765
```

The correlaton also is quite large. Therefore, we can conclude that the relationship between these two values is quite strong.

Let's fit the linear model and see the R-squared score.

```
imdb_critics <- lm (imdb_rating~critics_score, data=movies)
summary(imdb_critics)
```

```
##
## Call:
## lm(formula = imdb_rating ~ critics_score, data = movies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.93679 -0.39499  0.04512  0.43875  2.47556
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.8075715  0.0620690   77.45   <2e-16 ***
## critics_score 0.0292177  0.0009654   30.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6991 on 649 degrees of freedom
## Multiple R-squared:  0.5853, Adjusted R-squared:  0.5846
## F-statistic: 915.9 on 1 and 649 DF,  p-value: < 2.2e-16
```

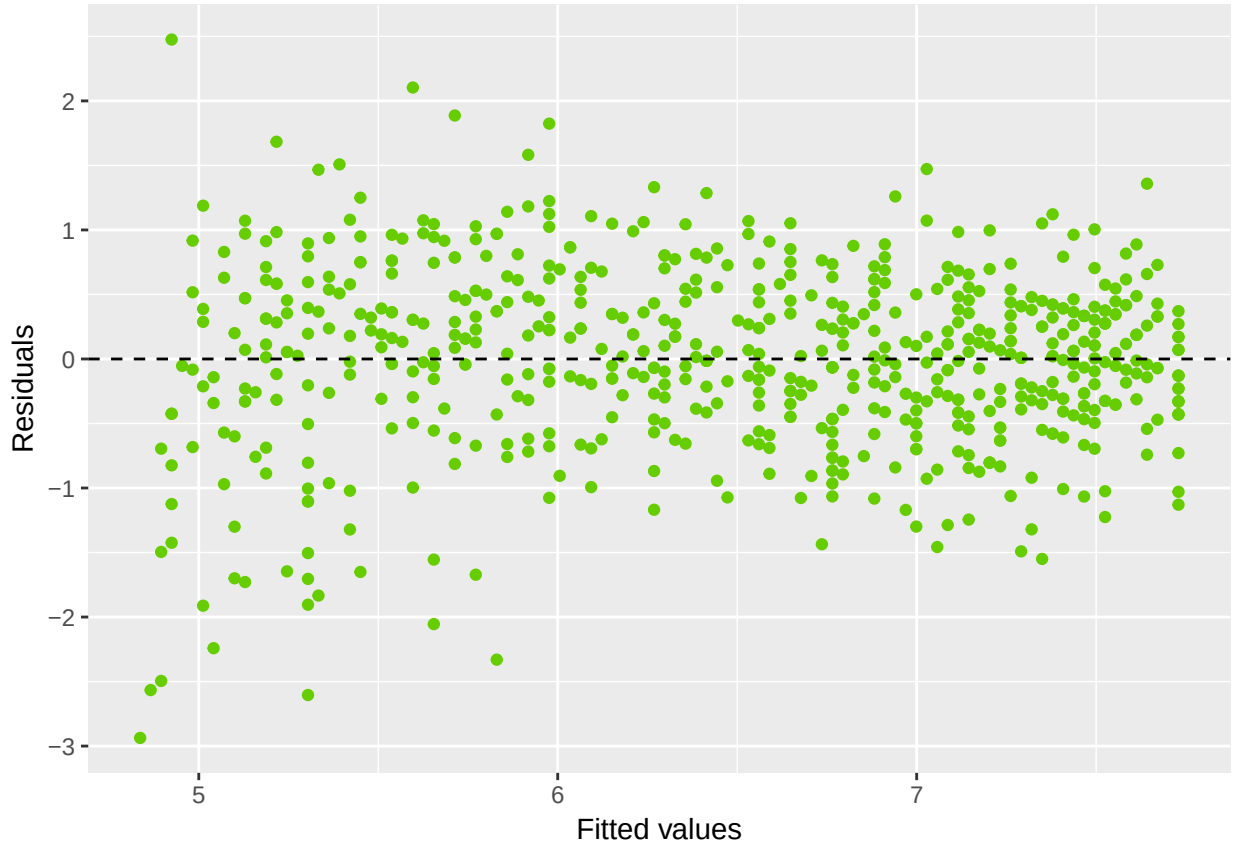
p-value of the relationship is very small and the adjusted R-squred demonstrate that 58.5% of the variability of the data can be explained by this feature, which is very impressive. However, we need to also assure that this linear regression model is credible.

3.1.2 Model Diagnostics

We want to assure that this linear model is relaiable. We need to chec for (1) linearity, (2) nearly normal distribution, and (3) constant variability.

Linearity: We already checked this using a scatter plot. We can also verify this using a plot of the residuals vs. fitted (predicted) values.

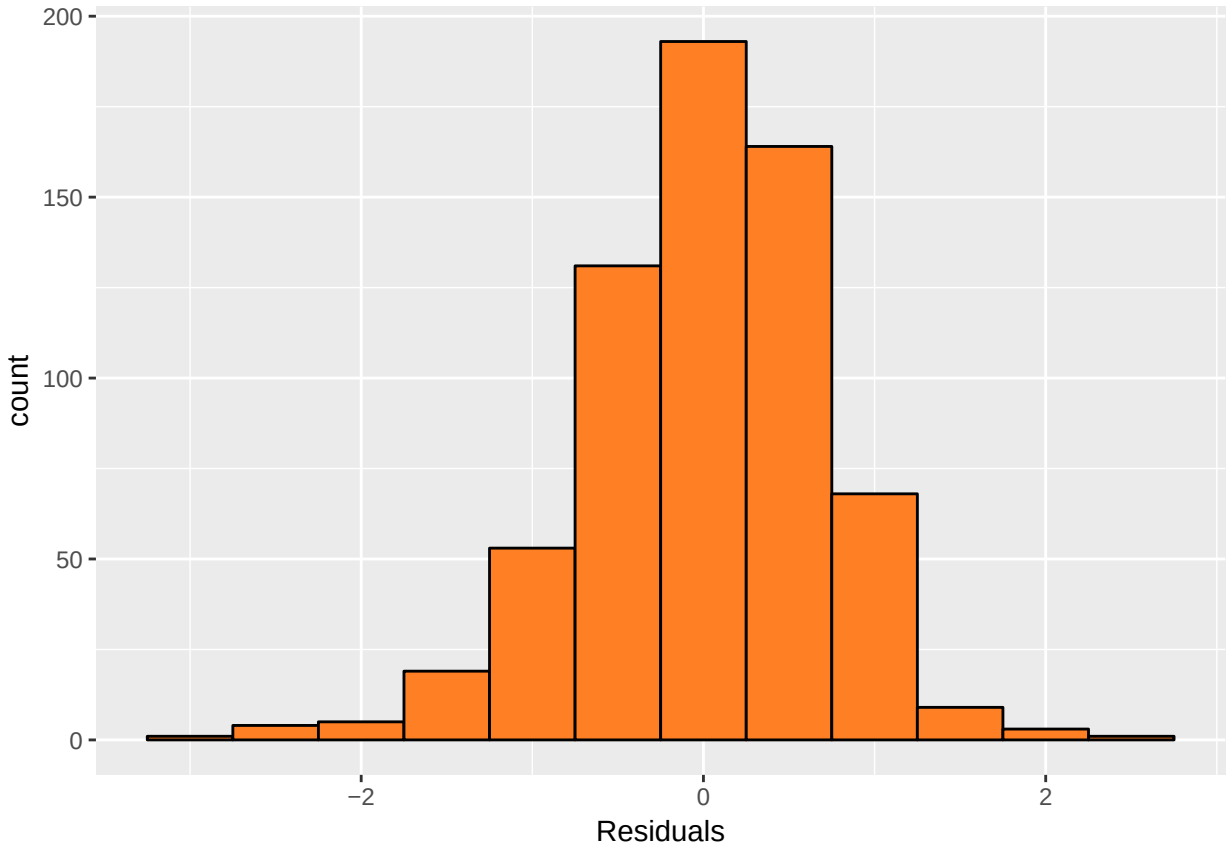
```
ggplot(data = imdb_critics, aes(x = .fitted, y = .resid)) +
  geom_point(color = 'chartreuse3') +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Fitted values") +
  ylab("Residuals")
```



The plot seems to be randomly distributed. However, there seems to be a bit more data in higher rating s and more scatter around lower ratings.

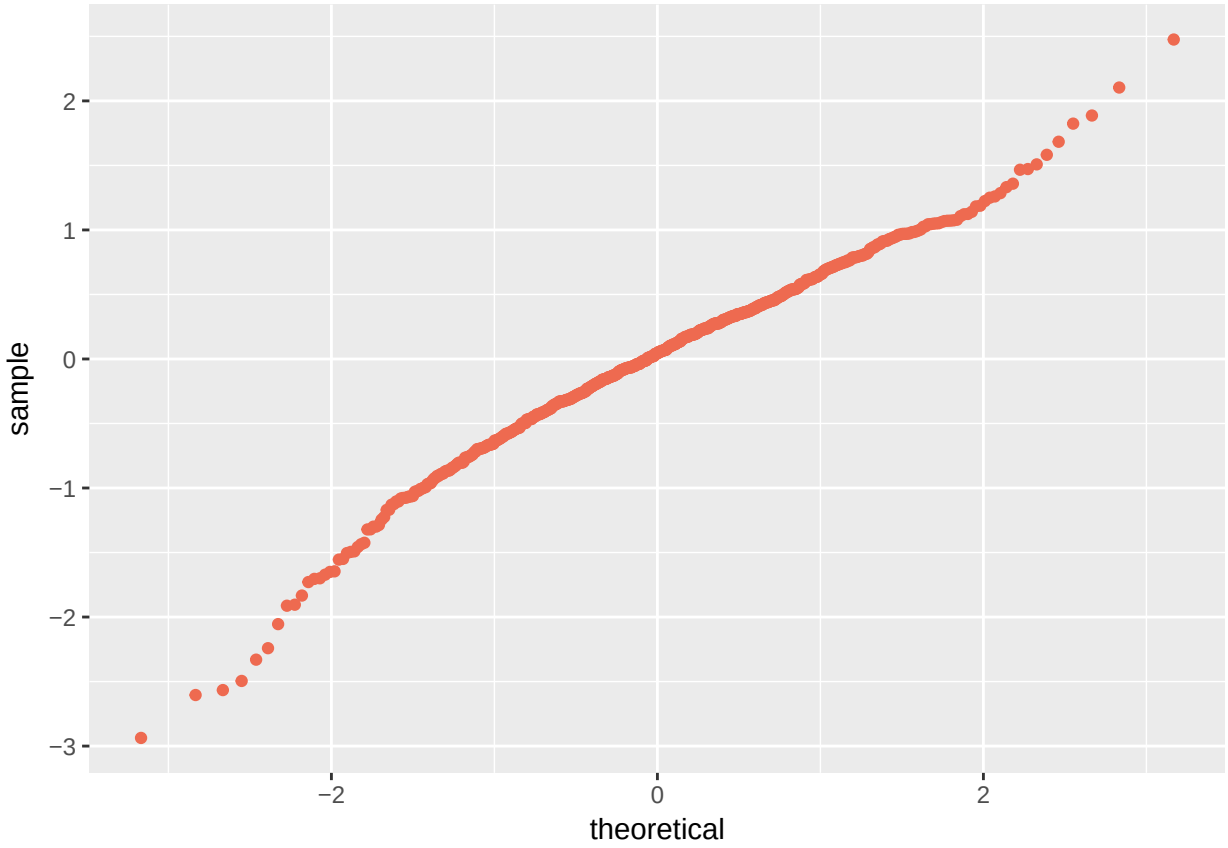
Nearly Normal Residuals: To check this condition, we will check the histograms.

```
ggplot(data = imdb_critics, aes(x = .resid)) +
  geom_histogram(binwidth = .5, fill = 'chocolate1', color = 'black' ) +
  xlab("Residuals")
```



There seems to be a symettry around 0 and data could be roughly considered normal. We can also use the normal probablity plot of residuals.

```
ggplot(data = imdb_critics, aes(sample = .resid)) +
  stat_qq(color = 'coral2')
```



This relationship also seems to be linear which assures us that the residuals are distributed normally. Therefore, considering that the model is credible, we can say that the critics score can be a very good predictor of the movie popularity (IMDB rating).

3.2 EDA on `audience_score` vs `imdb_rating`:

This is the second pair that we are studying. Let's first see the data type of these features to decide the required statistics.

3.2.1 Scatter Plot and Fitting Linear Model

```
lapply(movies, class)$audience_score
```

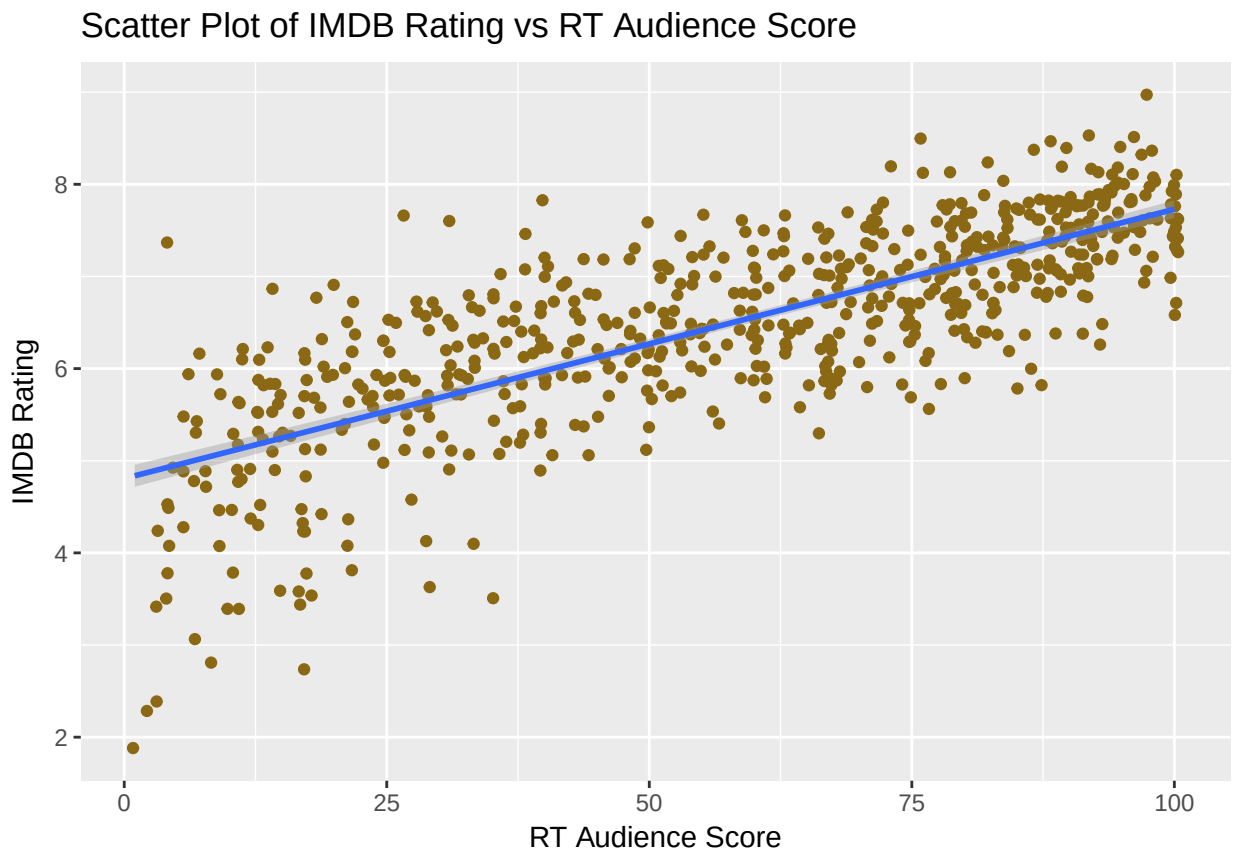
```
## [1] "numeric"
```

```
lapply(movies, class)$imdb_rating
```

```
## [1] "numeric"
```

Both variables are numeric. Therefore, we go with a scatter plot to first get an understanding of the data. Explanatory variable (`audience_score`) will be on the x-axis and the target (`imdb_rating`) on the y-axis.

```
ggplot(movies, aes(x=critics_score, y=imdb_rating))+geom_jitter(color='goldenrod4')+
  labs(x="RT Audience Score", y="IMDB Rating",
       title = 'Scatter Plot of IMDB Rating vs RT Audience Score')+
  stat_smooth(method = "lm", se = TRUE)
```



There is a very obvious linear relationship between these two values. It also makes sense since they are both audience score, and it is interesting to know that the audience of both website tend to have similar attitude toward movies. The relationship is positive, linear, and strong. Let's quantify this relationship. we are going to fit a linear model. Before that however, we need to see the correlation between these two variables.

```
movies%>%summarise(cor(audience_score, imdb_rating))
```

```
## # A tibble: 1 x 1
##   `cor(audience_score, imdb_rating)`
##                                <dbl>
## 1                              0.865
```

The correlaton also is quite large, even larger than the critics score. Therefore, we can conclude that the relationship between these two values is quite strong.

Let's fit the linear model and see the R-squared score.

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

```
##
## Call:
## lm(formula = imdb_rating ~ audience_score, data = movies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2082 -0.1866  0.0712  0.3093  1.1516
##
## Coefficients:
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.599992   0.069291   51.95   <2e-16 ***
## audience_score 0.046392   0.001057   43.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.545 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
```

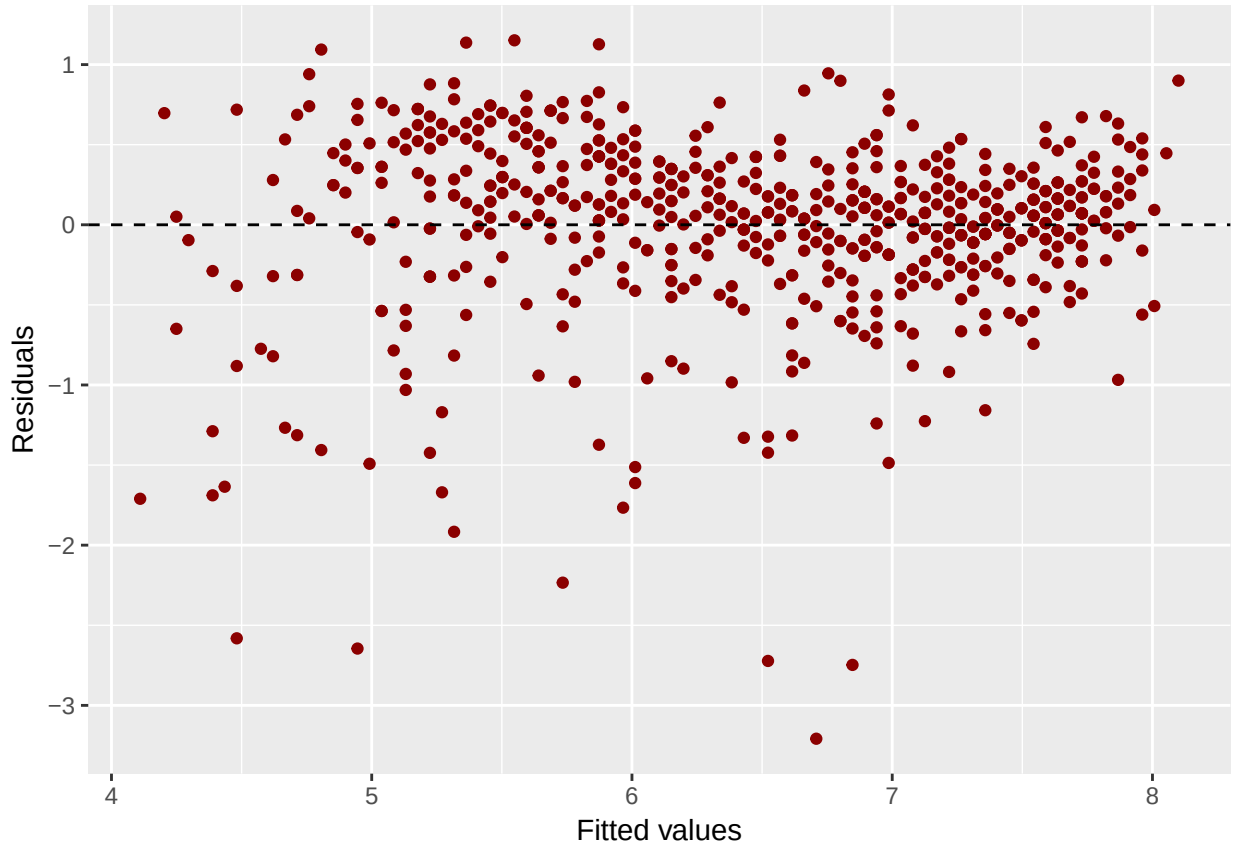
p-value of the relationship is very small and the adjusted R-squred demonstrate that 74.8% of the variability of the data can be explained by this feature, which is very impressive. However, we need to also assure that this linear regression model is credible.

3.2.2 Model Diagnostics

We want to assure that this linear model is relaiable. We need to chec for (1) linearity, (2) nearly normal distribution, and (3) constant variability.

Linearity: We already checked this using a scatter plot. We can also verify this using a plot of the residuals vs. fitted (predicted) values.

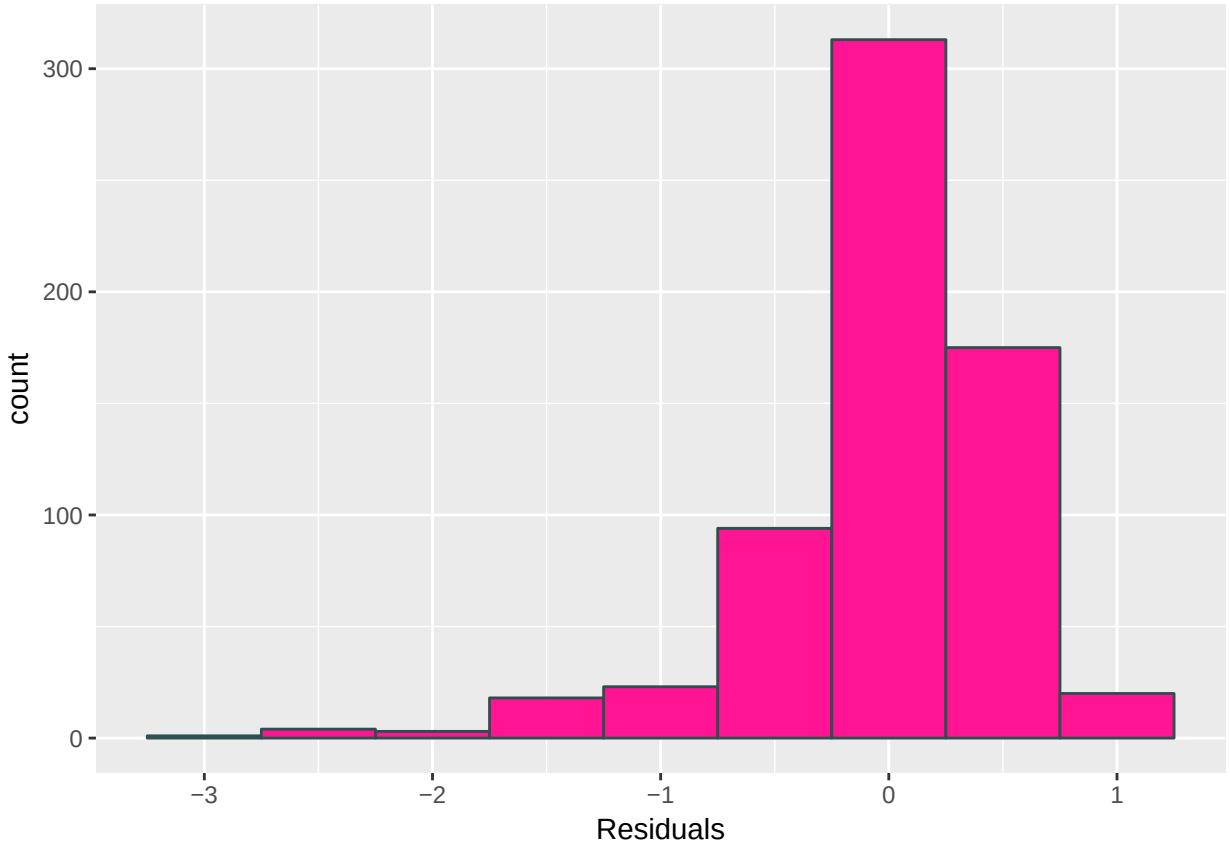
```
ggplot(data = imdb_RT, aes(x = .fitted, y = .resid)) +
  geom_point(color = 'darkred') +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Fitted values") +
  ylab("Residuals")
```



The plot seems to be randomly distributed. However, there seems to be a bit more data in higher rating s and more scatter around lower ratings.

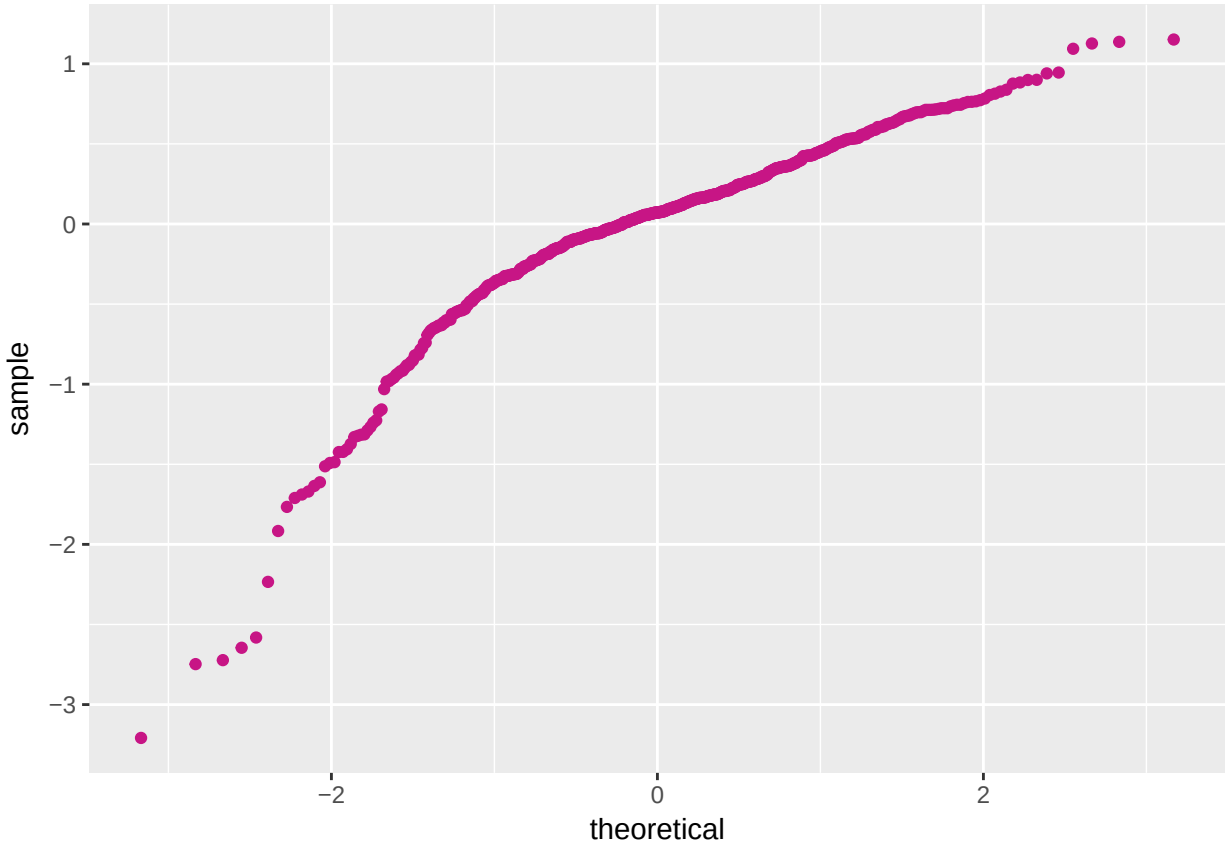
Nearly Normal Residuals: To check this condition, we will check the histograms.

```
ggplot(data = imdb_RT, aes(x = .resid)) +
  geom_histogram(binwidth = .5, fill = 'deeppink', color = 'darkslategrey' ) +
  xlab("Residuals")
```



There seems to be a symmetry around 0 and data could be hardly be considered normal. We can also use the normal probablity plot of residuals to see how much of deviation we have around the normal estimate.

```
ggplot(data = imdb_RT, aes(sample = .resid)) +
  stat_qq(color = 'mediumvioletred')
```



This relationship also seems to be not completely linear which assures us that the residuals are not necesaarily distributed normally. Therefore, considering that the model is credible. However, we can assume that the asumption holds to perform our modelling.

3.3 EDA on `title_type` vs `imdb_rating`:

This is the third pair that we are studying. Let's first see the data type of these features to decide the required statistics.

3.3.1 Scatter Plot and Fitting Linear Model

```
lapply(movies, class)$top200_box
```

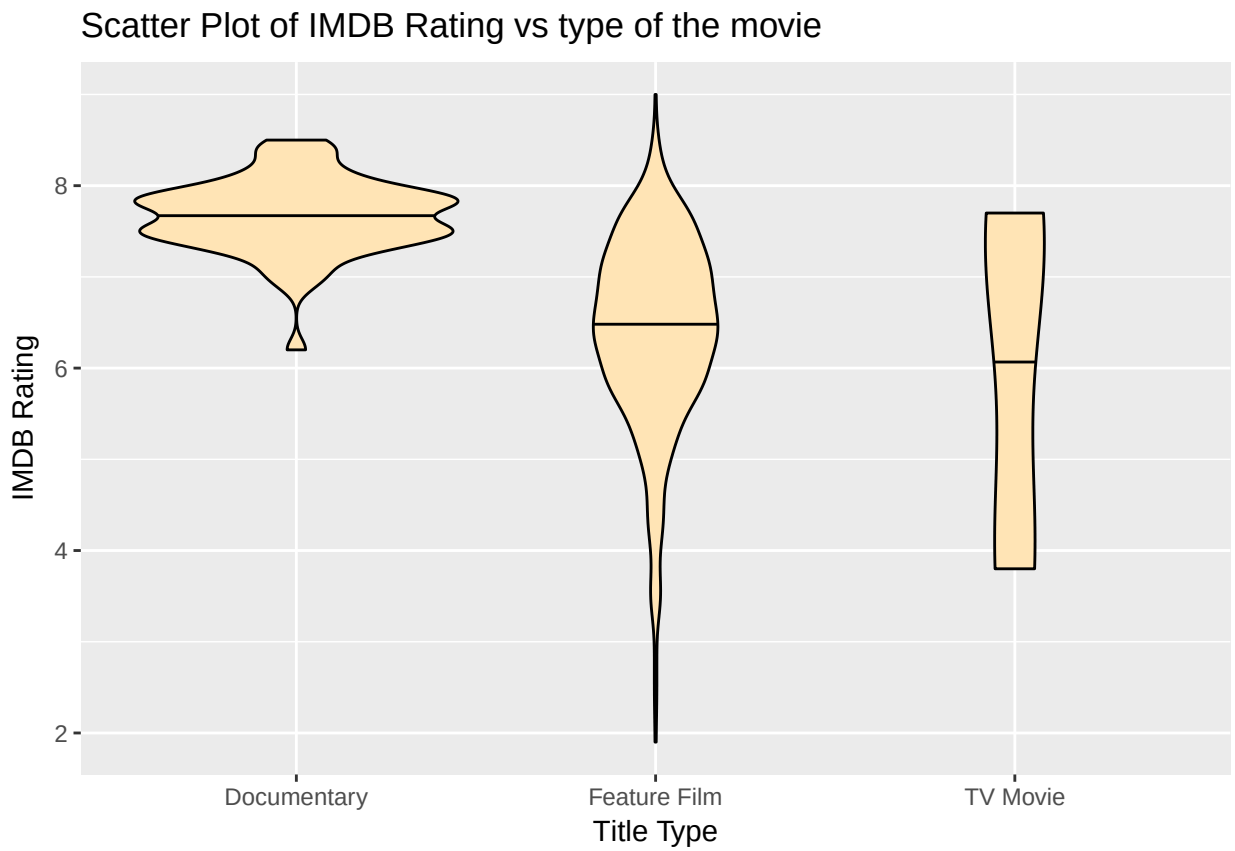
```
## [1] "factor"
```

```
lapply(movies, class)$imdb_rating
```

```
## [1] "numeric"
```

One variable is numeric and the other is categorical. Therefore, we go with a violin plot to first get an understanding of the data. Explanatory variable (`title_type`) will be on the x-axis and the target (`imdb_rating`) on the y-axis.

```
ggplot(movies, aes(x=title_type, y=imdb_rating))+geom_violin(scale='area',color='black', fill='moccasin
  labs(x="Title Type", y="IMDB Rating",
       title = 'Scatter Plot of IMDB Rating vs type of the movie')
```



There is a difference between the shape and median of different categories. We are going to fit a linear model.

Let's fit the linear model and see the R-squared score.

```
imdb_type <- lm (imdb_rating~title_type, data=movies)
summary(imdb_type)
```

```
##
## Call:
## lm(formula = imdb_rating ~ title_type, data = movies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4875 -0.4875  0.1125  0.7125  2.6125
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)              7.6691     0.1382  55.474  < 2e-16 ***
## title_typeFeature Film  -1.2816     0.1445  -8.867  < 2e-16 ***
## title_typeTV Movie      -1.6291     0.4789  -3.402 0.000711 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.025 on 648 degrees of freedom
## Multiple R-squared:  0.1094, Adjusted R-squared:  0.1067
## F-statistic:  39.8 on 2 and 648 DF,  p-value: < 2.2e-16
```

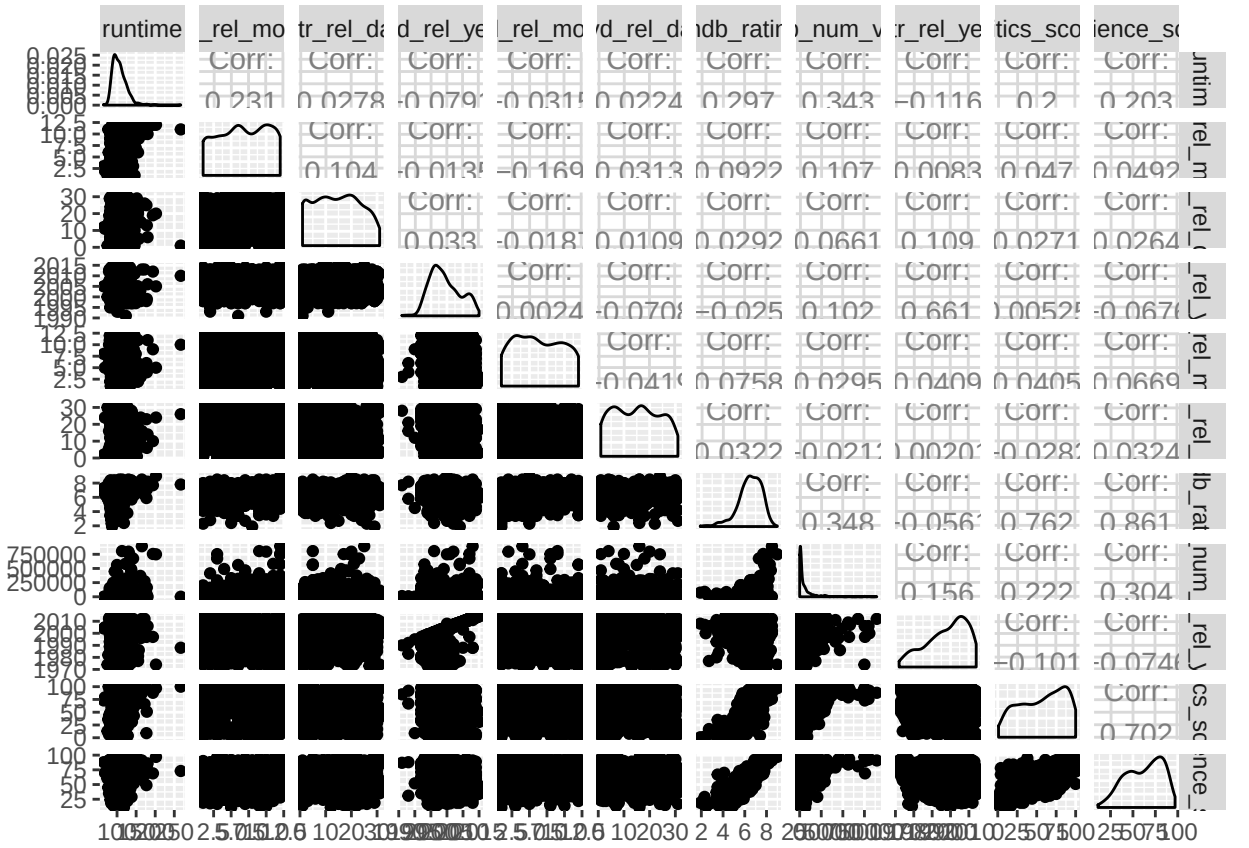
p-value of the relationship is very small and the adjusted R-squred demonstrate that on;y 10.9% of the

variability of the data can be explained by this feature, which is not very impressive. Although only around 11% of the variability is explained with this feature, we can add it to our model becuase it might be covering some information that is not covered with other features.

Part 4: Modeling

Before looking at all features, let's take a look at all the reationships, to assure we are not including redundunt (colinear) features.

```
ggpairs(na.omit(movies), columns =c(4,8,9,10,11,12,13,14,7,16,18))
```



There doesn't seem to be a particular correlation between different numerical features. Therefore, we can say that no feature is **co-linear** with other features. Looking at the ggpairs we can see that only three of the features are highly correlated, `audience_rating`,`critics_rating`,`imdb_rating`.

`imdb_rating` is our target and its correlation with other components is not important. However, we have seen that `audience_rating` and `critics_rating` are correlated. We might have to drop one of them that do not contribute to the model.

4.1 Backward Elimination Ensemble of the Model

Based on the model selection lecture from Duke University, "We start with a **full model** (containing all co-variates), drop one predictor at a time untill the ***parsimonious*** model is reached." There are many criteria for model selection, main criteria are p-value and adjusted-R^2. We work with p-value criterion. We will build the full model and drop the variable with the highet p-value and refit a smaller model. We repeat

untill all variables left in the model are significant (tiny p-value). Although due to arbitrary p-value, our model might not be most accurate, due to the number of features, using p-value will be simpler to implement.

```
clean_movies <- na.omit(movies)
full_model <- lm(imdb_rating ~ title_type + genre + runtime + mpaa_rating +
                   thtr_rel_year + thtr_rel_month + thtr_rel_day +
                   dvd_rel_year + dvd_rel_month + dvd_rel_day + imdb_num_votes
                 + critics_rating + critics_score + audience_rating + audience_score +
                   best_pic_nom + best_pic_win + best_actor_win + best_actress_win +
                   best_dir_win +    top200_box, data = clean_movies)
#summary(full_model)
summary(full_model)$adj.r.squared
```

```
## [1] 0.8257023
```

We are not going to show the results for each model. You can uncomment the last line of the code, to see the summary. We chose to drop `best_pic_win` that has the largest p-value 0.89393.

```
step1_model <- lm(imdb_rating ~ title_type + genre + runtime + mpaa_rating + thtr_rel_year +
 thtr_rel_month + thtr_rel_day + dvd_rel_year + dvd_rel_month + dvd_rel_day +
 imdb_num_votes + critics_rating + critics_score + audience_rating + audience_score +
 best_pic_nom + best_actor_win + best_actress_win + best_dir_win + top200_box,
 data = clean_movies)
#summary(step1_model)
summary(step1_model)$adj.r.squared
```

```
## [1] 0.8259959
```

We chose to drop `dvd_rel_year` that has the largest p-value 0.853523.

```
step2_model <- lm(imdb_rating ~ title_type + genre + runtime + mpaa_rating + thtr_rel_year +
thtr_rel_month + thtr_rel_day + dvd_rel_month + dvd_rel_day + imdb_num_votes + critics_rating +
critics_score + audience_rating + audience_score +  best_pic_nom + best_actor_win +
  best_actress_win + best_dir_win + top200_box, data = clean_movies)
#summary(step2_model)
summary(step2_model)$adj.r.squared
```

```
## [1] 0.8262837
```

We chose to drop `best_dir_win` that has the largest p-value 0.695969.

```
step3_model <- lm(imdb_rating ~ title_type + genre + runtime + mpaa_rating +
                    thtr_rel_year +thtr_rel_month + thtr_rel_day +
                    dvd_rel_month + dvd_rel_day + imdb_num_votes + critics_rating +
                    critics_score + audience_rating + audience_score +
                    best_pic_nom + best_actor_win +   best_actress_win + top200_box,
                  data = clean_movies)
#summary(step3_model)
summary(step3_model)$adj.r.squared
```

```
## [1] 0.8265353
```

We chose to drop `best_actor_win` that has the largest p-value 0.630932. Here you can see the backward elimination process:

```
step4_model <- lm(imdb_rating ~ title_type + genre + runtime + mpaa_rating +
                    thtr_rel_year +thtr_rel_month + thtr_rel_day +
                    dvd_rel_month + dvd_rel_day + imdb_num_votes + critics_rating +
                    critics_score + audience_rating + audience_score +
```

```
                    best_pic_nom + best_actress_win + top200_box,
                  data = clean_movies)
#summary(step4_model)
summary(step4_model)$adj.r.squared
```

```
## [1] 0.8267629
```

We chose to drop `thtr_rel_day` that has the largest p-value 0.538427.

```
step5_model <- lm(imdb_rating ~ title_type + genre + runtime + mpaa_rating +
                    thtr_rel_year +thtr_rel_month + dvd_rel_month + dvd_rel_day +
                    imdb_num_votes + critics_rating +critics_score + audience_rating +
                    audience_score +best_pic_nom + best_actress_win + top200_box,
                  data = clean_movies)
#summary(step5_model)
summary(step5_model)$adj.r.squared
```

```
## [1] 0.8269462
```

We chose to drop `thtr_rel_year` that has the largest p-value 0.545489.

```
step6_model <- lm(imdb_rating ~ title_type + genre + runtime + mpaa_rating +
                    +thtr_rel_month + dvd_rel_month + dvd_rel_day +
                    imdb_num_votes + critics_rating +critics_score + audience_rating +
                    audience_score +best_pic_nom + best_actress_win + top200_box,
                  data = clean_movies)
#summary(step6_model)
summary(step6_model)$adj.r.squared
```

```
## [1] 0.8271328
```

We chose to drop `title_type` that has the largest p-value 0.373162.

```
step7_model <- lm(imdb_rating ~ genre + runtime + mpaa_rating +
                    +thtr_rel_month + dvd_rel_month + dvd_rel_day +
                    imdb_num_votes + critics_rating +critics_score + audience_rating +
                    audience_score +best_pic_nom + best_actress_win + top200_box,
                  data = clean_movies)
#summary(step7_model)
summary(step7_model)$adj.r.squared
```

```
## [1] 0.8274824
```

We chose to drop `top200_boxyes` that has the largest p-value 0.338563.

```
step8_model <- lm(imdb_rating ~ genre + runtime + mpaa_rating +
                    +thtr_rel_month + dvd_rel_month + dvd_rel_day +
                    imdb_num_votes + critics_rating +critics_score + audience_rating +
                    audience_score +best_pic_nom + best_actress_win , data = clean_movies)
#summary(step8_model)
summary(step8_model)$adj.r.squared
```

```
## [1] 0.8275065
```

We chose to drop `mpaa_rating` that its most significant category has the largest p-value 0.386720.

```
step9_model <- lm(imdb_rating ~ genre + runtime + thtr_rel_month + dvd_rel_month +
                    dvd_rel_day + imdb_num_votes + critics_rating +critics_score +
                    audience_rating +audience_score +best_pic_nom + best_actress_win,
                  data = clean_movies)
```

```
#summary(step9_model)
summary(step9_model)$adj.r.squared
```

```
## [1] 0.8277929
```

We chose to drop `best_actress_win` that has the largest p-value 0.331141.

```
step10_model <- lm(imdb_rating ~ genre + runtime + thtr_rel_month + dvd_rel_month +
                     dvd_rel_day + imdb_num_votes + critics_rating +critics_score +
                     audience_rating +audience_score +best_pic_nom,data = clean_movies)
#summary(step10_model)
summary(step10_model)$adj.r.squared
```

```
## [1] 0.8278085
```

We chose to drop `best_pic_nom` that has the largest p-value 0.26386.

```
step11_model <- lm(imdb_rating ~ genre + runtime + thtr_rel_month + dvd_rel_month +
                     dvd_rel_day + imdb_num_votes + critics_rating +critics_score +
                     audience_rating +audience_score,data = clean_movies)
#summary(step11_model)
summary(step11_model)$adj.r.squared
```

```
## [1] 0.8277363
```

We chose to drop `dvd_rel_month` that has the largest p-value 0.177900.

```
step12_model <- lm(imdb_rating ~ genre + runtime + thtr_rel_month +
                     dvd_rel_day + imdb_num_votes + critics_rating +critics_score +
                     audience_rating +audience_score,data = clean_movies)
summary(step12_model)$adj.r.squared
```

```
## [1] 0.8275007
```

We chose to drop `dvd_rel_day` that has the largest p-value 0.149349.

```
step13_model <- lm(imdb_rating ~ genre + runtime + thtr_rel_month +
                     imdb_num_votes + critics_rating +critics_score +
                     audience_rating +audience_score,data = clean_movies)
summary(step13_model)$adj.r.squared
```

```
## [1] 0.827189
```

This model is now p-value = 10% significant. Now we stop, and start using adjusted R^2. I dropped all of the existing features and there is no improvement in the adjusted- R^2. Therefore, we can conclude that this mode is our best model.

```
best_model <- step13_model
```

Here you can see the summary **backward elimination** process:

Step	Dropped Feature	p-value	adjusted-R^2
1	best_pic_win	0.89	0.8257
2	dvd_rel_year	0.85	0.8259
3	best_dir_win	0.70	0.8262
4	best_actor_win	0.63	0.8265
5	thtr_rel_day	0.53	0.8268
6	thtr_rel_year	0.55	0.8269
7	title_type	0.37	0.8271

Step	Dropped Feature	p-value	adjusted-R^2
8	top200_box	0.34	0.8274
9	mpaa_rating	0.39	0.8275
10	best_actress_win	0.33	0.8277
11	best_pic_nom	0.26	0.8278
12	dvd_rel_month	0.18	0.8277
13	dvd_rel_day	0.15	0.8272

One interesting point about the result is that the p-value and adjusted R^2 do not necessarily go hand to hand. We see that after 11th step, the adjusted R^2 start decreasing, while the features being deleted where not significant. Since, we put our process based on p-value, we continue choosing the results from p-value. ### Interpretation of the model parameters:

Let's take a look at our model and its parameters.

```
summary(best_model)
```

```
##
## Call:
## lm(formula = imdb_rating ~ genre + runtime + thtr_rel_month +
##     imdb_num_votes + critics_rating + critics_score + audience_rating +
##     audience_score, data = clean_movies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.33709 -0.17602  0.03561  0.25359  1.06277
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     2.659e+00  1.749e-01  15.200  < 2e-16 ***
## genreAnimation                 -4.329e-01  1.695e-01  -2.555 0.010876 *
## genreArt House & International  3.747e-01  1.435e-01   2.611 0.009263 **
## genreComedy                    -1.311e-01  7.501e-02  -1.747 0.081090 .
## genreDocumentary                3.120e-01  1.001e-01   3.117 0.001912 **
## genreDrama                      1.135e-01  6.497e-02   1.747 0.081166 .
## genreHorror                     8.259e-02  1.120e-01   0.737 0.461315
## genreMusical & Performing Arts  9.806e-02  1.457e-01   0.673 0.501171
## genreMystery & Suspense         2.717e-01  8.366e-02   3.248 0.001227 **
## genreOther                      1.587e-02  1.305e-01   0.122 0.903246
## genreScience Fiction & Fantasy -5.043e-02  1.686e-01  -0.299 0.765036
## runtime                         3.138e-03  1.094e-03   2.869 0.004264 **
## thtr_rel_month                  9.059e-03  5.274e-03   1.718 0.086397 .
## imdb_num_votes                  8.898e-07  1.965e-07   4.528 7.18e-06 ***
## critics_ratingFresh             8.504e-02  5.726e-02   1.485 0.137999
## critics_ratingRotten            3.414e-01  9.149e-02   3.731 0.000209 ***
## critics_score                   1.468e-02  1.528e-03   9.610  < 2e-16 ***
## audience_ratingUpright         -3.524e-01  7.352e-02  -4.793 2.07e-06 ***
## audience_score                  3.956e-02  2.107e-03  18.776  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4468 on 600 degrees of freedom
## Multiple R-squared:  0.8322, Adjusted R-squared:  0.8272
## F-statistic: 165.3 on 18 and 600 DF,  p-value: < 2.2e-16
```

- `genre` faeture is categorical, and its most significant lelve is the documentary lelvel. Based on the genre, it can be concluded that in general movie popularity increases when the movie is not Animation, Comedy, or Science Fiction, while it increases if the movie genre is Art, Documentary, Drama, Horror, Performing Arts, Mystery, or other. In general Art movies result in higher jump in popularity, while Animation result in higher decrease.
- `runtime` feature is numerical and has a very high significance. In general the longer movies tend to be more popular.
- `thtr_rel_month` also has almost high significance. However, this does make sense since in different months of the year, people might be more intrested in going to cinema and also better movies might be on air.

-`imdb_num_votes`, this feature is regarding the number of imdb votes of the movie which is proportionallu related to its popularity.

-`critics_rating`, this feature is categorical and has high correlation, especially on Rotten level. On average Rotten critics have 0.34 more IMDB rating than non-Rotten ones.

-`critics_score`, this feature is numerical and has a positive and very strong relationship with IMDB rating.

-`audience_raing`, it is categorical and has a very negative strong relationship with the IMDB rating,

-`audience_score`, it is numerical and has extremely strong positive relationship with IMDB rating.

4.2. Model Diagnosis

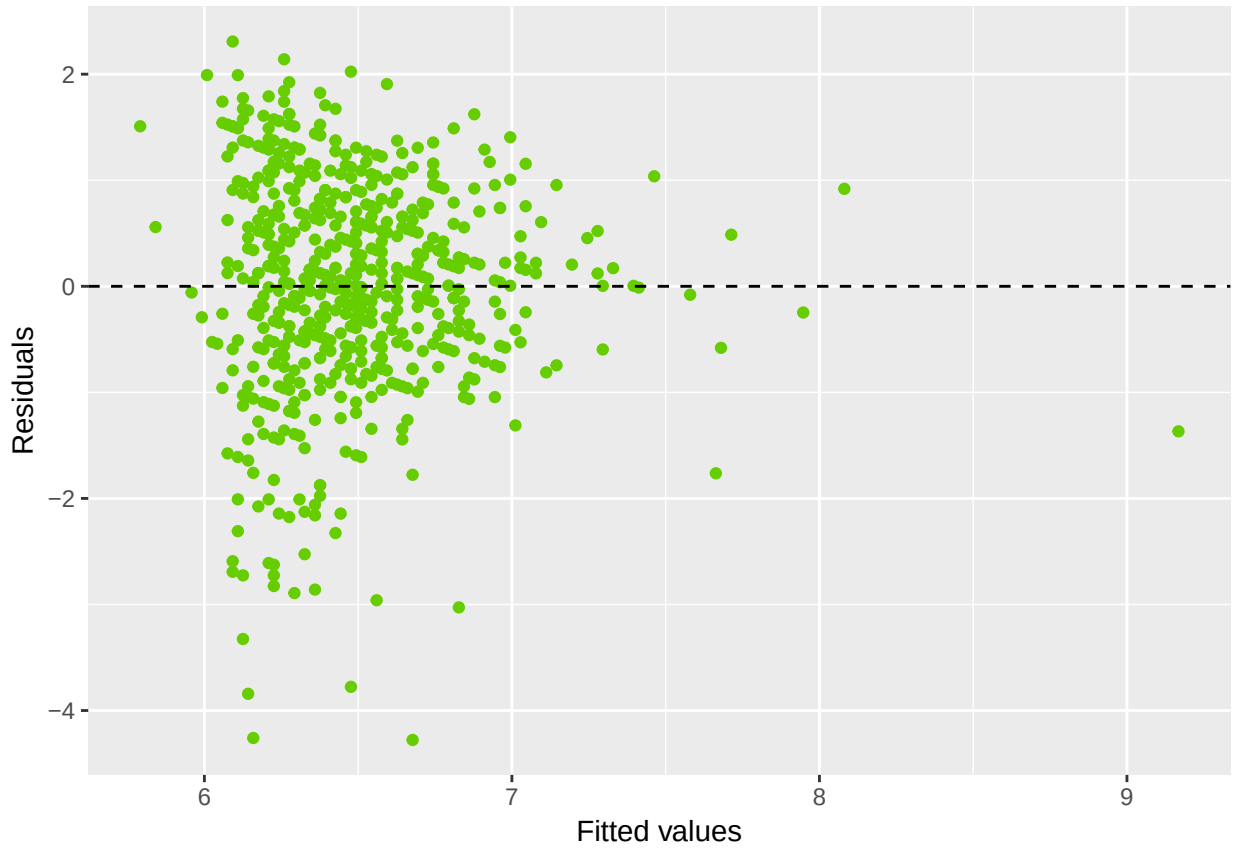
We have performed model diagnosis on two variables,`critics_score`, `audience_score`, in section 2, let's do diagnosis on the rest.

4.2.1. `runtime`, `imdb_rating`

We want to assure that this linear model is relaiable. We need to chec for (1) linearity, (2) nearly normal distribution, and (3) constant variability.

Linearity: We can also verify this using a plot of the residuals vs. fitted (predicted) values.

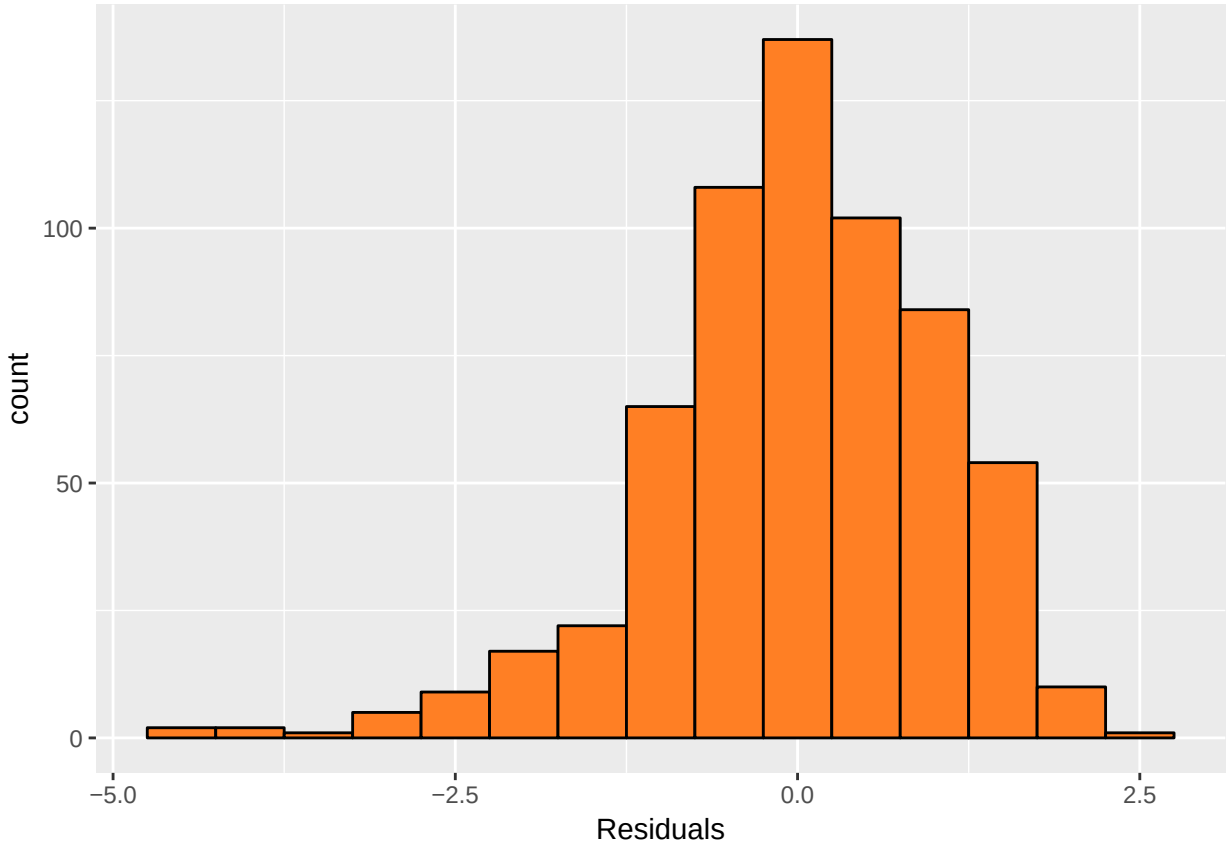
```
imdb_runtime<-lm(data=clean_movies, imdb_rating~runtime)
ggplot(data = imdb_runtime, aes(x = .fitted, y = .resid)) +
  geom_point(color = 'chartreuse3') +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Fitted values") +
  ylab("Residuals")
```



The plot seems to be randomly distributed. However, there seems to be some outlier on the right side of the graph.

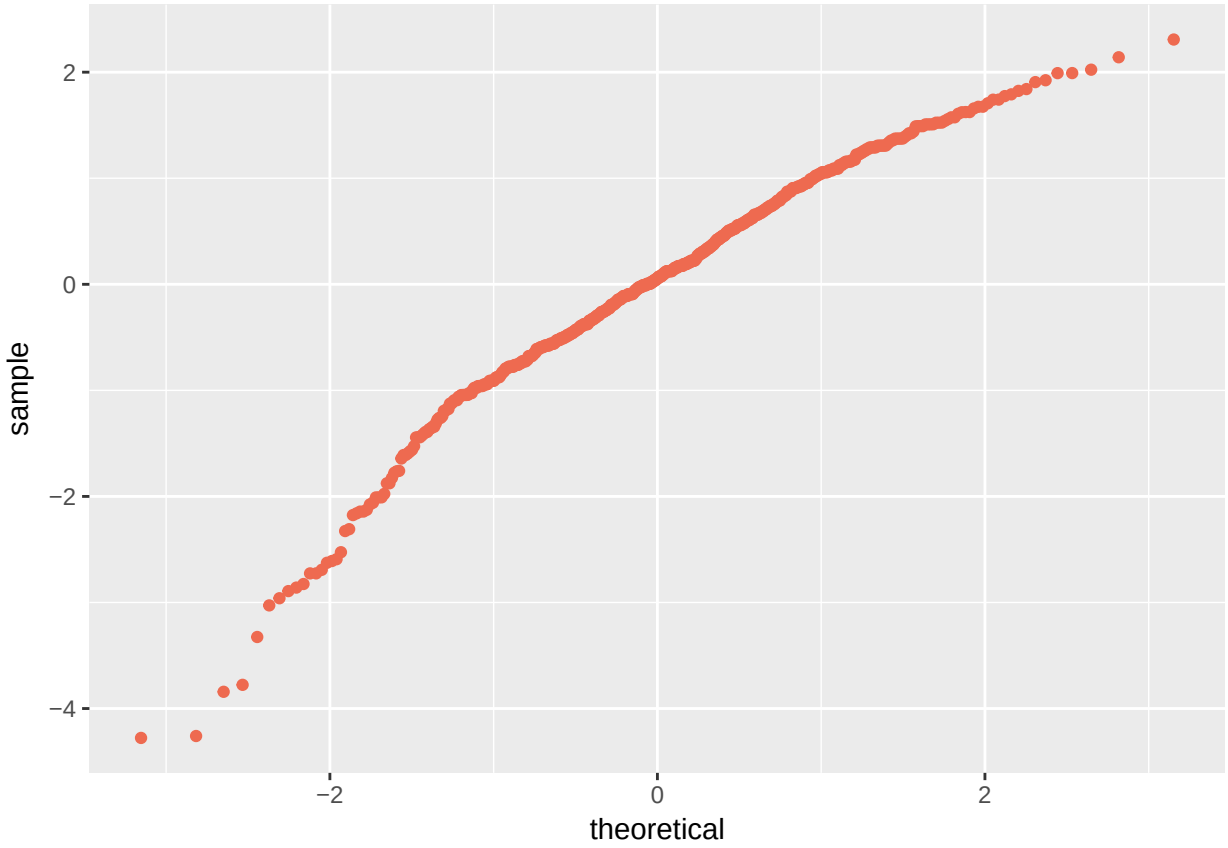
Nearly Normal Residuals: To check this condition, we will check the histograms.

```
ggplot(data = imdb_runtime, aes(x = .resid)) +
  geom_histogram(binwidth = .5, fill = 'chocolate1', color = 'black' ) +
  xlab("Residuals")
```



There seems to be some symmetry around 0 and data could be roughly considered normal. We can also use the normal probablity plot of residuals.

```
ggplot(data = imdb_runtime, aes(sample = .resid)) +
  stat_qq(color = 'coral2')
```

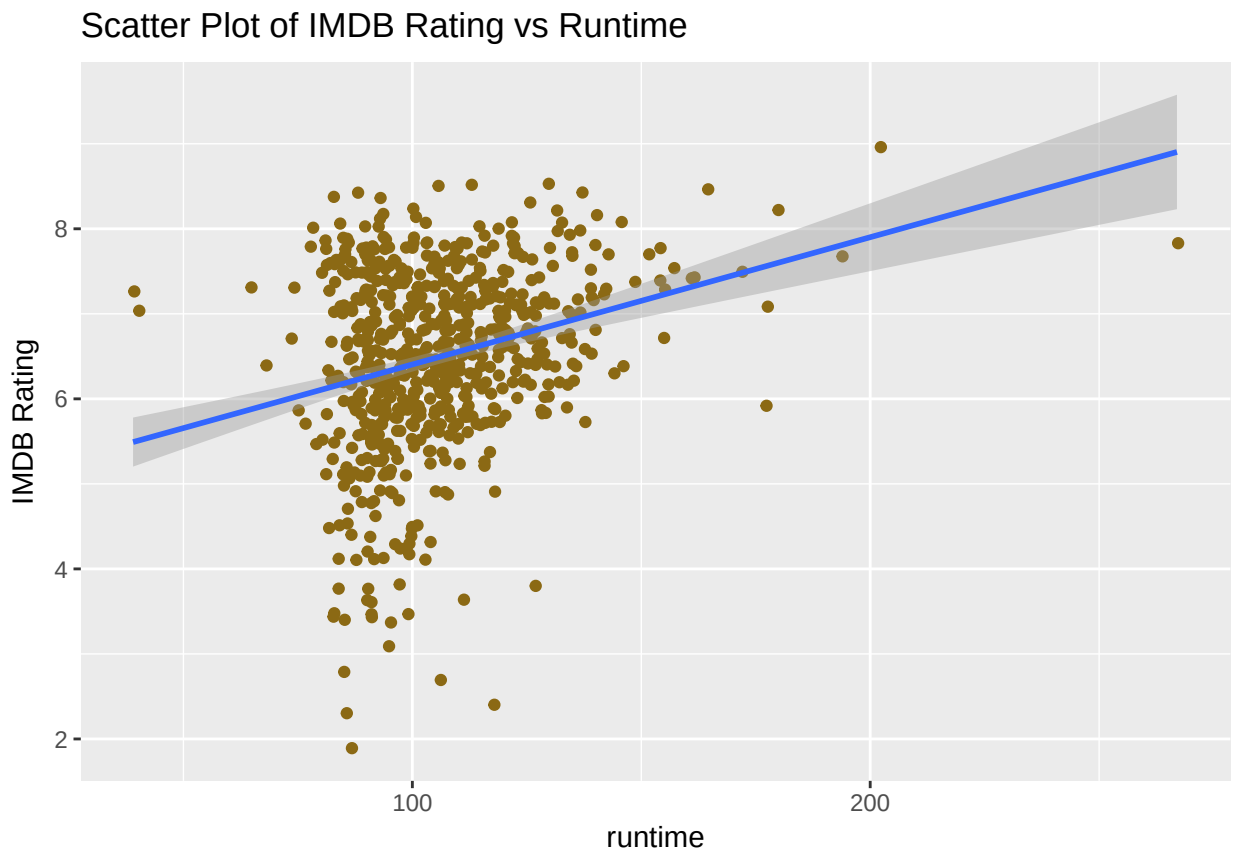


This relationship also seems to be linear. Let's take a look at the scatter plot to assure the non-linearity.

```
ggplot(movies, aes(x=runtime, y=imdb_rating))+geom_jitter(color='goldenrod4')+
  labs(x="runtime", y="IMDB Rating",
       title = 'Scatter Plot of IMDB Rating vs Runtime')+
  stat_smooth(method = "lm", se = TRUE)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



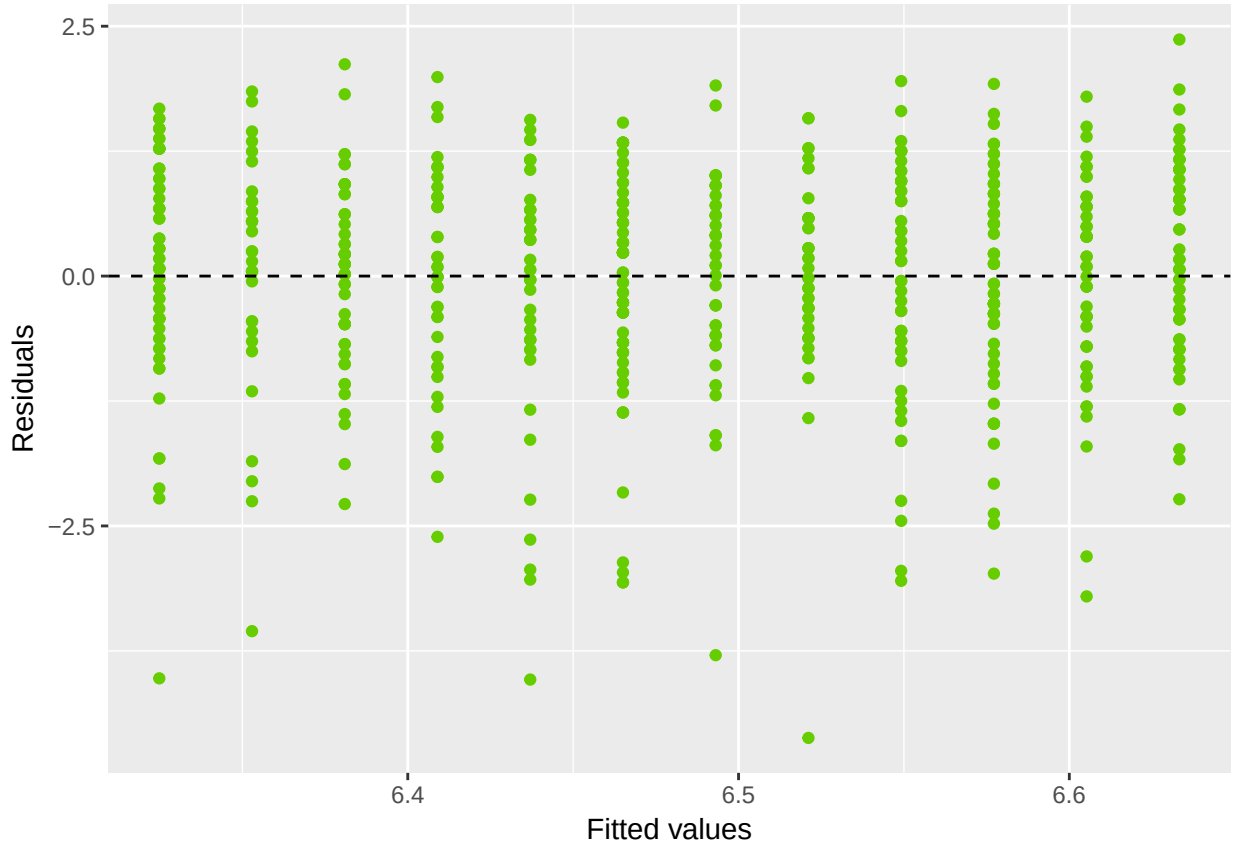
The run time feature, though showed very good significance, but it doesn't pass the model diagnosis. It is not linear, though residuals somehow normaly distributed around zero.

4.2.2. `thtr_rel_month, imdb_rating`

We want to assure that this linear model is relaiable. We need to chec for (1) linearity, (2) nearly normal distribution, and (3) constant variability.

Linearity: We can verify this using a plot of the residuals vs. fitted (predicted) values.

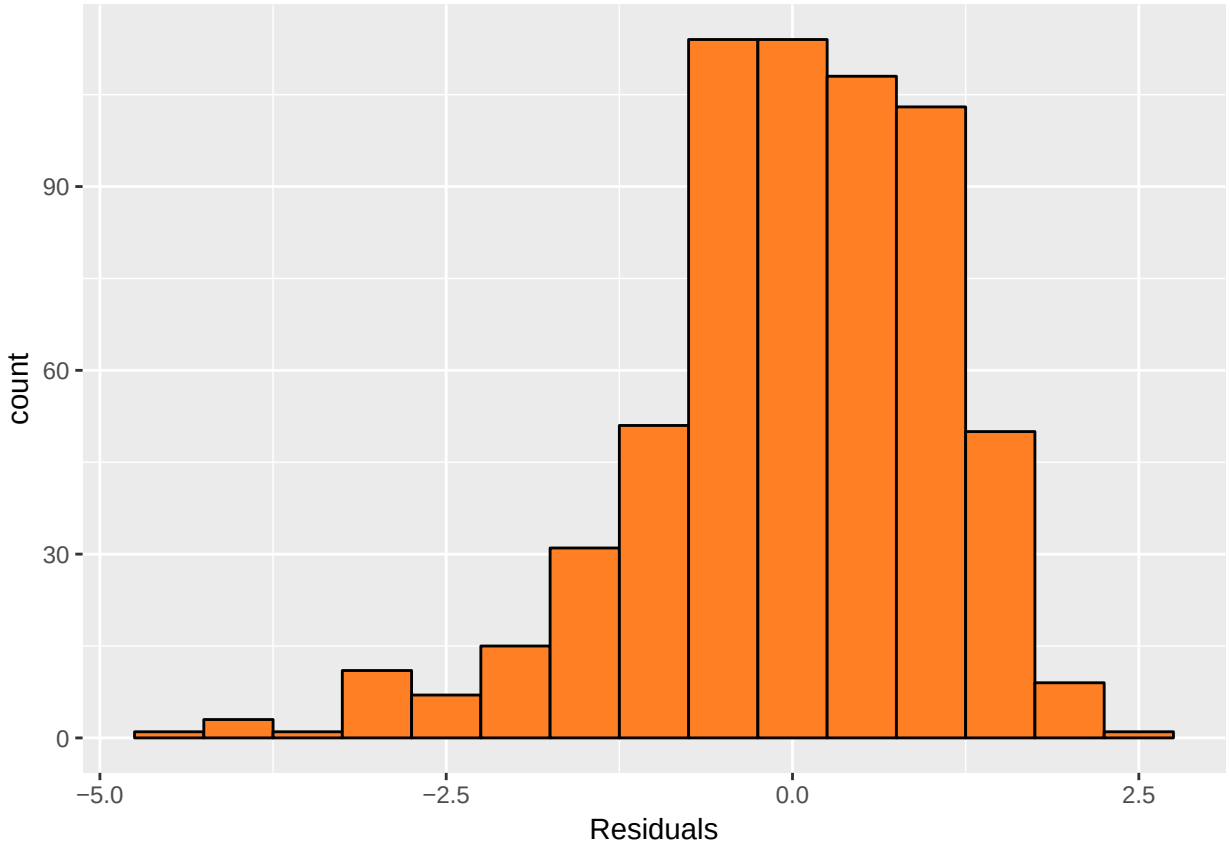
```
imdb_thtr_rel_month<-lm(data=clean_movies, imdb_rating~thtr_rel_month)
ggplot(data = imdb_thtr_rel_month, aes(x = .fitted, y = .resid)) +
  geom_point(color = 'chartreuse3') +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Fitted values") +
  ylab("Residuals")
```



The plot seems to be randomly distributed around zero.

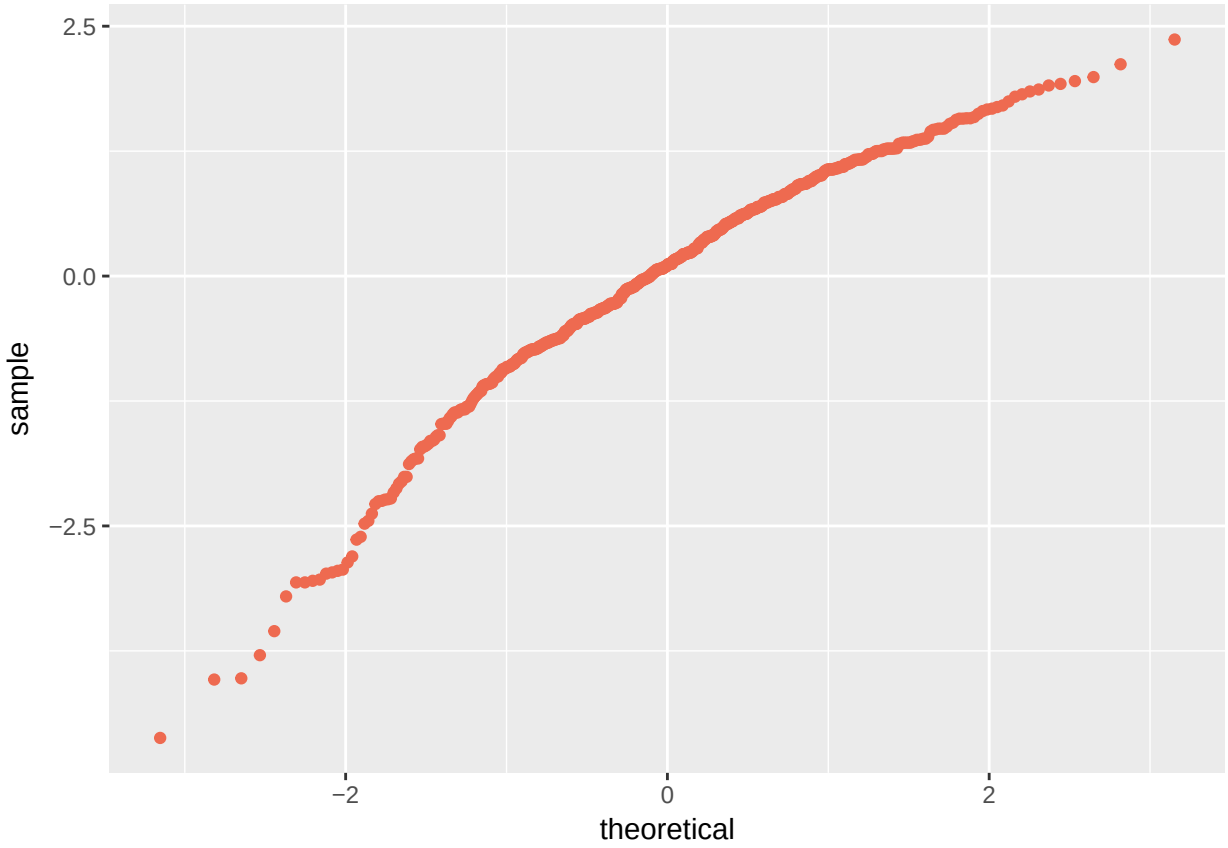
Nearly Normal Residuals: To check this condition, we will check the histograms.

```
ggplot(data = imdb_thtr_rel_month, aes(x = .resid)) +
  geom_histogram(binwidth = .5, fill = 'chocolate1', color = 'black' ) +
  xlab("Residuals")
```



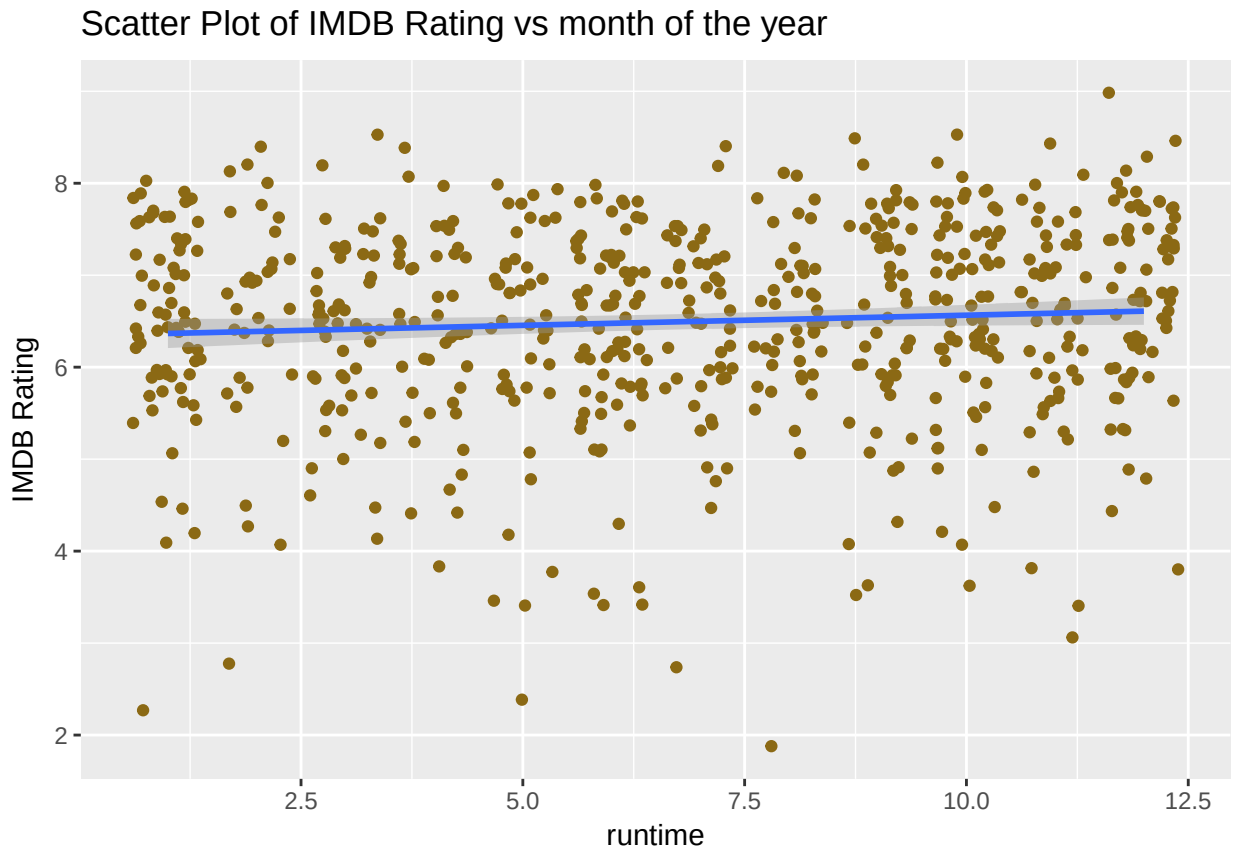
There seems to be some symmetry around 0 and data could be roughly considered normal. But the pick is not at zero. We can also use the normal probablity plot of residuals.

```
ggplot(data = imdb_thtr_rel_month, aes(sample = .resid)) +
  stat_qq(color = 'coral2')
```



This relationship also seems to be curved. Let's take a look at the scatter plot to assure the non-linearity.

```
ggplot(movies, aes(x=thtr_rel_month, y=imdb_rating))+geom_jitter(color='goldenrod4')+
  labs(x="runtime", y="IMDB Rating",
       title = 'Scatter Plot of IMDB Rating vs month of the year')+
  stat_smooth(method = "lm", se = TRUE)
```



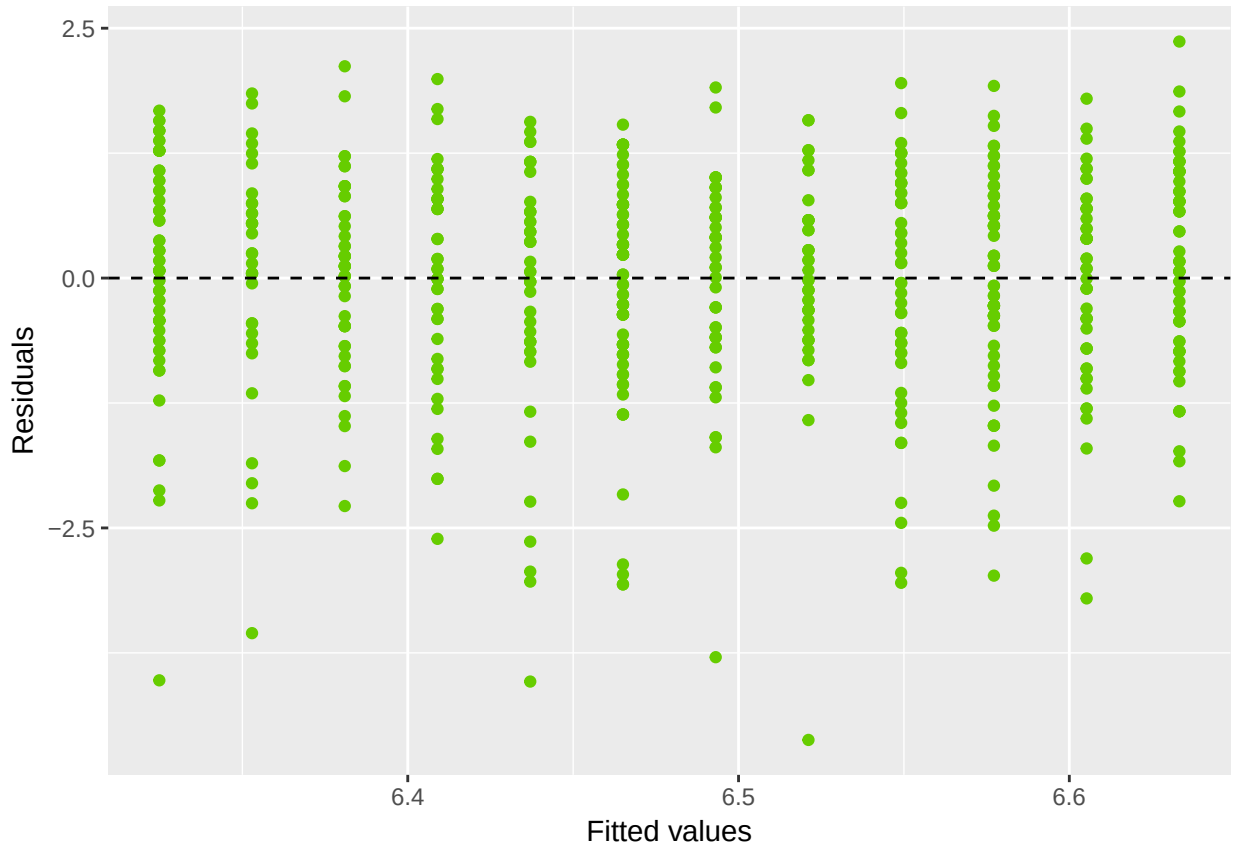
The run time feature, though showed very good significance, but it doesn't pass the model diagnosis. It is not linear, though residuals somehow normaly distributed around zero. We will drop it from our model.

4.2.3. `imdb_num_votes, imdb_rating`

We want to assure that this linear model is relaiable. We need to chec for (1) linearity, (2) nearly normal distribution, and (3) constant variability.

Linearity: We can verify this using a plot of the residuals vs. fitted (predicted) values.

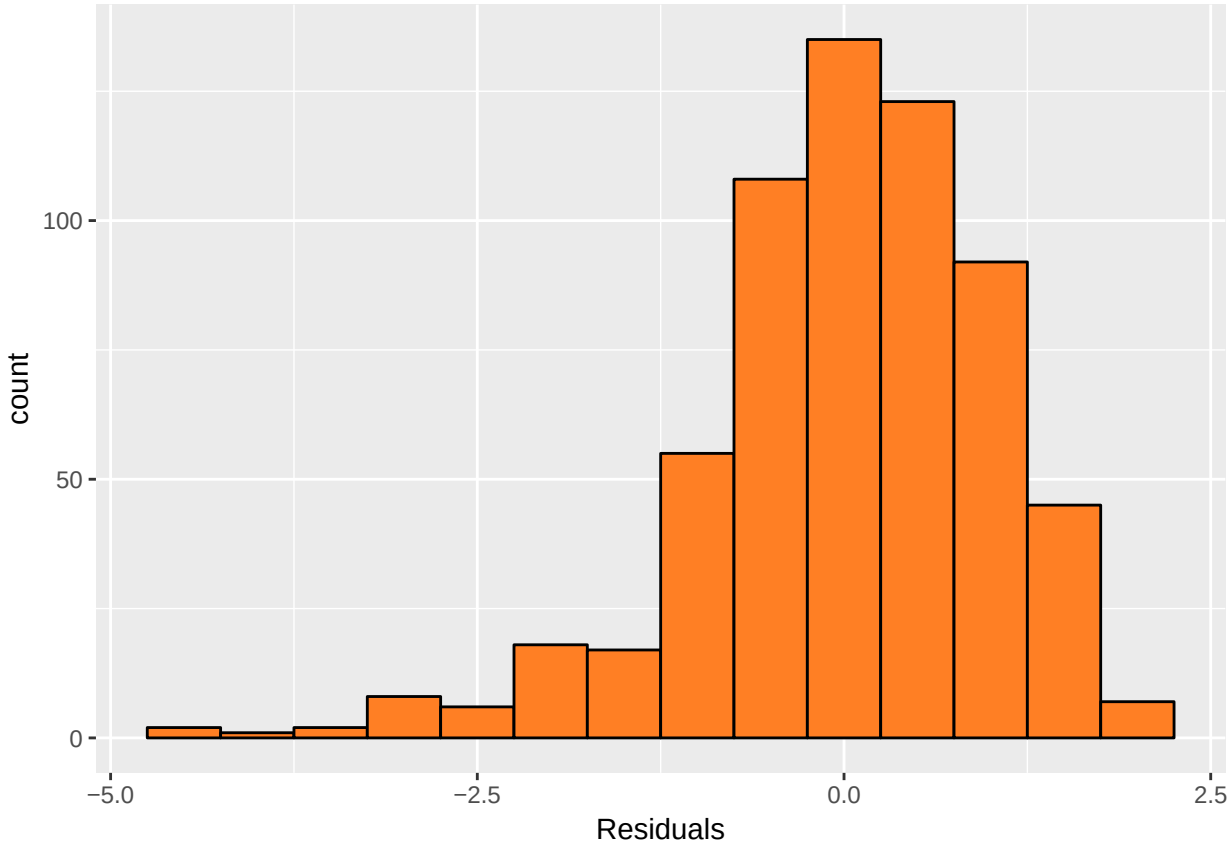
```
imdb_imdb_num_votes<-lm(data=clean_movies, imdb_rating~imdb_num_votes)
ggplot(data = imdb_thtr_rel_month, aes(x = .fitted, y = .resid)) +
  geom_point(color = 'chartreuse3') +
  geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Fitted values") +
  ylab("Residuals")
```



The plot seems to be randomly distributed around zero.

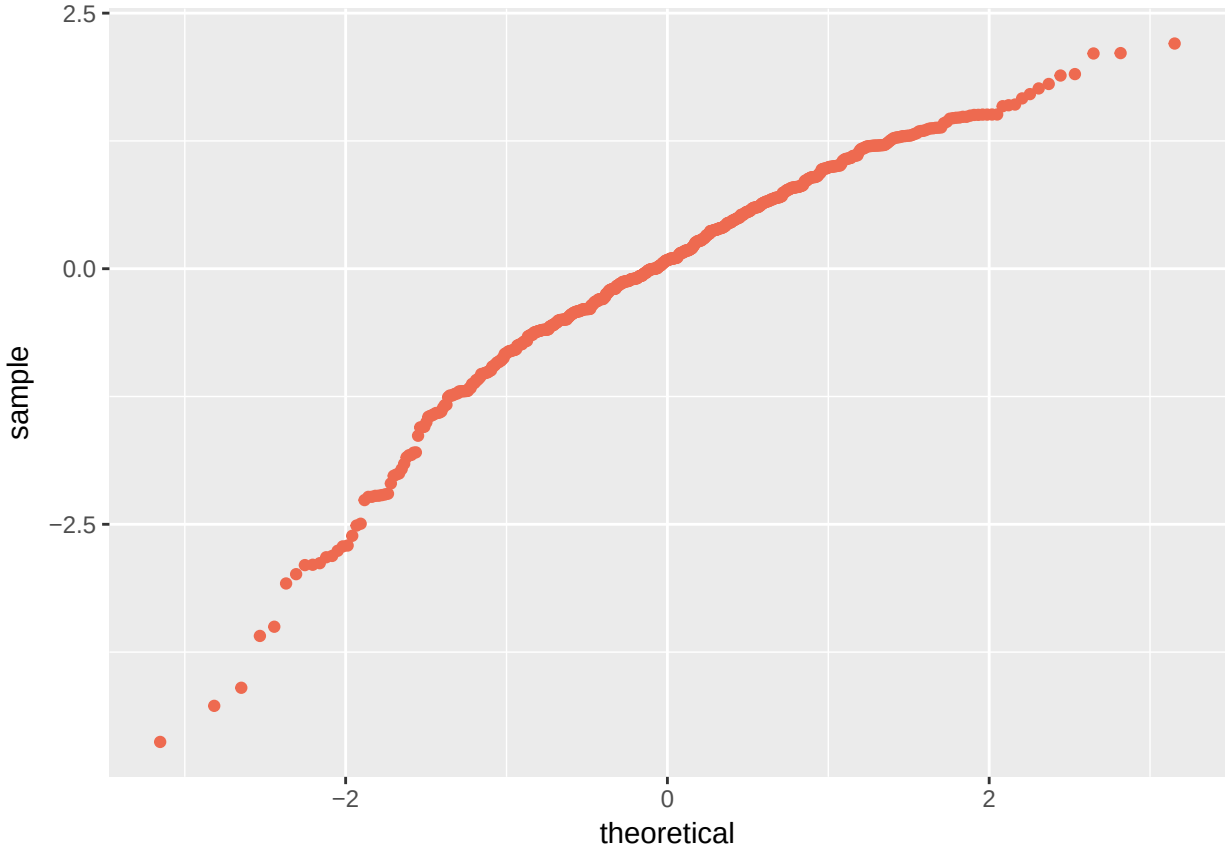
Nearly Normal Residuals: To check this condition, we will check the histograms.

```
ggplot(data = imdb_imdb_num_votes, aes(x = .resid)) +
  geom_histogram(binwidth = .5, fill = 'chocolate1', color = 'black' ) +
  xlab("Residuals")
```



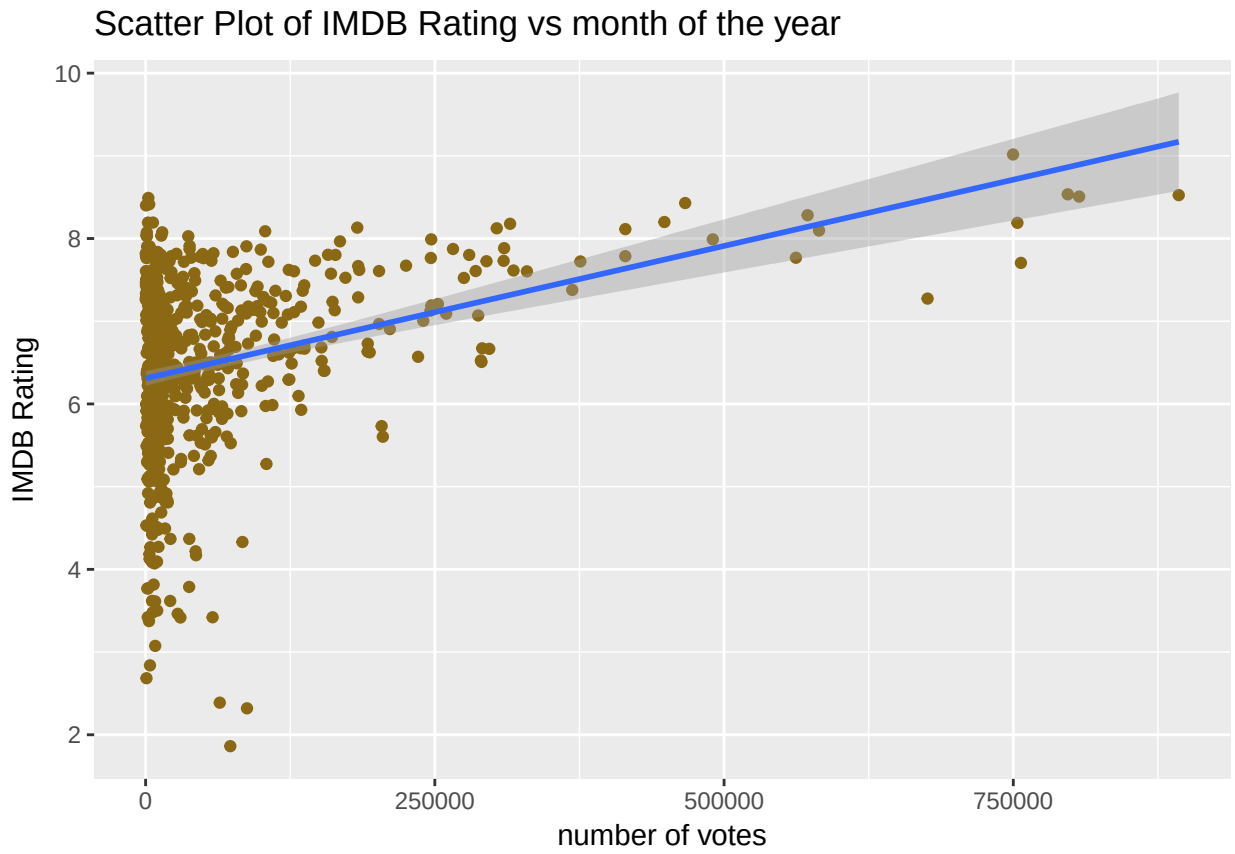
There seems to be some symmetry around 0 and data could be roughly considered normal. But the pick is not at zero. We can also use the normal probablity plot of residuals.

```
ggplot(data = imdb_imdb_num_votes, aes(sample = .resid)) +
  stat_qq(color = 'coral2')
```



This relationship also seems to be a bit curved. Let's take a look at the scatter plot to assure the non-linearity.

```
ggplot(movies, aes(x=imdb_num_votes, y=imdb_rating))+geom_jitter(color='goldenrod4')+
  labs(x="number of votes", y="IMDB Rating",
       title = 'Scatter Plot of IMDB Rating vs month of the year')+
  stat_smooth(method = "lm", se = TRUE)
```



The number of votes feature, though showed very good significance, but it doesn't pass the model diagnosis. It is not linear and has many high leverage outliers, though residuals somehow normaly distributed around zero. We will drop it from our model.

Let's drop these three features and develop the final model.

```
final_model <- lm(imdb_rating ~ genre + critics_rating +critics_score +
                    audience_rating +audience_score,data = clean_movies)
summary(final_model)$adj.r.squared
```

```
## [1] 0.8141786
```

We lost about 2.3% of the adjusted R^2, which means we won't be able to explain 2.3% of the variability, but we assure our model is healthy. * * *

Part 5: Prediction

Now, we want to find the movie score for a movie that has not been in this list, using our `final_model`. Movie should be from the same population, movies before 2016.

We chose `Batman Begins (2005)`. Let's make a dataframe from the results.

```
new_movie <- data.frame(genre='Action & Adventure' ,  critics_rating='Fresh' ,critics_score=84,
                    audience_rating='Upright', audience_score=94)
predict(final_model, new_movie, interval = "prediction", level = 0.95)
```

```
##        fit      lwr      upr
## 1 7.862699 6.939465 8.785933
```

Hence, the model predicts, with 95% confidence, that `Batman Begins` has an IMDB score between 6.8 and 8.8, which is the correct prediction as has 8.3 score.

Here is the reference to movie websites: IMDB, Rotten Tomatoes * * *

Part 6: Conclusion

In this work we performed a study on the movie data acquired from IMDB and Rotten Tomato APIs. The goal of the study was to predict the popularity of the movie (its IMDB rating) from the available data. We performed a linear regression backward elimination to develop the model. Model selection uses p-value as a criterion. Moreover, we conclude that we need to drop three variables as they do not comply with the model diagnosis. Finally, we developed a 95% interval for the prediction of Batman Begins. Our prediction was correct and the actual IMDB rating was inside the confidence interval. Our proposed research questions demonstrated high capability. We need to state that although 81% of the variability was explained by the proposed features, we still require more features to explain the variability.

”