Modeling and prediction for movies

Setup

Load packages

```r
library(ggplot2)
library(dplyr)
library(statsr)
library(GGally)
library(gridExtra)
library(cowplot)
```

Load data

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

Part 1: Data

Data was taken randomly sample from movies produced and released before 2016.

Part 2: Research question

What are the minimum characteristics a movie should have to get a decent rating score?

Part 3: Exploratory data analysis

There are different kinds of movie genres, but, do they have something in common? Certeanly there is. How can we build a winning formula to asses such goal. Let's find out with this Dataset sampled from Rotten Tomatoes and IMBD.

First, let's check the Dataset structure. We can visualize this with RStudio Viewer, but we can also accomplish this with Code.

```r
names(movies)
```

```
##  [1] "title"            "title_type"       "genre"            "runtime"
##  [5] "mpaa_rating"      "studio"           "thtr_rel_year"    "thtr_rel_month"
##  [9] "thtr_rel_day"     "dvd_rel_year"     "dvd_rel_month"    "dvd_rel_day"
## [13] "imdb_rating"      "imdb_num_votes"   "critics_rating"   "critics_score"
## [17] "audience_rating"  "audience_score"   "best_pic_nom"     "best_pic_win"
## [21] "best_actor_win"   "best_actress_win" "best_dir_win"     "top200_box"
## [25] "director"         "actor1"           "actor2"           "actor3"
## [29] "actor4"           "actor5"           "imdb_url"         "rt_url"
```

Above command, show the column names, this is useful when do have access to a CodeBook.

Let's moving checking the structure of the Dataset.

```r
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/
```

That was the Data type of the Dataset.

Moving Forward, Let's check how this dataset Classifies the movies as good, indiferent or bad. Let's Rembember, the sample comes from two different Web Sites. Rotten Tomatoes and IMDb.

- critics_rating : Categorical variable for critics rating on Rotten Tomatoes (Certified Fresh, Fresh, Rotten)
- critics_score : Critics score on Rotten Tomatoes
- imdb_rating : Rating on IMDB
- imdb_num_votes : Number of votes on IMDB

So let's inspect them. I'll add the title variable. To get a bigger picture.

```
movies %>% select(title, critics_rating, critics_score, imdb_rating, imdb_num_votes) %>% head(5)
```

```
## # A tibble: 5 x 5
##   title                  critics_rating  critics_score imdb_rating imdb_num_votes
##   <chr>                  <fct>                   <dbl>       <dbl>          <int>
## 1 Filly Brown            Rotten                     45         5.5            899
## 2 The Dish               Certified Fresh            96         7.3          12285
## 3 Waiting for Guffman    Certified Fresh            91         7.6          22381
## 4 The Age of Innocence Certified Fresh              80         7.2          35096
## 5 Malevolence            Rotten                     33         5.1           2386
```

There it is, the first five rows of the selection.

But according to the

as well, there is a runtime variable, which has the duration length of a movie. Let's do a summary of this variable.

```
movies %>% select(runtime) %>% summary(runtime)
```

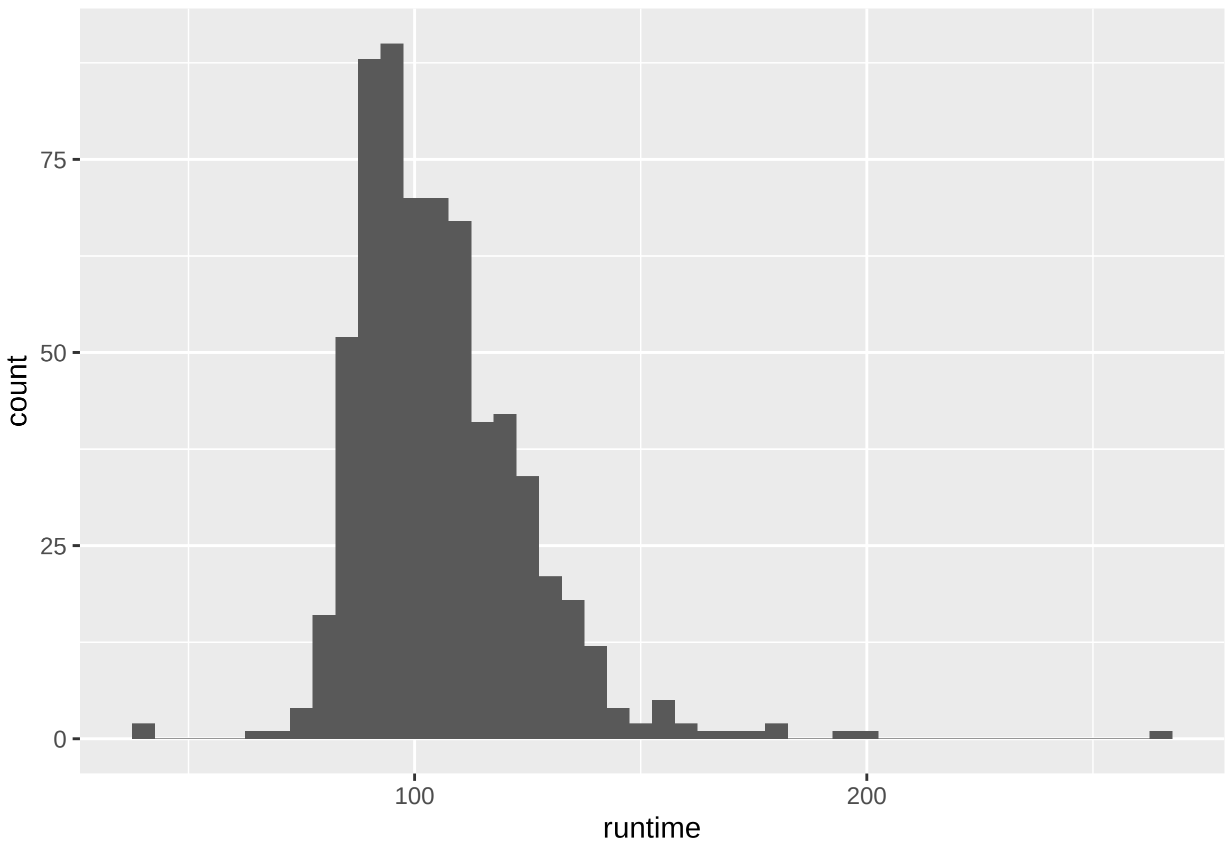
```
##     runtime
##  Min.   : 39.0
##  1st Qu.: 92.0
##  Median :103.0
##  Mean   :105.8
##  3rd Qu.:115.8
##  Max.   :267.0
##  NA's   :1
```

The minimum a movie lasted is 39 minutes, maximum 267 minutes, on average a movie last 105 minutes.

However, before deciding this mean is truthful, let's plot the data. Remember, when working with the mean, is a good practice to plot the data to verify for outliers, mean is really sensitive to the outliers, therefore sometimes is not a good option as a central tendancy measure.

```
ggplot(data = movies, aes(x = runtime)) + geom_histogram(binwidth = 5)
```

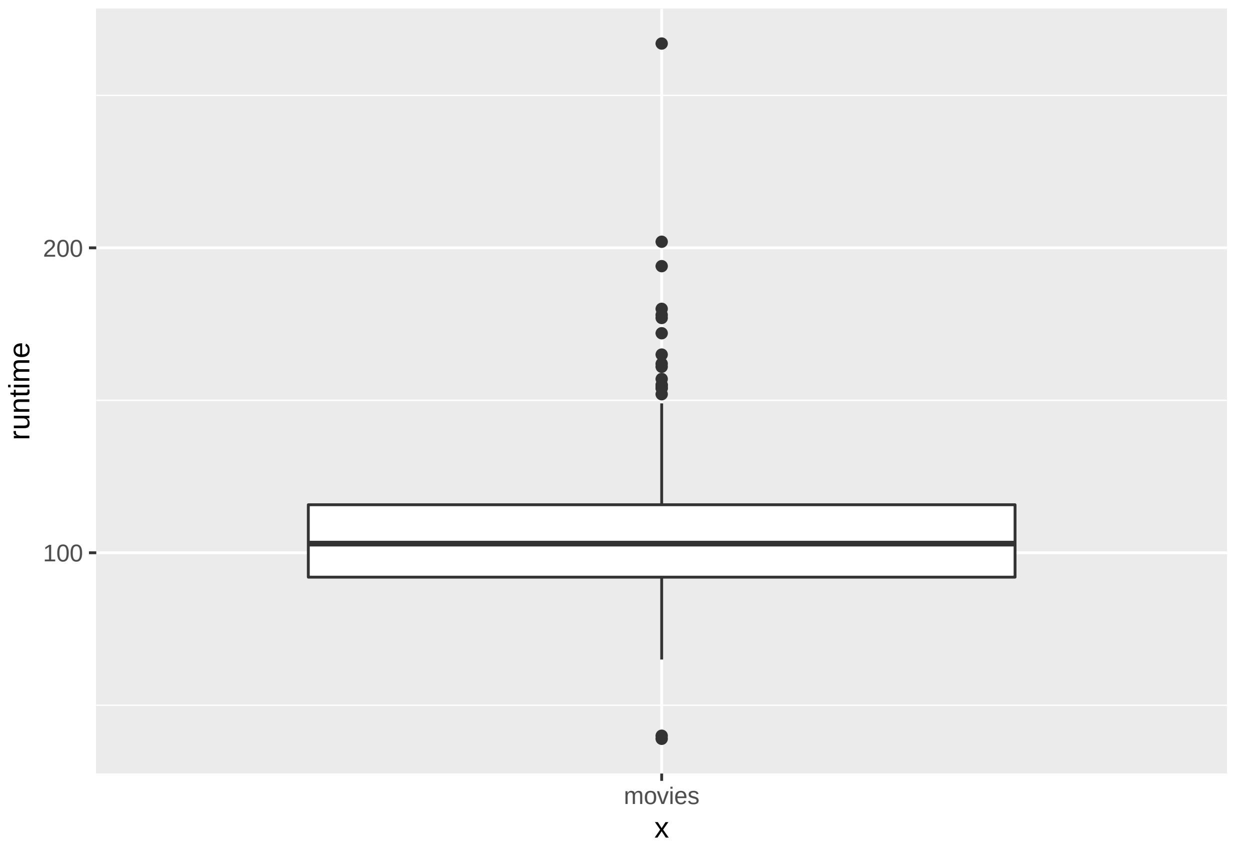
```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```



Viewing the plot, it shows there are a few outliers and skweness to the right.

```
ggplot(data = movies, aes(x= "movies", y  = runtime)) + geom_boxplot()
```

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



In the boxplot we easily can observe the are some outliers. So, the meean will be not a good central tendency pick. However, this is for the whole sample. Let's actually build for good movies and bad movies in relation with the scores from Rotten Tomatoes and IMBd.

Rotten Tomatoes Plots

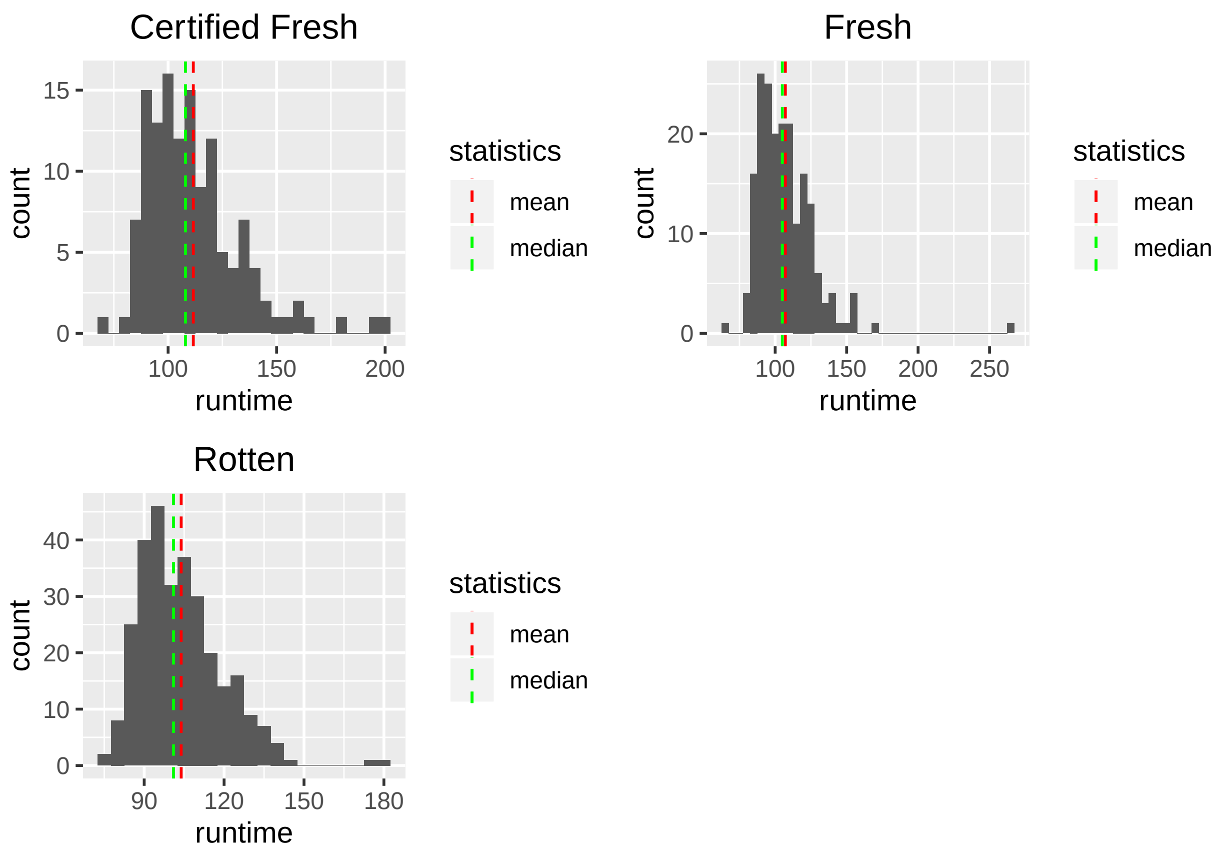
Let's build indivual plots for each classification, to visualize how the data is distributed. Let's stard building histograms of the Rotten Tomatoes.

First, let's show a summary of the statitics.

```
na.omit(movies) %>% group_by(critics_rating) %>% summarise(runtime_mean = mean(runtime), runtime_median
```

```
## # A tibble: 3 x 3
##   critics_rating  runtime_mean runtime_median
##   <fct>                  <dbl>          <dbl>
## 1 Certified Fresh         112.            108
## 2 Fresh                   107.            105
## 3 Rotten                  104.            101
```

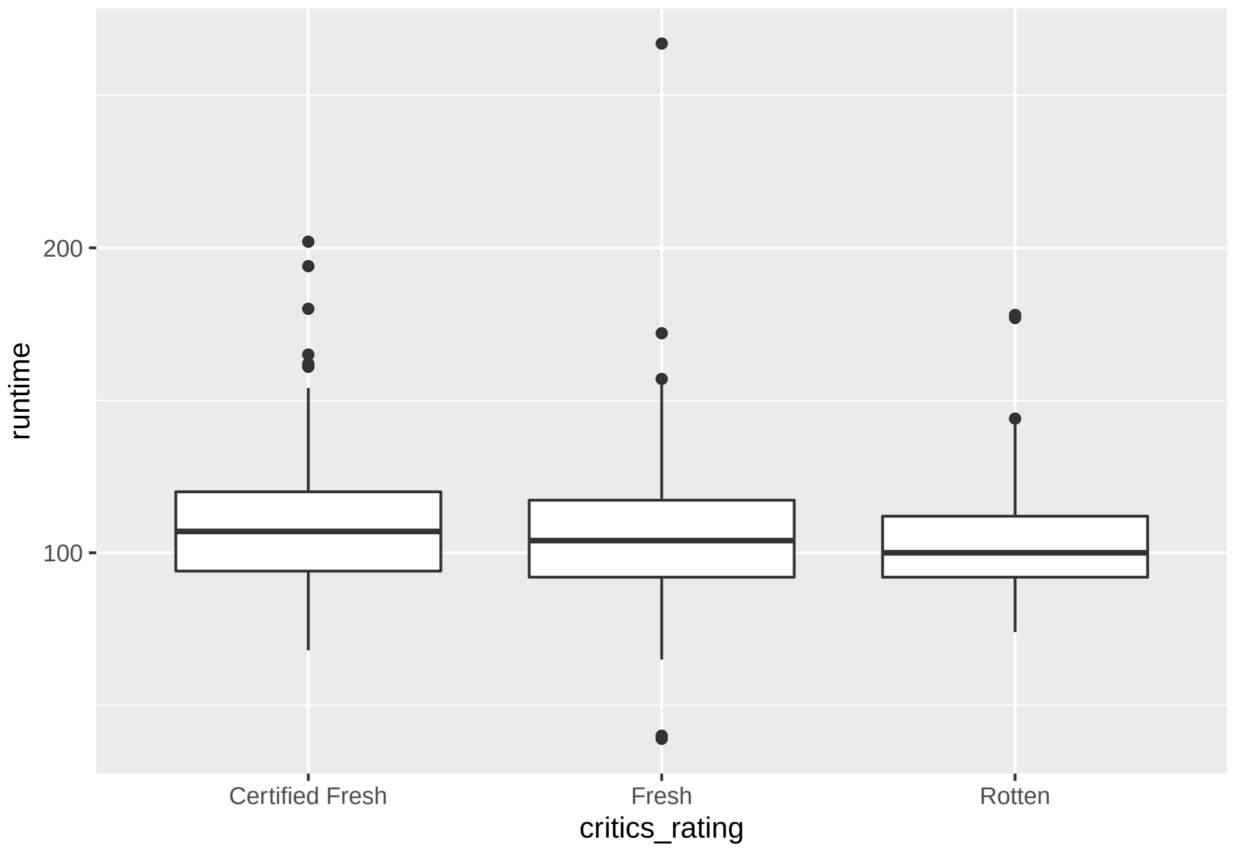
```
g1 <- na.omit(movies) %>% filter(critics_rating == "Certified Fresh")  %>% ggplot(aes(x = runtime)) + g

g2 <- na.omit(movies) %>% filter(critics_rating == "Fresh")  %>% ggplot(aes(x = runtime)) + geom_histog

g3 <- na.omit(movies) %>% filter(critics_rating == "Rotten")  %>% ggplot(aes(x = runtime)) + geom_histog


plot_grid(g1,g2,g3)
```



We can observe a Rigth Skewness of each plot. It seems there are some oultliers. Let's visualize this in a boxplot and because the data is rigth skwed.

```
movies %>% ggplot(aes(x = critics_rating, y  = runtime)) + geom_boxplot()
```

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



There it is. Apperently, the mean stay around the 100 minutes, but still, there are obvious outliers. We will deal this later.

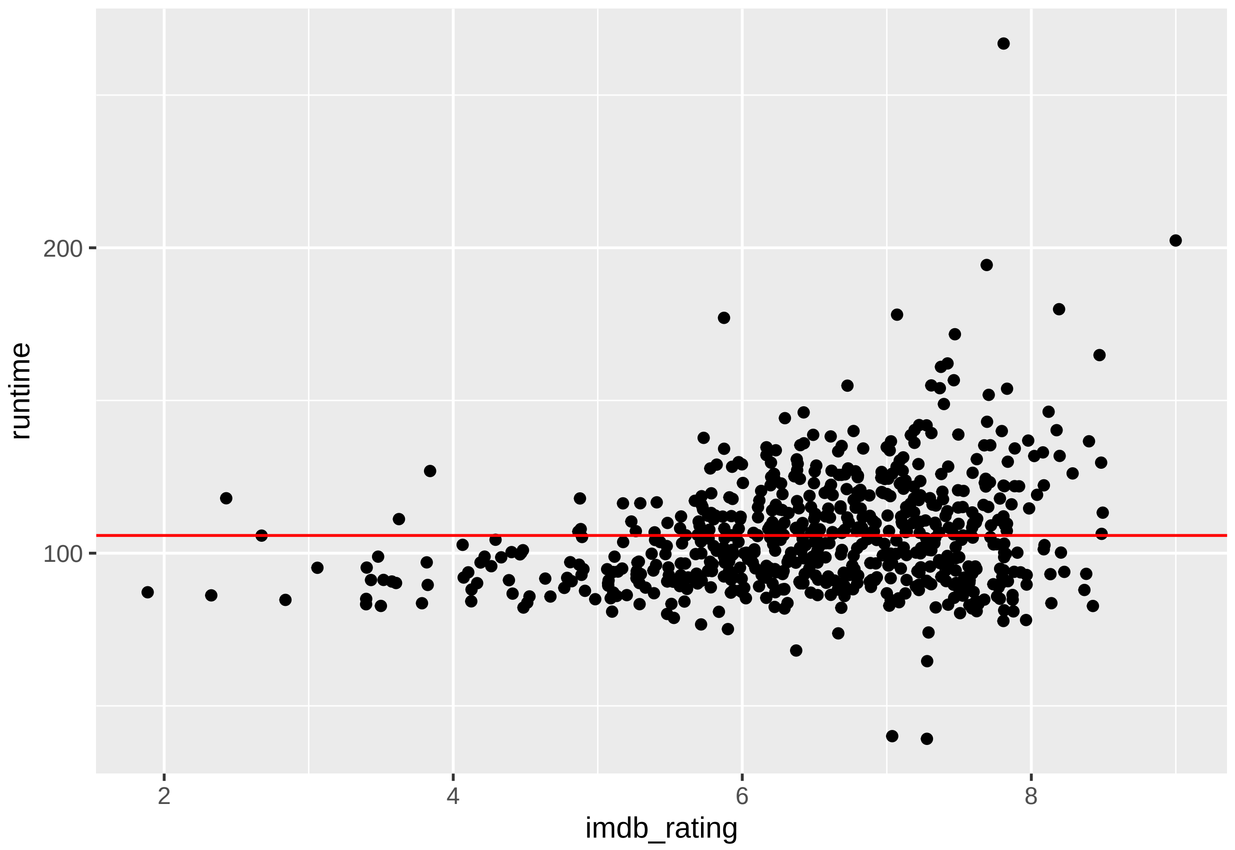
Moving on let's plot the IMBd.

IMBd Plot.

Unlike Rotten Tomatoes data, with IMBd we are dealing with numerical values, therefore we need a different kind of plot.

```
movies %>% select(runtime, imdb_rating) %>% ggplot(aes( x= imdb_rating, y = runtime)) + geom_jitter()
```

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



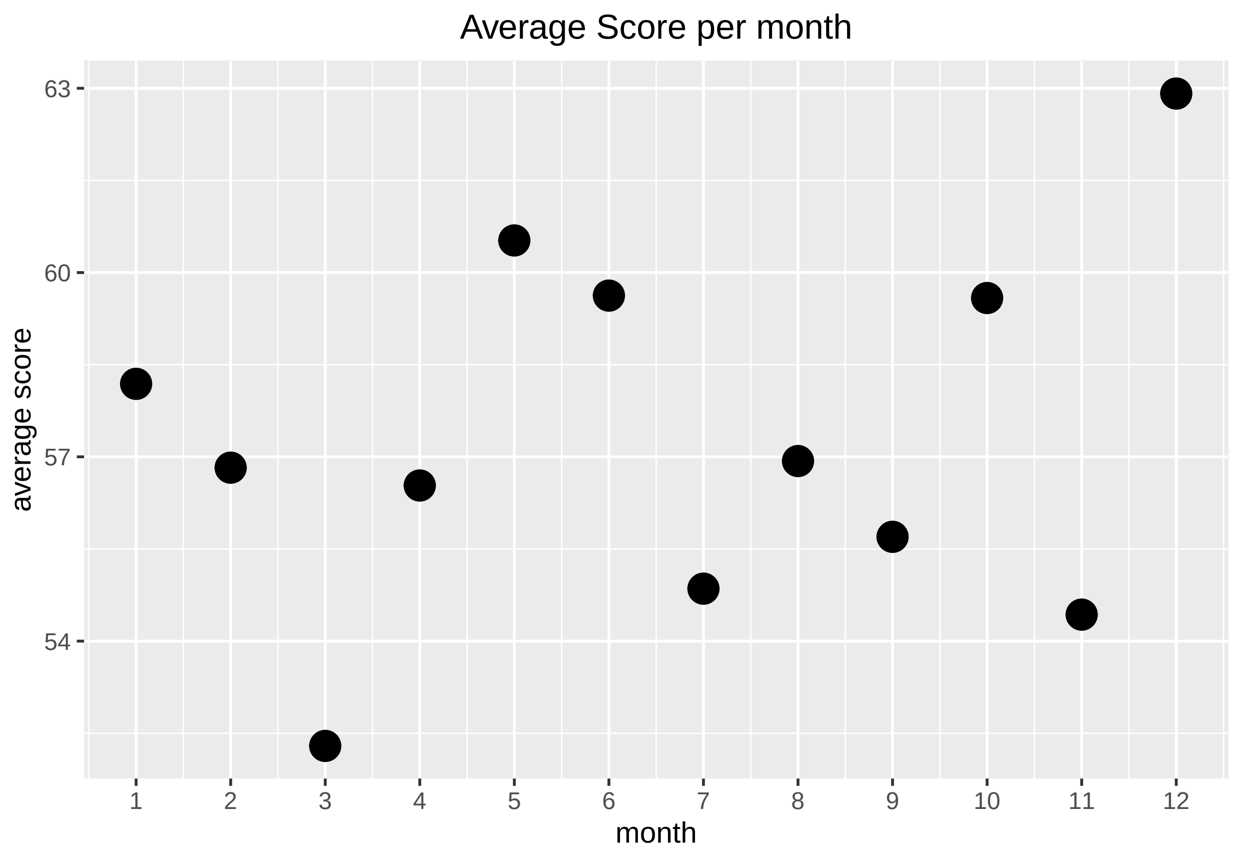
Average movie length is 105, and we plot a horizontal line to cut the data and see how density is the data around that line, we can observe the most data in the rating from 6 to 8.

Alrigth, but we don't like outliers we need a more robust tool, let's pick up the median as our central tendency tool.

Let's inspect the movies popularity by Date.

```
sum_scores_crit <- movies %>% group_by(thtr_rel_month) %>% summarise(mean_critics_score = mean(critics_s
```

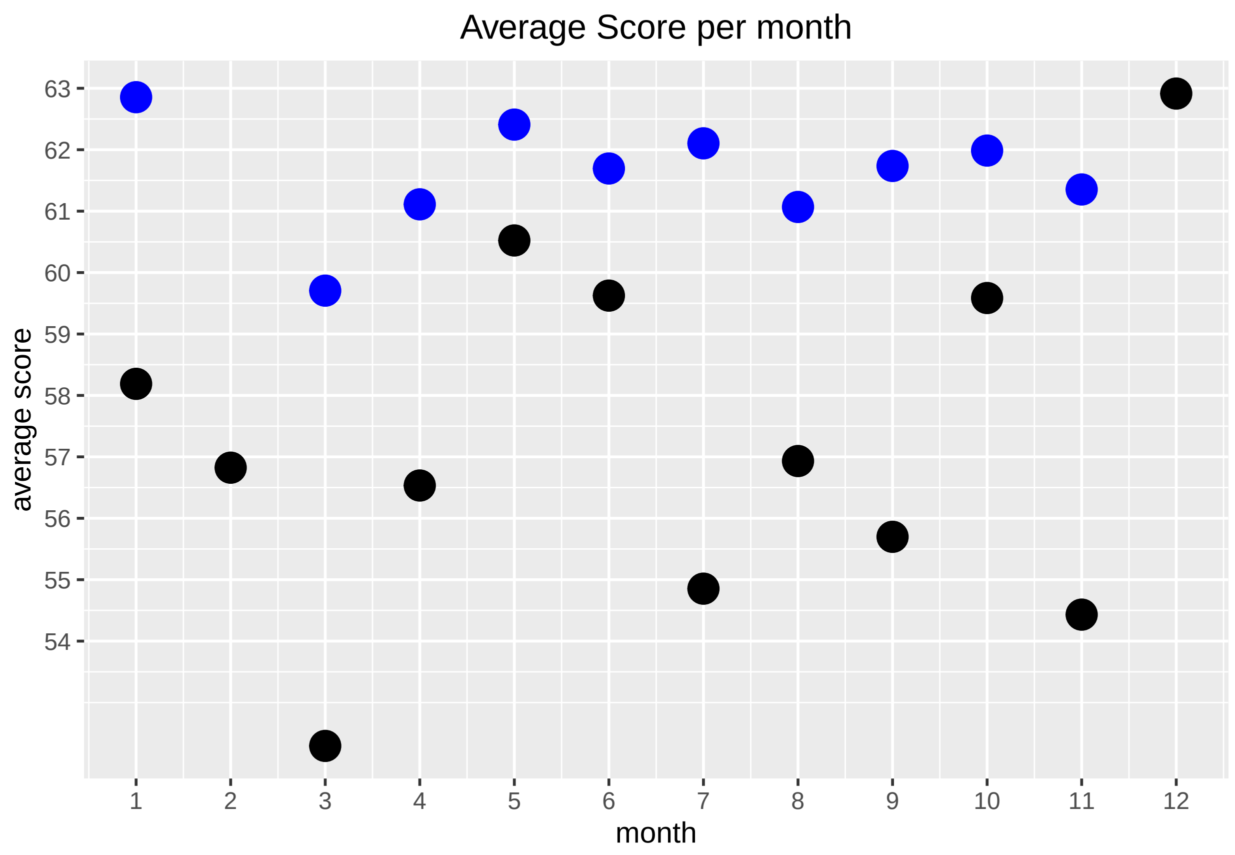
```
ggplot(data= sum_scores_crit, aes(x= thtr_rel_month, y = mean_critics_score )) + geom_point(size=5) + sc
```



It seems movies released in december tend to score better, because their average score is greather than the others. But these are the critics opinions. Does the audience thinks the same?

```
aud_score <- movies %>% group_by(thtr_rel_month) %>% summarise(mean_audience = mean(audience_score))
g5<-ggplot(data= sum_scores_crit, aes(x= thtr_rel_month, y = mean_critics_score )) + geom_point(size=5)
g5 + geom_point( y = aud_score$mean_audience, size = 5, color="blue") + scale_y_continuous(breaks = c(5
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
```



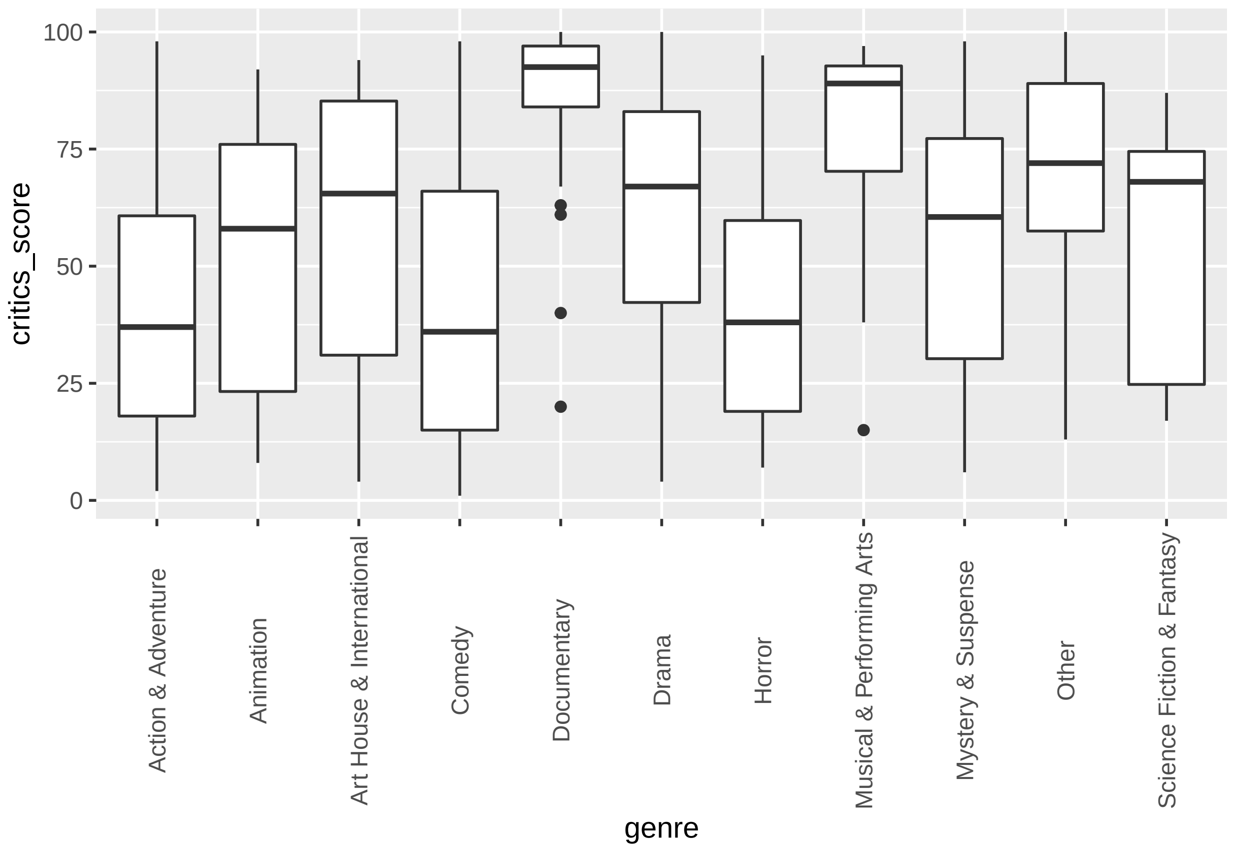
it Seems audience tends to be nicer when critizing, because they give higher score, but still, movies released in december gets the highest score. Following table tell us that

```
aud_score
```

```
## # A tibble: 12 x 2
##    thtr_rel_month mean_audience
##             <dbl>         <dbl>
##  1              1          62.9
##  2              2          63.9
##  3              3          59.7
##  4              4          61.1
##  5              5          62.4
##  6              6          61.7
##  7              7          62.1
##  8              8          61.1
##  9              9          61.7
## 10             10          62.0
## 11             11          61.4
## 12             12          67.1
```

Now let's inspect the movies score by genre and critics score average.

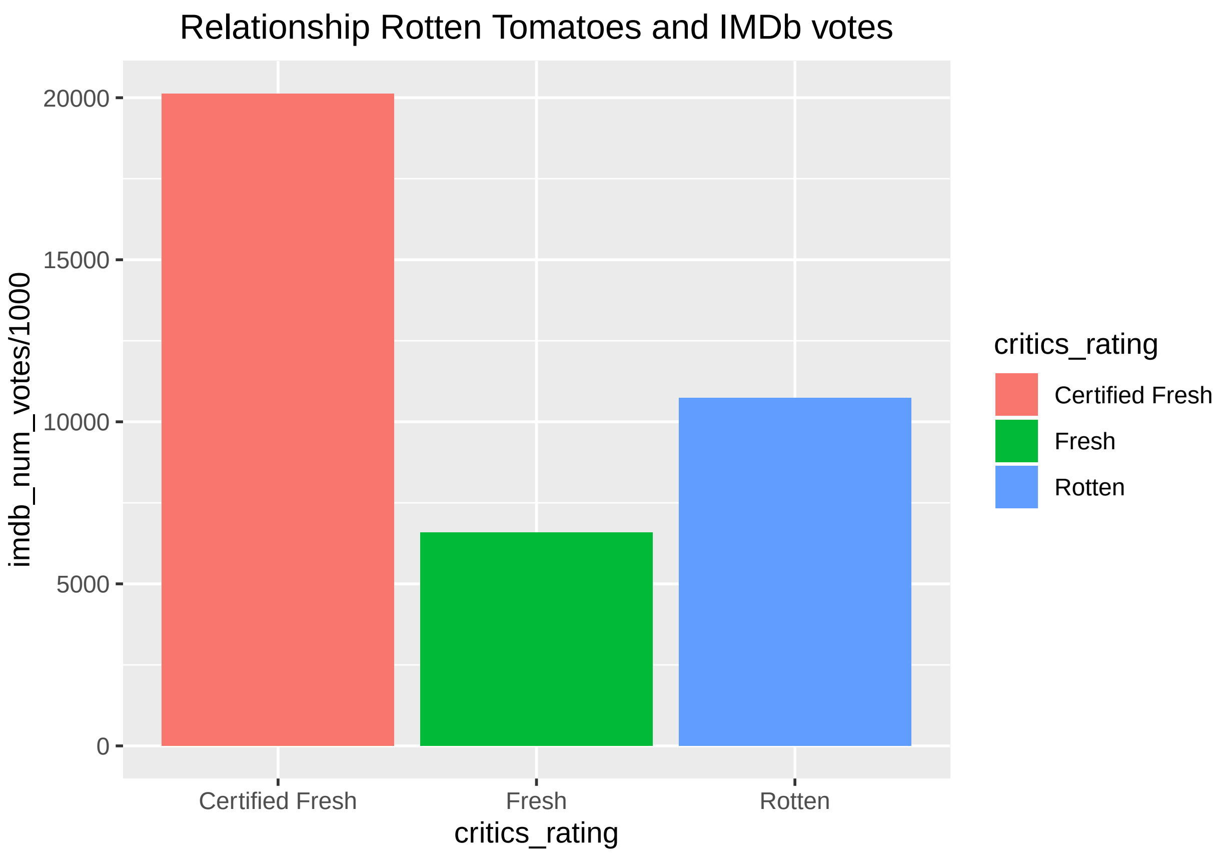
```
na.omit(movies) %>% ggplot(aes(x = genre, y = critics_score)) + geom_boxplot() + theme(axis.text.x = el
```



These box plots shows the average scores in relation with their genre, anything above 50 is good critic. We can observe the average and the data where is placed.

Now let's find out if the number of votes in IMDb relates to the critics rating.

```r
ggplot(data = movies, aes(x = critics_rating, y = imdb_num_votes/1000, fill = critics_rating ) ) + geom_
```



Certified Fresh has the highst amount of votes, seems they do tend to be same in the different websites.

Part 4: Modeling

Now, let's build a linear model to predict a critics_score. Let's use the following variables.

- genre
- critics_rating
- thtr_rel_month
- runtime
- imdb_num_votes

And i'll choose to follow the backwards methodology on picking the most powerful and simplest one by dropping variables in relation with their P value.

```
movies_model <-  lm(critics_score ~  genre + critics_rating +  thtr_rel_month + runtime + imdb_num_vote
summary(movies_model)
```

```
##
## Call:
## lm(formula = critics_score ~ genre + critics_rating + thtr_rel_month +
##     runtime + imdb_num_votes, data = movies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -29.2115  -9.9115   0.1008   9.1823  31.4827
```

```
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      7.180e+01  3.671e+00  19.559  < 2e-16 ***
## genreAnimation                   5.419e-01  4.646e+00   0.117  0.90718
## genreArt House & International -5.824e-01  3.849e+00  -0.151  0.87979
## genreComedy                     -3.898e-01  2.137e+00  -0.182  0.85532
## genreDocumentary                 1.477e+01  2.595e+00   5.692 1.92e-08 ***
## genreDrama                       4.860e+00  1.830e+00   2.656  0.00810 **
## genreHorror                      2.133e+00  3.164e+00   0.674  0.50043
## genreMusical & Performing Arts  1.160e+01  4.145e+00   2.799  0.00528 **
## genreMystery & Suspense          2.401e+00  2.358e+00   1.018  0.30891
## genreOther                       3.325e+00  3.652e+00   0.910  0.36293
## genreScience Fiction & Fantasy -4.689e+00  4.618e+00  -1.015  0.31032
## critics_ratingFresh             -9.152e+00  1.565e+00  -5.849 7.93e-09 ***
## critics_ratingRotten            -5.197e+01  1.543e+00 -33.681  < 2e-16 ***
## thtr_rel_month                  -5.673e-03  1.477e-01  -0.038  0.96937
## runtime                          8.860e-02  3.022e-02   2.932  0.00349 **
## imdb_num_votes                   1.465e-06  5.457e-06   0.268  0.78847
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.95 on 634 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.7971, Adjusted R-squared:  0.7923
## F-statistic: 166.1 on 15 and 634 DF,  p-value: < 2.2e-16
```

Let's drop the variables with the highest P-value and start moving backwards, keep in mind we are looking in reducing the R-squared coefficient, which is the measure on how well the model does.

In this the variable thtr_rel_month has the hightest P-value. The model, just score a R-squared of 0.7923, let's drop this thtr_rel_month variable to see if we increase the R-squared value.

```
movies_model_2 <- lm(critics_score ~  genre + critics_rating  + runtime + imdb_num_votes, data = movies
summary(movies_model_2)
```

```
##
## Call:
## lm(formula = critics_score ~ genre + critics_rating + runtime +
##     imdb_num_votes, data = movies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -29.2301  -9.9050   0.1073   9.1778  31.4879
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      7.178e+01  3.651e+00  19.663  < 2e-16 ***
## genreAnimation                   5.344e-01  4.638e+00   0.115  0.90831
## genreArt House & International -5.824e-01  3.846e+00  -0.151  0.87968
## genreComedy                     -3.930e-01  2.134e+00  -0.184  0.85392
## genreDocumentary                 1.477e+01  2.593e+00   5.697 1.87e-08 ***
## genreDrama                       4.861e+00  1.828e+00   2.659  0.00804 **
## genreHorror                      2.131e+00  3.161e+00   0.674  0.50040
## genreMusical & Performing Arts  1.160e+01  4.141e+00   2.801  0.00525 **
## genreMystery & Suspense          2.405e+00  2.354e+00   1.022  0.30733
## genreOther                      3.331e+00  3.646e+00   0.914  0.36130
## genreScience Fiction & Fantasy -4.688e+00  4.614e+00  -1.016  0.31001
## critics_ratingFresh            -9.150e+00  1.563e+00  -5.855 7.64e-09 ***
## critics_ratingRotten           -5.196e+01  1.541e+00 -33.712  < 2e-16 ***
## runtime                         8.836e-02  2.954e-02   2.992  0.00288 **
## imdb_num_votes                  1.460e-06  5.451e-06   0.268  0.78891
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.94 on 635 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.7971, Adjusted R-squared:  0.7926
## F-statistic: 178.2 on 14 and 635 DF,  p-value: < 2.2e-16
```

The R-squared Value just incread a few points, which is good, as long the R-squared model keeps increasing it is fine to move foward and drop variables, next let's keep dropping variables, but before, you may be asking youself why don't we drop genre variable, it is obvious its P-Value it is not significant, but the reason we are not dropping the genre variable is beacause it is a variable with many levels we need to handle this one as a whole individual and because at least 1 level of this variable has a significant P-value we cannot drop it. Therefore our next move will be dropping the IMDb number of votes variable and let's see if the R-squared valeu changes.

```r
movies_model_3 <- lm(critics_score ~  genre + critics_rating  + runtime, data = movies)
summary(movies_model_3)
```

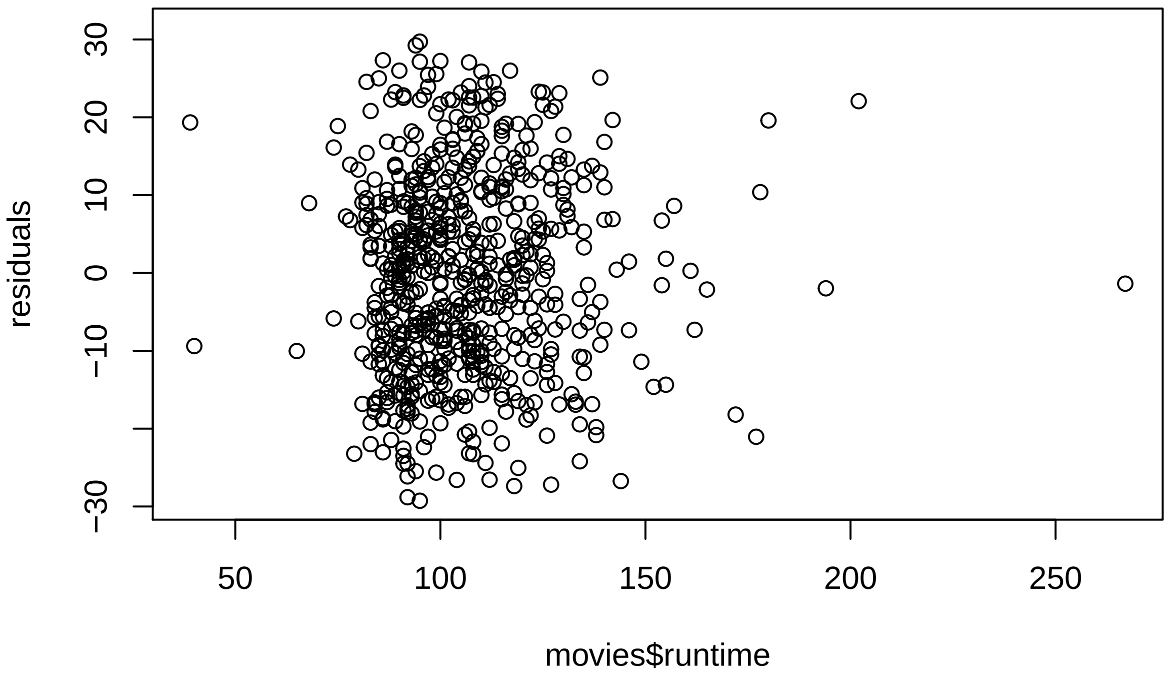
```
##
## Call:
## lm(formula = critics_score ~ genre + critics_rating + runtime,
##     data = movies)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -29.2620  -9.8143   0.0392   9.1757  31.5319
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    71.79167    3.64802  19.680  < 2e-16 ***
## genreAnimation                  0.50677    4.63396   0.109  0.91295
## genreArt House & International  -0.69847    3.81879  -0.183  0.85493
## genreComedy                    -0.42607    2.12839  -0.200  0.84140
## genreDocumentary               14.64133    2.54522   5.752 1.37e-08 ***
## genreDrama                      4.79584    1.81083   2.648  0.00829 **
## genreHorror                     2.08864    3.15461   0.662  0.50815
## genreMusical & Performing Arts  11.45677    4.10356   2.792  0.00540 **
## genreMystery & Suspense         2.38941    2.35117   1.016  0.30989
## genreOther                      3.35020    3.64263   0.920  0.35807
## genreScience Fiction & Fantasy -4.69053    4.61065  -1.017  0.30939
## critics_ratingFresh            -9.31032    1.44287  -6.453 2.18e-10 ***
## critics_ratingRotten          -52.13434    1.40472 -37.114  < 2e-16 ***
## runtime                         0.09082    0.02805   3.238  0.00127 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.93 on 636 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.7971, Adjusted R-squared:  0.7929
## F-statistic: 192.2 on 13 and 636 DF,  p-value: < 2.2e-16
```

The R-squared Value hast just increased a few points, that is good. Now we don't anymore variables to drop, so let's keep this model and test it.

Just as any model, let's make a diagnoistic of the model. First we will check the linearity condition among our model variables, notice the only numerical variable we have in the model is the runtime variable.

Random Scatter around 0

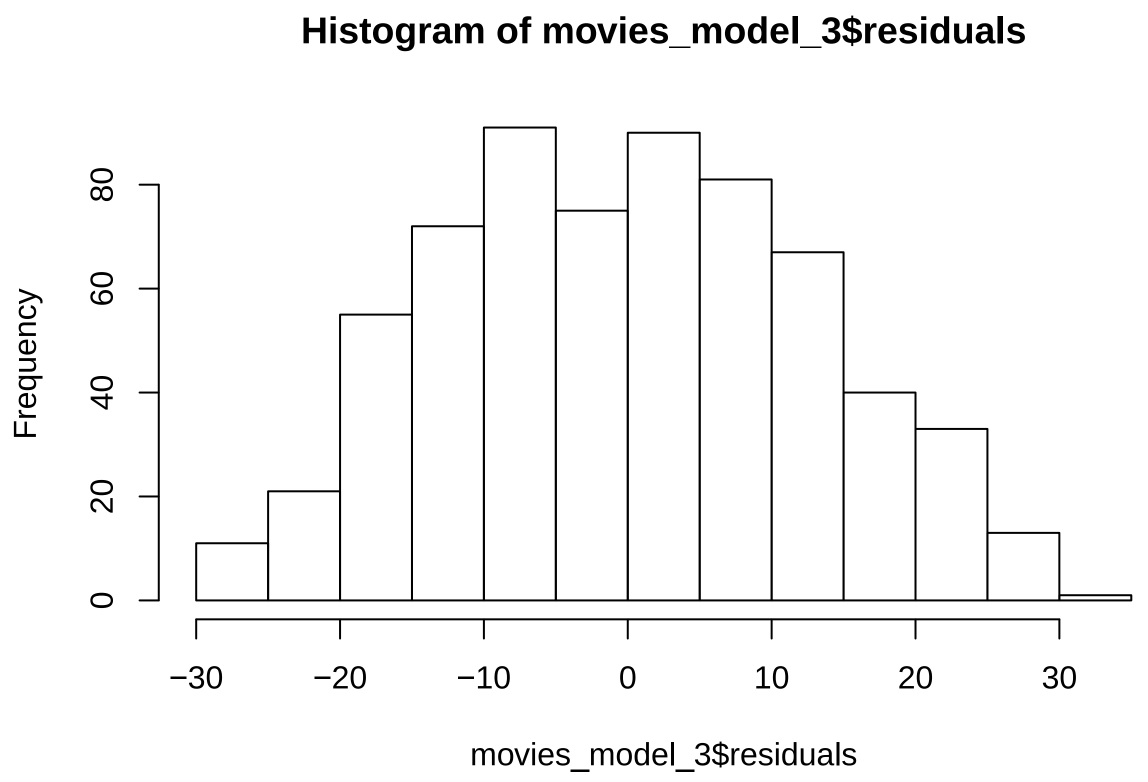
```
residuals <- c(movies_model_3$residuals, 0)
plot(residuals ~ movies$runtime)
```



We want our random scatter around 0. I think this is fairly ok.

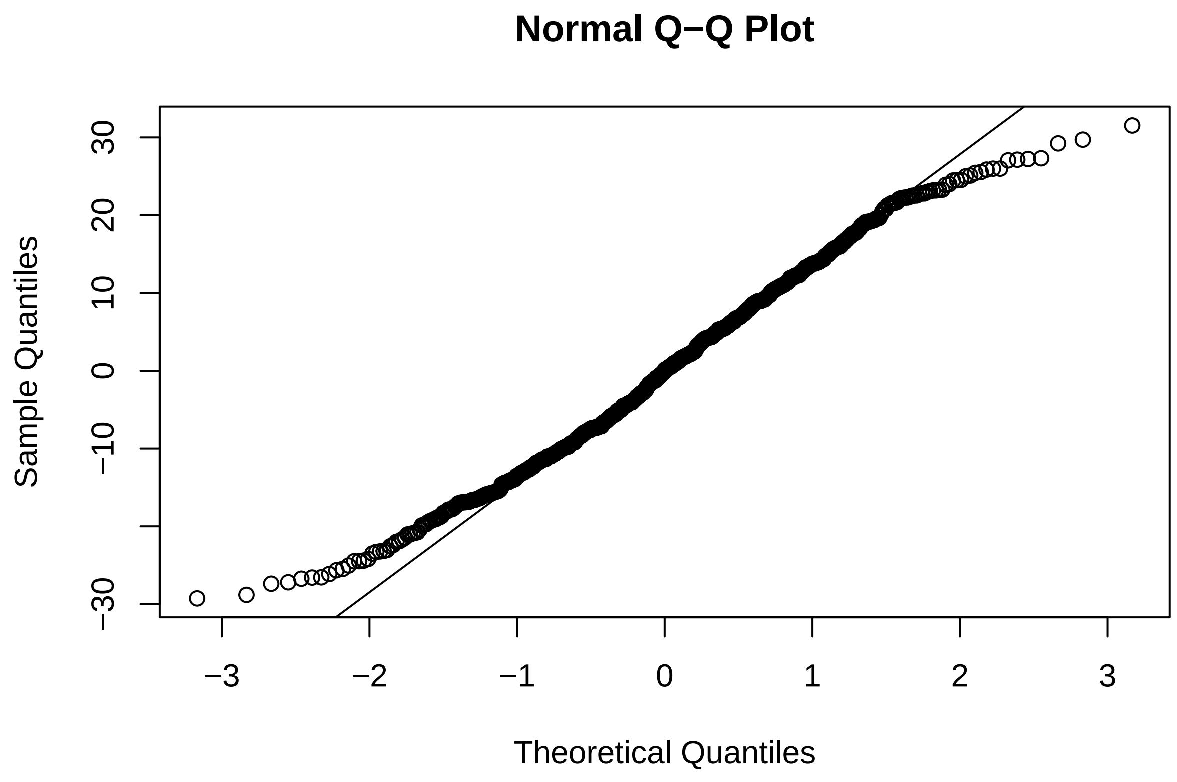
Nearly normal residuals with mean 0

```
hist(movies_model_3$residuals)
```



It seems it resembles a normal distribution. As well, let's build a normal probability plot.

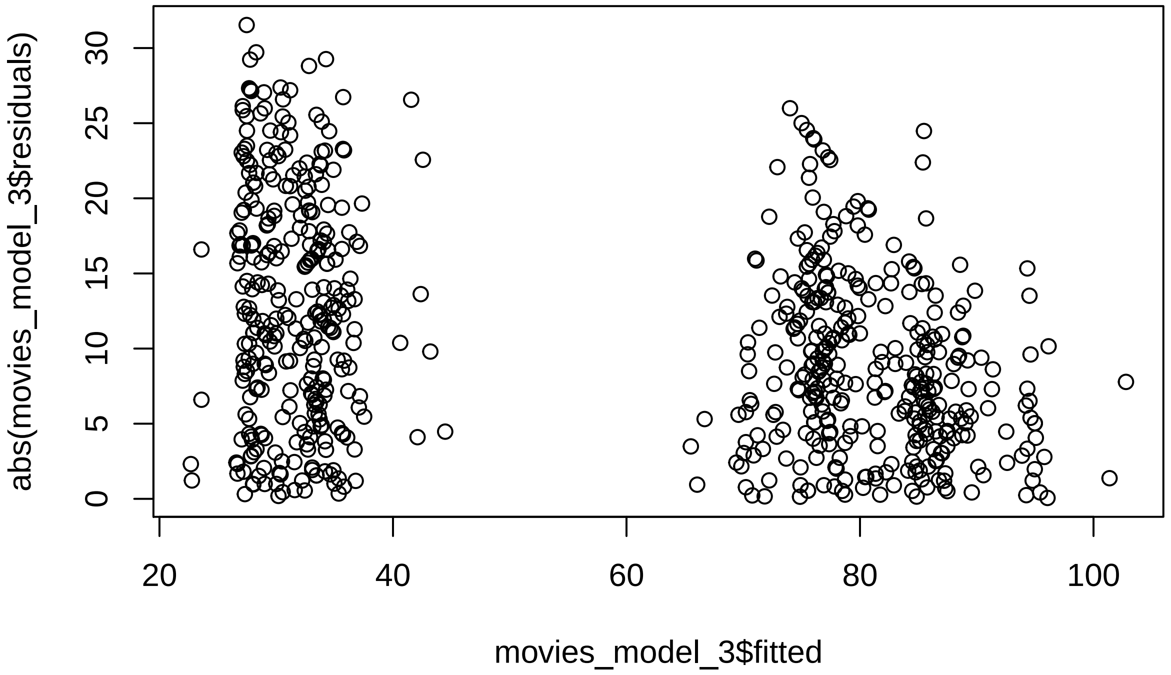
```r
qqnorm(movies_model_3$residuals)
qqline(movies_model_3$residuals)
```



It follows a little S shaped, however, there's not much variability but in the tails. This is fairly satisfied

Constant variability of residuals.

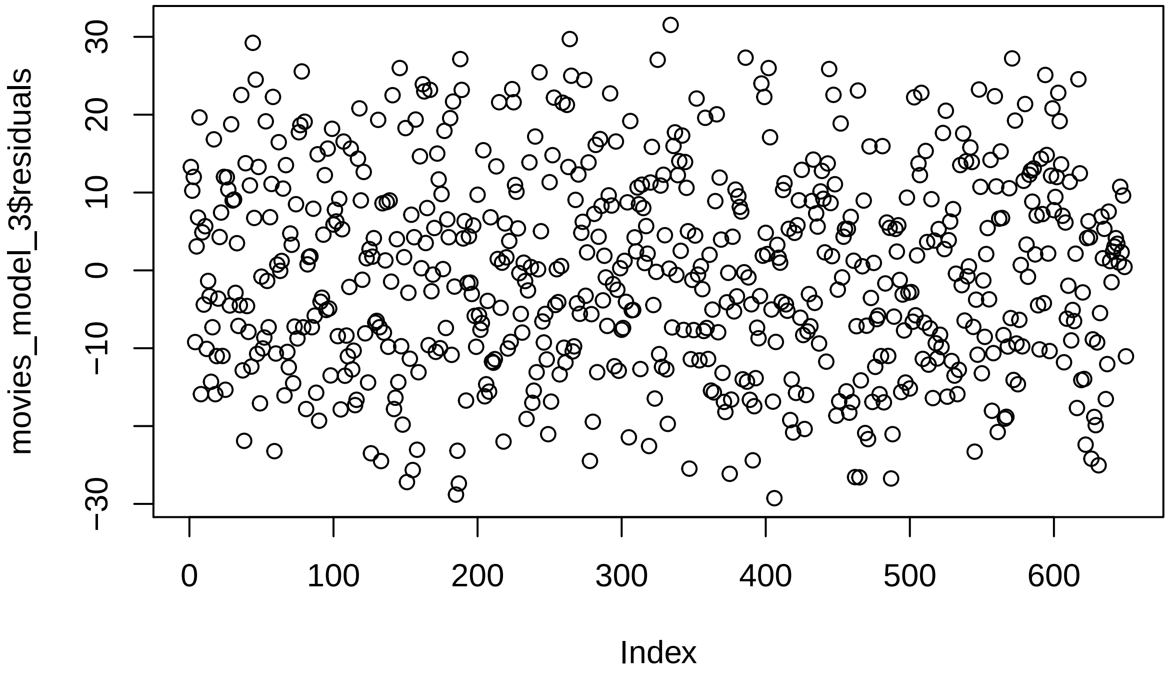
```r
plot(abs(movies_model_3$residuals) ~ movies_model_3$fitted)
```



This plot s

It shows any pattern in the residual plots. eems a little bit of strange, spite of that there's no fan shape or anything like, let's check if there's a pattern among the residuals.

```r
plot(movies_model_3$residuals)
```



* * *

Part 5: Prediction

let's prredict DATA!.

```r
y_test <- data.frame(genre = "Documentary", critics_rating = "Fresh", runtime = 105)
predict(movies_model_3, y_test, interlval="prediction", level = 0.05)
```

```
##        1
## 86.65909
```

With this we are predicting that a movie of genre Documentary, critics_rating Fresh and runtime 105, will score from 85 to 87 points interval, this means, according to Rotten Tomatoes a Certified Fresh label.

Part 6: Conclusion

Building a model to get a formula about what are the minimum characteristics these websites consider a good movie is not that easy, because spite of having the ratings, genre, actors and so on, there are other variables that comes to play such as personal taste, mood of the critic, personal issues againts actors or just simple, movie was great but critics dislike them.

This is educative purpouse only, do not tend to be a model for production or anything take it as step tutorial on building a multiple linear regression model.