Modeling and prediction for movies

March 2, 2018

Setup

This document serves the purpose of a final evaluation of the 4 week Linear Regression Model course by Duke University. The dataset contains information from Rotten Tomatoes, a website that keeps track of all reviews for each films and aggregates the results and Internet Movie Database IMDB, an online database of information related to film, television programs and video games. here (https://d3c33hcgiwev3.cloudfront.net/_e1fe0c85abec6f73c72d73926884eaca_movies.Rdata?Expires=1520035200&Signature=ee5Cfb-39QZowV7OycEEQY2SagxfuHflOdTQkYzs-OcWKVty8VbRtMp4QjuROorTxCZdl7M3O0P1qkkTea4-QuiC8eEbU~qdb7jXcX8XJ6CTNQwUtHC0HroLOeLZec846a4GVSxHGKSxtJIwXLltMvRYYE~APPl4tUl~WpuloMQ_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A):

Our purpose for this exercise is to develop a multiple linear regression model that will explain what makes movies popular given the variables in a dataset

Load packages

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

Load data

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

```
##  [1] "title"            "title_type"       "genre"
##  [4] "runtime"          "mpaa_rating"      "studio"
##  [7] "thtr_rel_year"    "thtr_rel_month"   "thtr_rel_day"
## [10] "dvd_rel_year"     "dvd_rel_month"    "dvd_rel_day"
## [13] "imdb_rating"      "imdb_num_votes"   "critics_rating"
## [16] "critics_score"    "audience_rating"  "audience_score"
## [19] "best_pic_nom"     "best_pic_win"     "best_actor_win"
## [22] "best_actress_win" "best_dir_win"     "top200_box"
## [25] "director"         "actor1"           "actor2"
## [28] "actor3"           "actor4"           "actor5"
## [31] "imdb_url"         "rt_url"
```

```
dim(movies)
```

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

Part 1: Data

Audience score is created by voulnteers, the dataset may suffer from voluntary response bias since people with strong responses are more likely to participate. The voting and rating are voluntary on IMDB and Rotten Tomatos website.

Our first task for this assignment is to choose which variables to include in our model.

I have decided not to include variables like name of director or actreses and actors, I would also not include the title of the movie, because it doesn't make sense to me for this particular anlysis to have a title as a potentioanl explanatory variable. Probably some spesific words in a title may influence the audience score..but this is not in the scope of this particular assignment. Year of release as well as month, day of release - I would not take into account as well as dvd release info. I would focus only on the genre, runtime, mpaa_rating, studio, imdb_rating, critics_rating, critics_score, audience_rating, audience_score, best_pic_nom, best_pic_win, best_actor_win, best_actress_win,best_dir_win. We'll make a smaller dataset containing only the variables of interest that would help us answer the research question.

```
# create a smaller dataset and remove NAs
df <- movies %>%
      select(genre, runtime,mpaa_rating,studio,imdb_rating,critics_rating,critics_score,
             audience_rating,audience_score,best_pic_nom,best_pic_win,best_actor_win,
             best_actress_win,best_dir_win) %>%
      na.omit()
# compare the initial dataset with the newly created one
dim(movies)
```

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

```
dim(df)
```

```
## [1] 642  14
```

Part 2: Research question

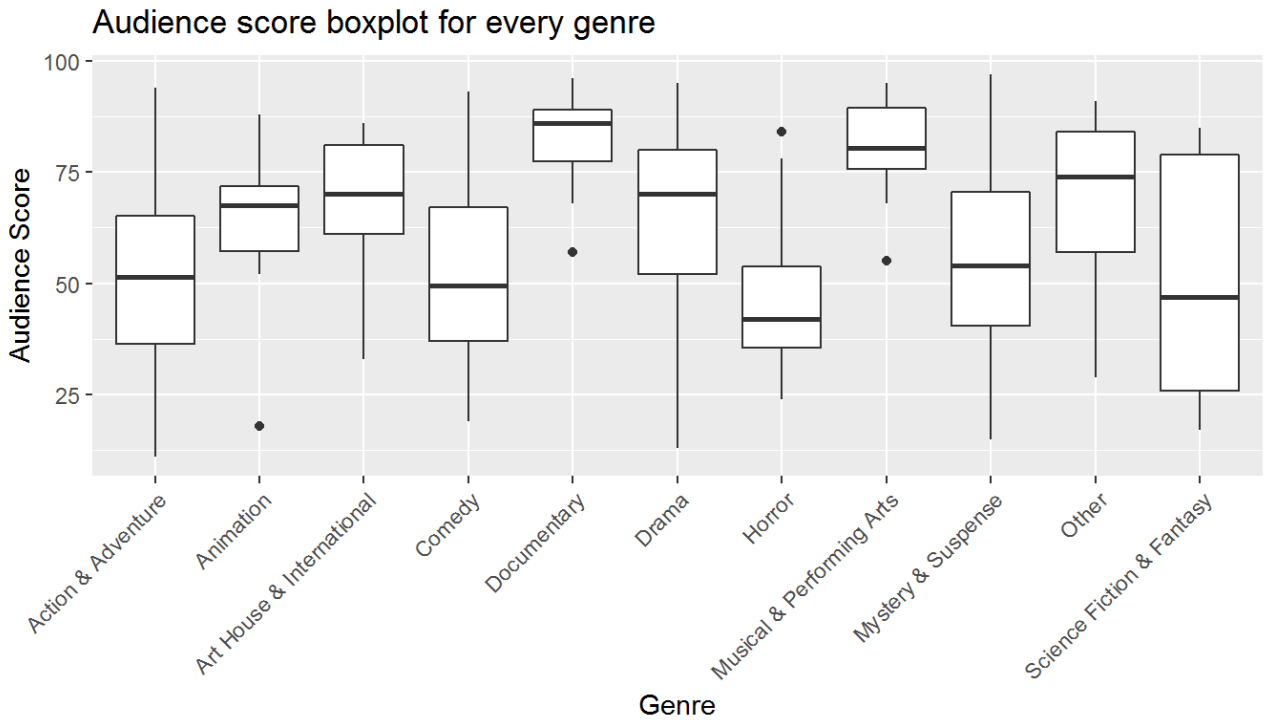
I am interested in learning what attributes make a movie popular. I would also like to learn something new about movies.

Part 3: Exploratory data analysis

```
# We'll explore our new dataset
summary(df)
```

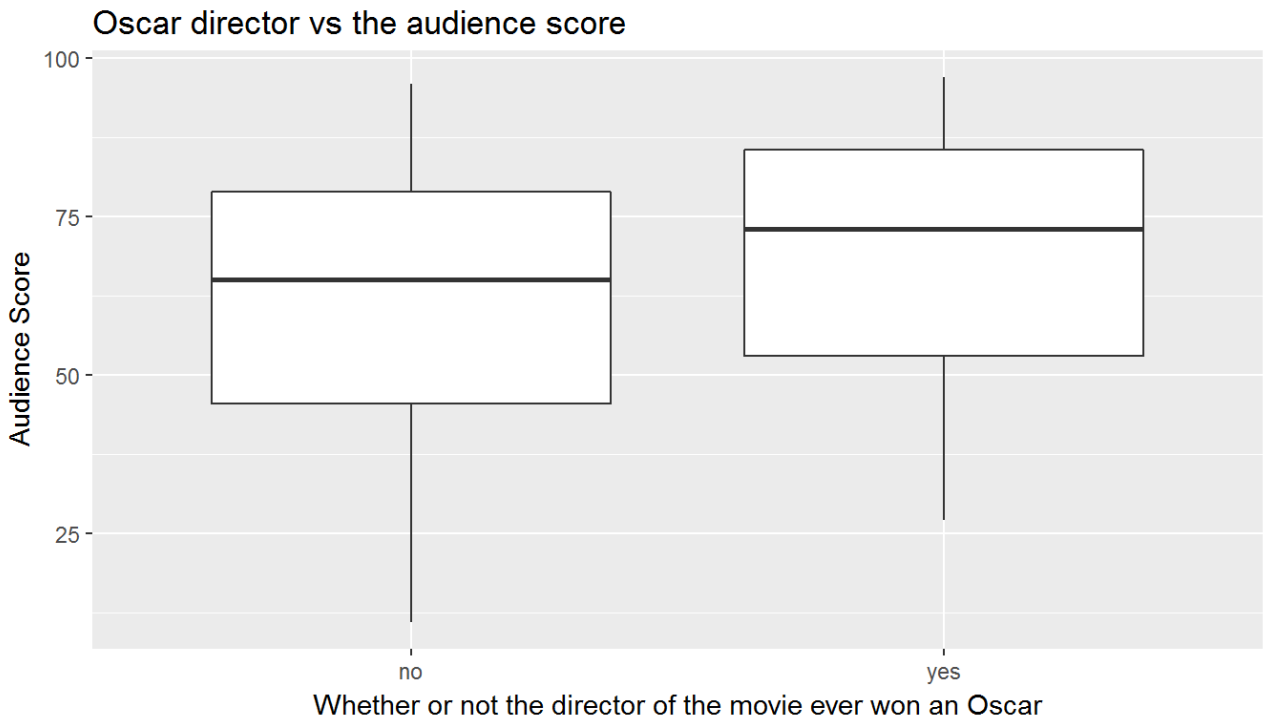
```
##                 genre        runtime     mpaa_rating
##  Drama             :303   Min.   : 39   G      : 18
##  Comedy            : 86   1st Qu.: 93   NC-17  :  1
##  Action & Adventure: 64   Median :103   PG     :117
##  Mystery & Suspense: 59   Mean   :106   PG-13  :133
##  Documentary       : 51   3rd Qu.:116   R      :324
##  Horror            : 22   Max.   :267   Unrated: 49
##  (Other)           : 57
##                                studio     imdb_rating
##  Paramount Pictures                : 37   Min.   :1.9
##  Warner Bros. Pictures             : 30   1st Qu.:5.9
##  Sony Pictures Home Entertainment: 27   Median :6.6
##  Universal Pictures                : 23   Mean   :6.5
##  Warner Home Video                 : 19   3rd Qu.:7.3
##  20th Century Fox                  : 18   Max.   :9.0
##  (Other)                           :488
##         critics_rating critics_score    audience_rating audience_score
##  Certified Fresh:135    Min.   :  1.00   Spilled:269     Min.   :11.0
##  Fresh          :205    1st Qu.: 33.00   Upright:373     1st Qu.:46.0
##  Rotten         :302    Median : 61.00                   Median :65.0
##                         Mean   : 57.78                   Mean   :62.5
##                         3rd Qu.: 83.00                   3rd Qu.:80.0
##                         Max.   :100.00                   Max.   :97.0
##
##  best_pic_nom best_pic_win best_actor_win best_actress_win best_dir_win
##  no :620      no :635      no :550        no :570          no :599
##  yes: 22      yes:  7      yes: 92        yes: 72          yes: 43
##
##
##
##
##
```

```
# We'll take a closer look at the audience_score distribution in every genre
ggplot(df, aes(x = factor(genre),y = audience_score)) +
      geom_boxplot() +
      theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
      labs(x = "Genre",y = "Audience Score",title = "Audience score boxplot for every genr
e")
```

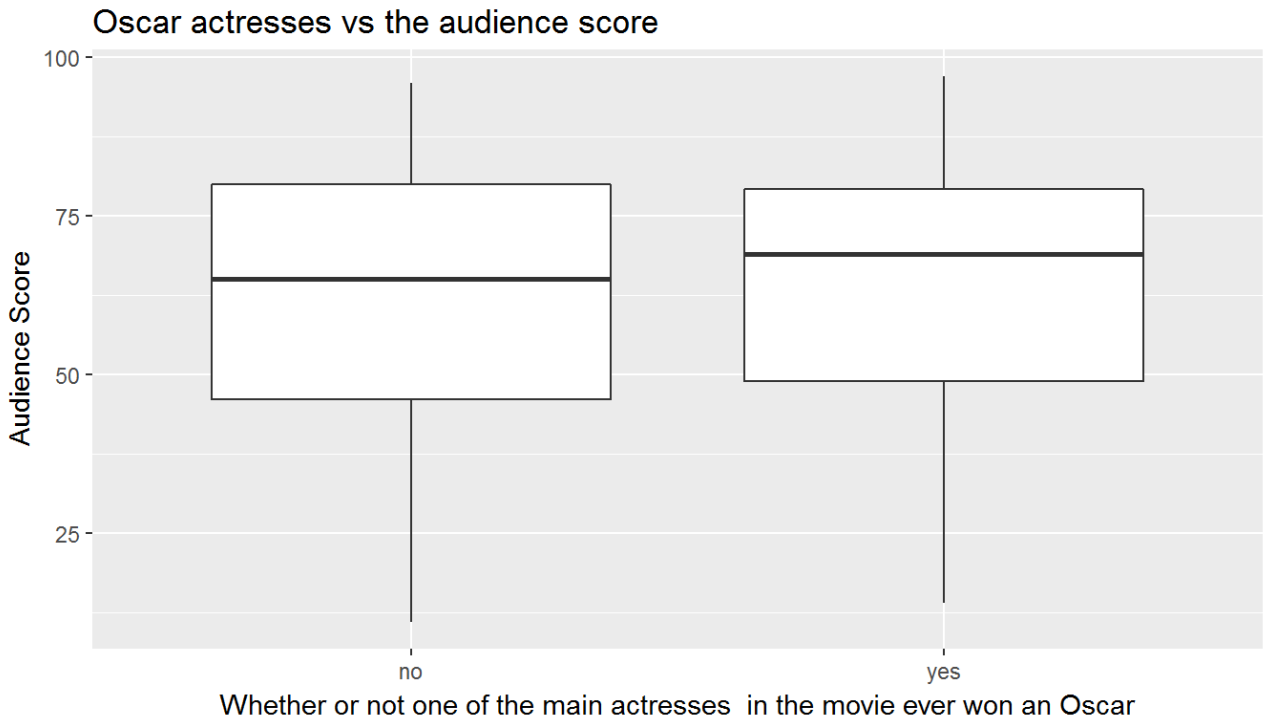


We see here that Documentary and Musicals tend to have higher scores on agerage than the other genres. Now I would like to see how the audience score is distributed among the variables of oscar winning.

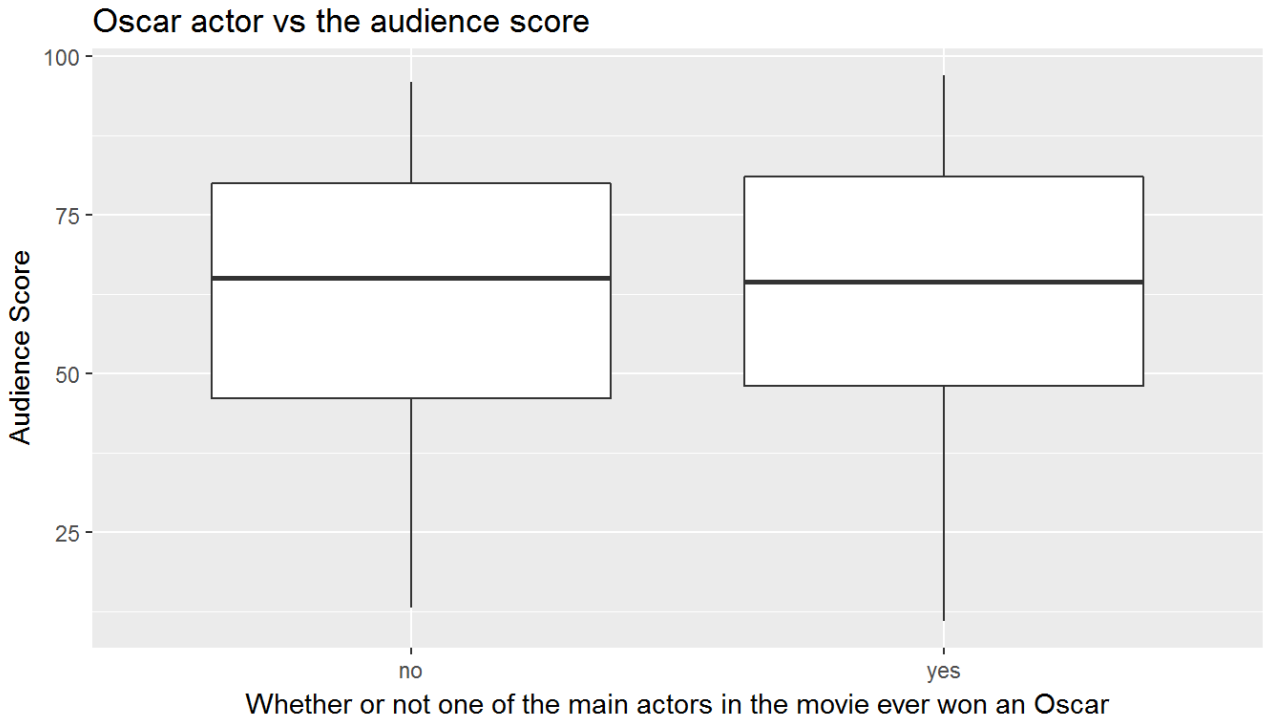
```
ggplot(df, aes(x = best_dir_win,y = audience_score)) +
      geom_boxplot() +
      labs(x = "Whether or not the director of the movie ever won an Oscar",y = "Audience
 Score",title = "Oscar director vs the audience score")
```



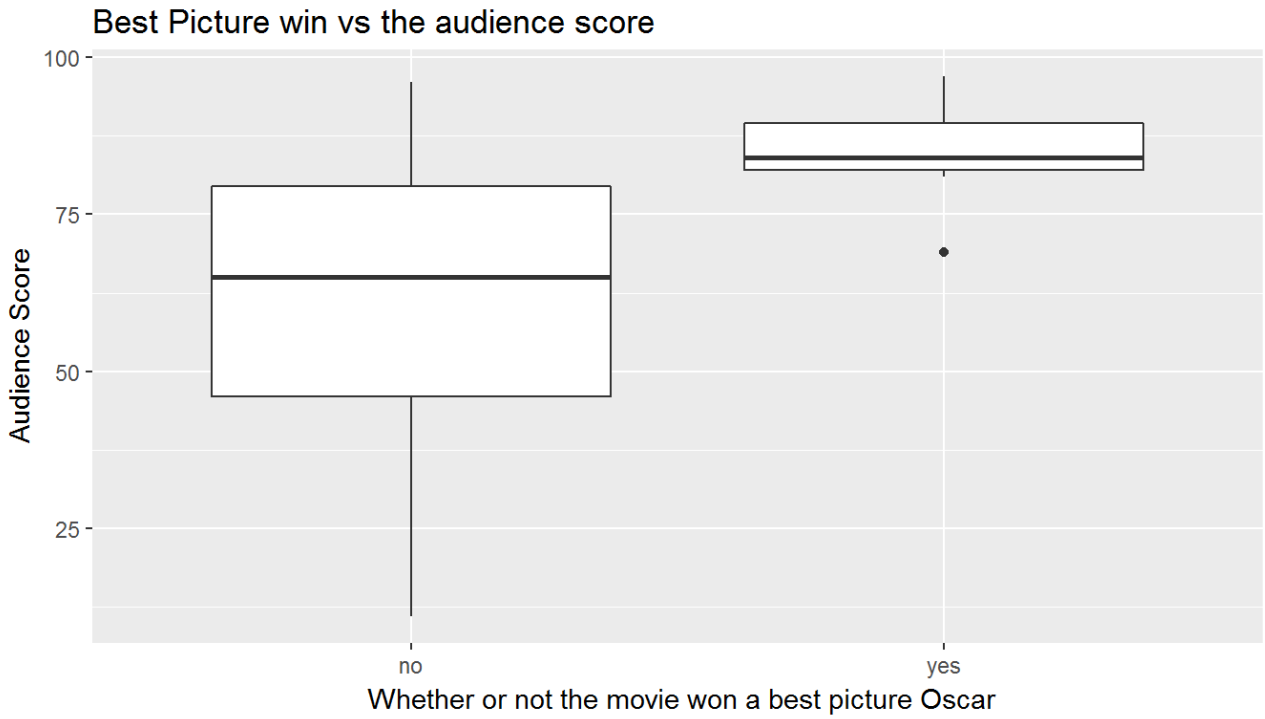

```
ggplot(df, aes(x = best_actress_win,y = audience_score)) +
      geom_boxplot() +
      labs(x = "Whether or not one of the main actresses  in the movie ever won an Oscar",
y = "Audience Score",title = "Oscar actresses vs the audience score")
```



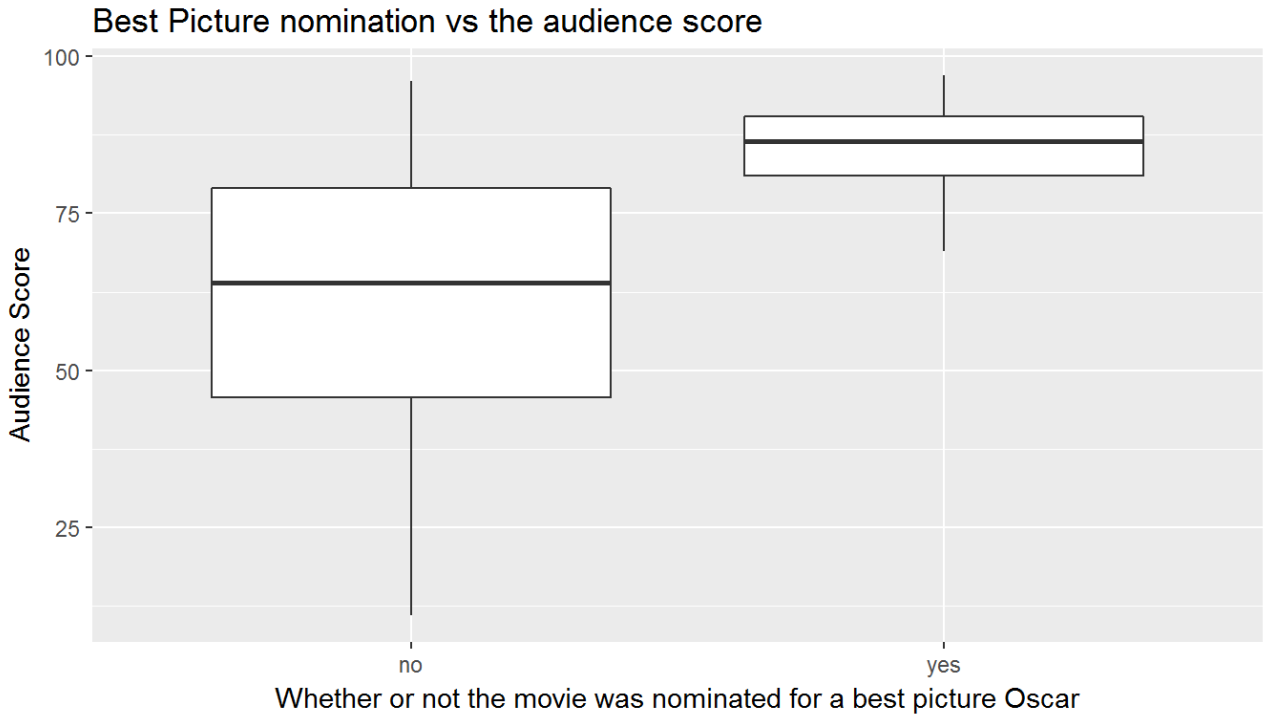

```
ggplot(df, aes(x = best_actor_win,y = audience_score)) +
      geom_boxplot() +
      labs(x = "Whether or not one of the main actors in the movie ever won an Oscar",y =
"Audience Score",title = "Oscar actor vs the audience score")
```




```
ggplot(df, aes(x = best_pic_win,y = audience_score)) +
      geom_boxplot() +
      labs(x = " Whether or not the movie won a best picture Oscar",y = "Audience Score",t
itle = "Best Picture win vs the audience score")
```




```
ggplot(df, aes(x = best_pic_nom,y = audience_score)) +
      geom_boxplot() +
      labs(x = "Whether or not the movie was nominated for a best picture Oscar",y = "Audi
ence Score",title = "Best Picture nomination vs the audience score")
```



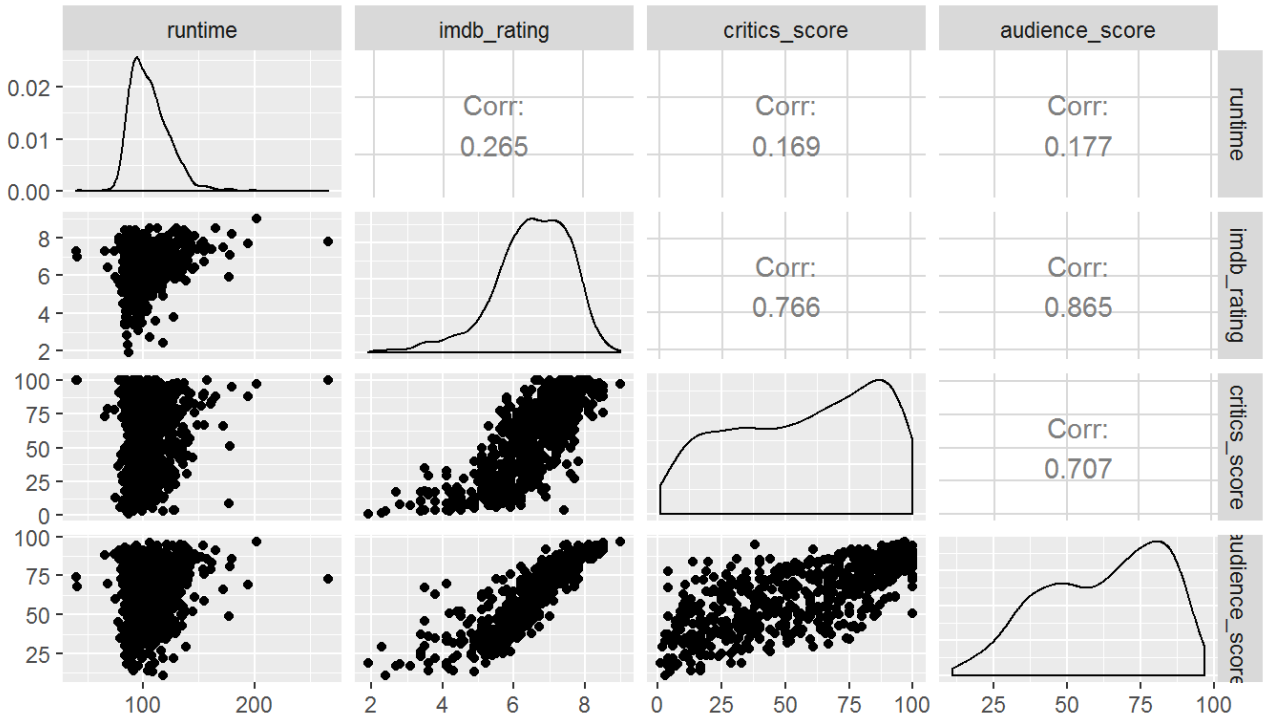
Directors who had won an onscar in their carrier produce a little bit higher graded movie. We see normally distributed audience score for a movie among actors and actresses who won an oscar. Best picture win and nomination have always more than the average audience score. Which of course shouldn't be suprising :) * * *

Part 4: Modeling

We'll create a Multiple Linear Regression model that predicts audience score and Backward elimination will help us define if better results can be obtained by using a smaller set of attributes. I'll use this approach because it evaluated both the significance and the proportion of variability as measured by adjusted R-square.

We'll take a look at a correlation matrix of the numerical variables, but first, let's make a smaller dataset consisting of numerical variables.

```
# make a smaller dataset containing only the numerical variables
small = df %>% select(runtime, imdb_rating,critics_score, audience_score)
# make the correlation matrix
ggpairs(small)
```



We see here that the correlation of critics_score and imdb_rating is 0.77. In order to avoid collinearity I will remove one of the variables from the future model. Multicollinearity exists whenever an independent variable is highly correlated with one or more of the other independent variables in a multiple regression equation. Multicollinearity is a problem because it undermines the statistical significance of an independent variable.

Here we start to build our model and compare Adj.R.Squared and of course we want as little as possible predictors that would yield the highest adj.r.squared

```
model1 <- lm(audience_score ~ genre + runtime + mpaa_rating + imdb_rating + critics_rating
+ audience_rating + best_pic_nom + best_pic_win + best_actor_win + best_actress_win + best
_dir_win, data = df)
summary(model1)$adj.r.squared
```

```
## [1] 0.8859603
```

```
summary(model1)
```

```
##
## Call:
## lm(formula = audience_score ~ genre + runtime + mpaa_rating +
##     imdb_rating + critics_rating + audience_rating + best_pic_nom +
##     best_pic_win + best_actor_win + best_actress_win + best_dir_win,
##     data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.3245  -4.3698   0.4382   4.2932  24.7648
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -6.72928    3.49783  -1.924   0.0548 .
## genreAnimation                  4.26946    2.78263   1.534   0.1255
## genreArt House & International -1.73159    2.14397  -0.808   0.4196
## genreComedy                     1.63617    1.15171   1.421   0.1559
## genreDocumentary                0.41288    1.57907   0.261   0.7938
## genreDrama                     -0.34602    1.00061  -0.346   0.7296
## genreHorror                    -1.44185    1.74016  -0.829   0.4077
## genreMusical & Performing Arts  3.02376    2.20404   1.372   0.1706
## genreMystery & Suspense        -2.57003    1.29602  -1.983   0.0478 *
## genreOther                      0.37753    2.00538   0.188   0.8507
## genreScience Fiction & Fantasy -0.03583    2.44963  -0.015   0.9883
## runtime                        -0.02529    0.01660  -1.524   0.1281
## mpaa_ratingNC-17               -6.92212    7.10893  -0.974   0.3306
## mpaa_ratingPG                  -0.63209    1.89810  -0.333   0.7392
## mpaa_ratingPG-13               -1.29390    1.94474  -0.665   0.5061
## mpaa_ratingR                   -1.55935    1.87672  -0.831   0.4064
## mpaa_ratingUnrated             -0.98803    2.14706  -0.460   0.6456
## imdb_rating                     9.56397    0.41979  22.783   <2e-16 ***
## critics_ratingFresh            -0.21285    0.80598  -0.264   0.7918
## critics_ratingRotten           -1.10063    0.91833  -1.199   0.2312
## audience_ratingUpright         20.14841    0.79370  25.385   <2e-16 ***
## best_pic_nomyes                 3.94823    1.79388   2.201   0.0281 *
## best_pic_winyes                -2.04352    3.13128  -0.653   0.5142
## best_actor_winyes              -0.04671    0.81881  -0.057   0.9545
## best_actress_winyes            -1.29463    0.90285  -1.434   0.1521
## best_dir_winyes                 0.48128    1.18427   0.406   0.6846
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.831 on 616 degrees of freedom
## Multiple R-squared:  0.8904, Adjusted R-squared:  0.886
## F-statistic: 200.2 on 25 and 616 DF,  p-value: < 2.2e-16
```

```
anova(model1)
```

```
## Analysis of Variance Table
##
## Response: audience_score
##                   Df Sum Sq Mean Sq   F value    Pr(>F)
## genre             10  52386    5239  112.2627 < 2.2e-16 ***
## runtime            1   5790    5790  124.0858 < 2.2e-16 ***
## mpaa_rating        5   5452    1090   23.3694 < 2.2e-16 ***
## imdb_rating        1 137644  137644 2949.7240 < 2.2e-16 ***
## critics_rating     2   1564     782   16.7547 8.213e-08 ***
## audience_rating    1  30402   30402  651.5258 < 2.2e-16 ***
## best_pic_nom       1    182     182    3.9027   0.04866 *
## best_pic_win       1     18      18    0.3915   0.53175
## best_actor_win     1      1       1    0.0147   0.90339
## best_actress_win   1     96      96    2.0541   0.15231
## best_dir_win       1      8       8    0.1652   0.68460
## Residuals        616  28745      47
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
model2 <- lm(audience_score ~ genre + runtime +  mpaa_rating + imdb_rating + critics_ratin
g + audience_rating + best_pic_nom + best_pic_win , data = df)
summary(model2)$adj.r.squared
```

```
## [1] 0.8861014
```

```
model3 <- lm(audience_score ~ genre + runtime +  mpaa_rating + imdb_rating + critics_ratin
g + audience_rating + best_pic_nom  , data = df)
summary(model3)$adj.r.squared
```

```
## [1] 0.8862131
```

```
model4 <- lm(audience_score ~ genre + runtime + imdb_rating + critics_rating + audience_ra
ting + best_pic_nom  , data = df)
summary(model4)$adj.r.squared
```

```
## [1] 0.8867118
```

```
model5 <- lm(audience_score ~ genre + runtime + imdb_rating + audience_rating + best_pic_n
om  , data = df)
summary(model5)$adj.r.squared
```

```
## [1] 0.8867163
```

```
summary(model5)
```

```
##
## Call:
## lm(formula = audience_score ~ genre + runtime + imdb_rating +
##     audience_rating + best_pic_nom, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.4975  -4.5551   0.6266   4.2395  24.9400
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -9.53668    2.51902  -3.786 0.000168 ***
## genreAnimation                  5.10084    2.57195   1.983 0.047774 *
## genreArt House & International -2.04766    2.08641  -0.981 0.326759
## genreComedy                     1.38196    1.13079   1.222 0.222123
## genreDocumentary                0.56677    1.38585   0.409 0.682701
## genreDrama                     -0.67937    0.96033  -0.707 0.479556
## genreHorror                    -1.76130    1.69630  -1.038 0.299521
## genreMusical & Performing Arts  3.02804    2.17945   1.389 0.165217
## genreMystery & Suspense        -3.01416    1.24277  -2.425 0.015575 *
## genreOther                      0.53829    1.97484   0.273 0.785271
## genreScience Fiction & Fantasy  0.34594    2.42485   0.143 0.886601
## runtime                        -0.03030    0.01551  -1.954 0.051156 .
## imdb_rating                     9.79298    0.38193  25.641  < 2e-16 ***
## audience_ratingUpright         20.32671    0.77658  26.175  < 2e-16 ***
## best_pic_nomyes                 3.36579    1.56739   2.147 0.032145 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.808 on 627 degrees of freedom
## Multiple R-squared:  0.8892, Adjusted R-squared:  0.8867
## F-statistic: 359.4 on 14 and 627 DF,  p-value: < 2.2e-16
```

```
anova(model5)
```

```
## Analysis of Variance Table
##
## Response: audience_score
##                  Df Sum Sq Mean Sq   F value  Pr(>F)
## genre            10  52386    5239  113.0120 < 2e-16 ***
## runtime           1   5790    5790  124.9139 < 2e-16 ***
## imdb_rating       1 142910  142910 3082.9990 < 2e-16 ***
## audience_rating   1  31925   31925  688.7176 < 2e-16 ***
## best_pic_nom      1    214     214    4.6113 0.03214 *
## Residuals       627  29064      46
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We made our model better be increasing its adj.r.squared and reducing the number of variables. We started with 11 variables and adj.r.squared 0.8859603, now we have 5 variables and a slightly better adj.r.squared 0.8867163

Intercept - -10.82576 is interpretted as the predicted mean audience_score when genre + runtime + imdb_rating + audience_rating is 0.

coefficient for genreAnimation coefficient 5.20719 - The model predicts that Animation films get an audience score that is 5.20719 higher than Action & Adventure(our reference category) films on average after controlling for all other variables. There are total 11 genre categories in the dataset, the audience score can higher or

lower than Action & Adventure films depends on what genre is selected.

coefficient for imdb_rating - All else held constant, for every one unit increase in imdb_rating the model predicts a 9.89 increase in audience_score on average.

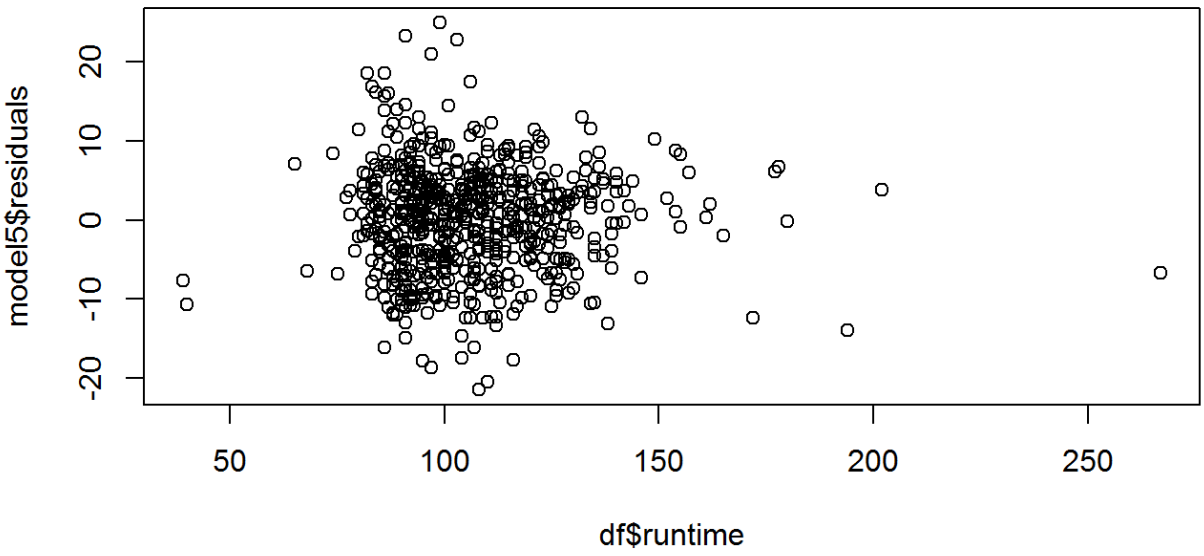
coefficient for audience_ratingUpright coefficient 20.37243: All else hold constant, the model predicts rating Upright movie is 20.3246 higher in audience score on average than rating Spilled movie.

R squared - 86.61 % of the variability in audience_score is explained by the model

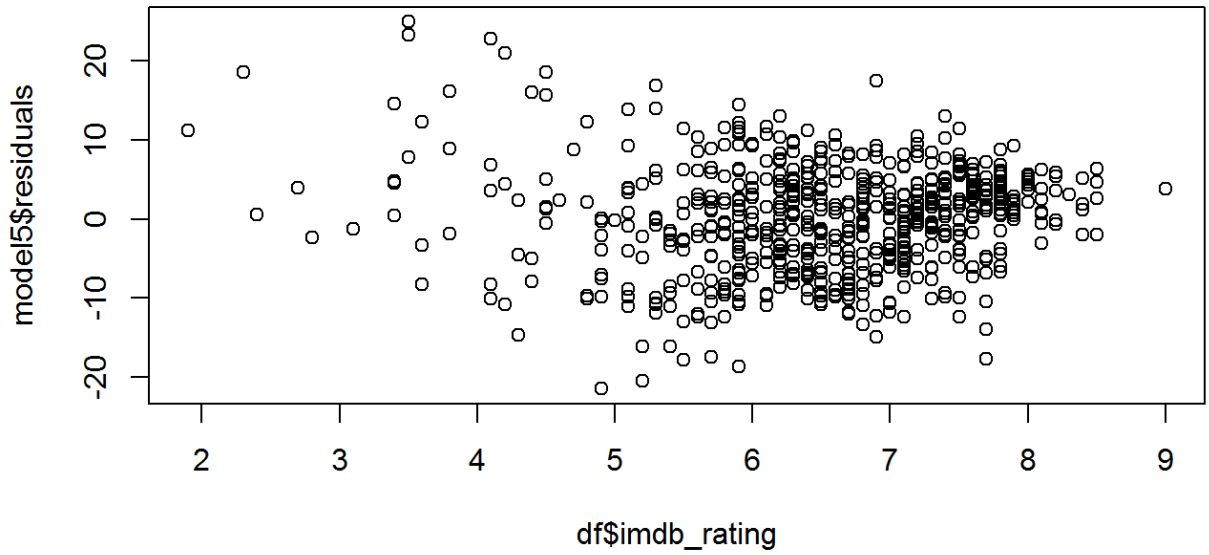
P-values: all coefficients in our model have a p-value that is less than 0.05.

Diagnostic for MLR: 1. linear relationships between x and y - each (numerical) explanatory variable needs to be linearly related to the response variable
2. nearly normal residuals - looking for random scatter around 0
3. constant variability of residuals - residuals should be equally variable for low and high values of the predicted response variable - checking using residuals plots of residuals vs. predicted (it allows for considering the entire model (with all explanatory variablles) at once)- residials randomly scattered in a band with a constat width around 0
4. independence of residuals - independent observations - if time series structure is suspected check using residuals vs order ot data collection

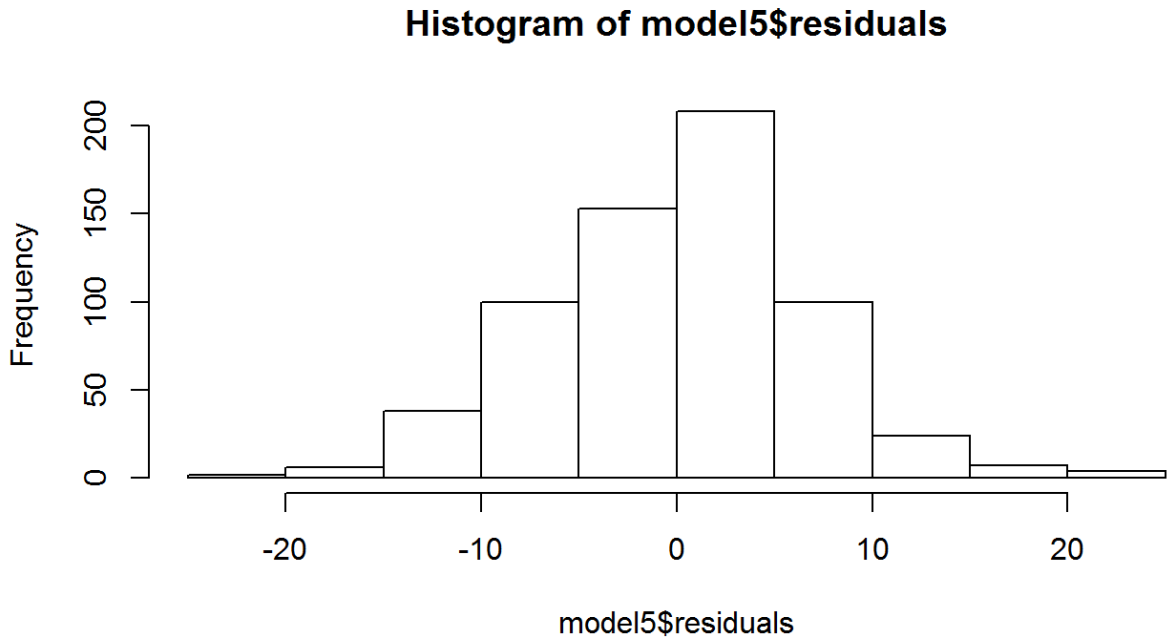
```
# 1
plot(model5$residuals ~ df$runtime)
```



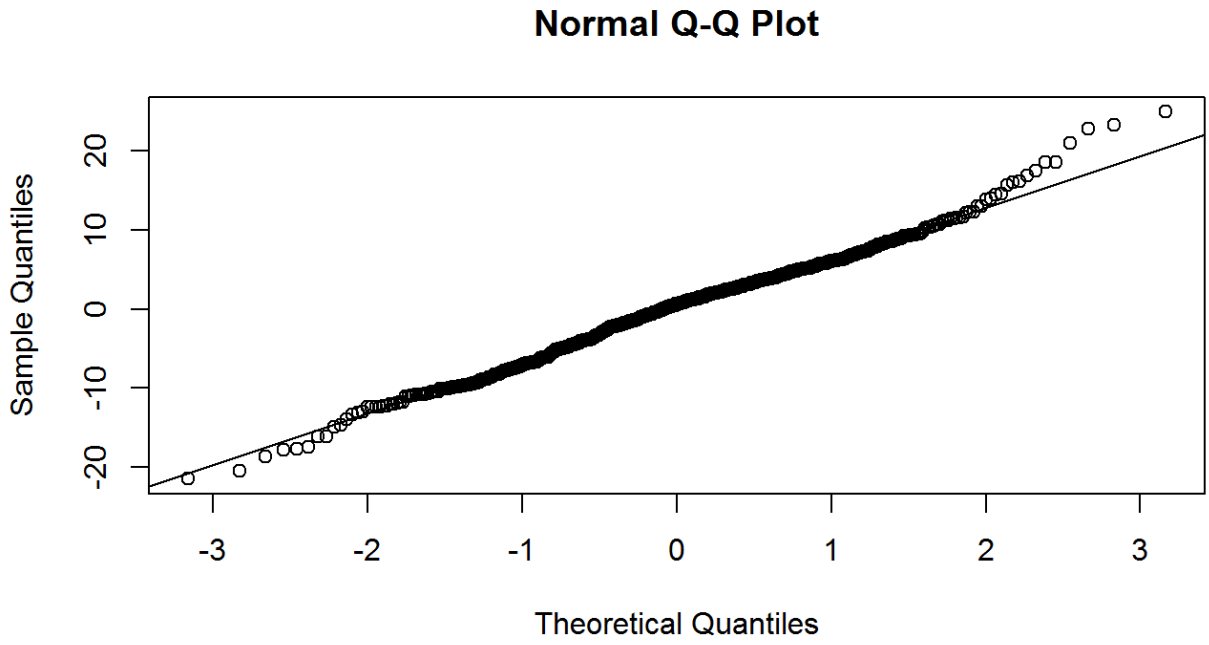

```
plot(model5$residuals ~ df$imdb_rating)
```



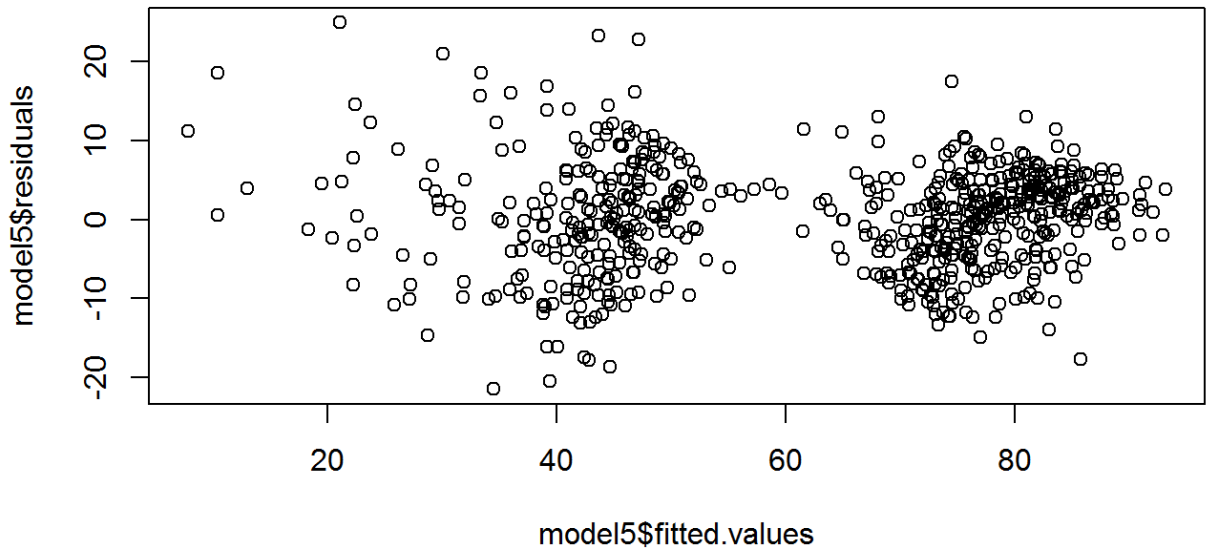

```
# 2
hist(model5$residuals)
```



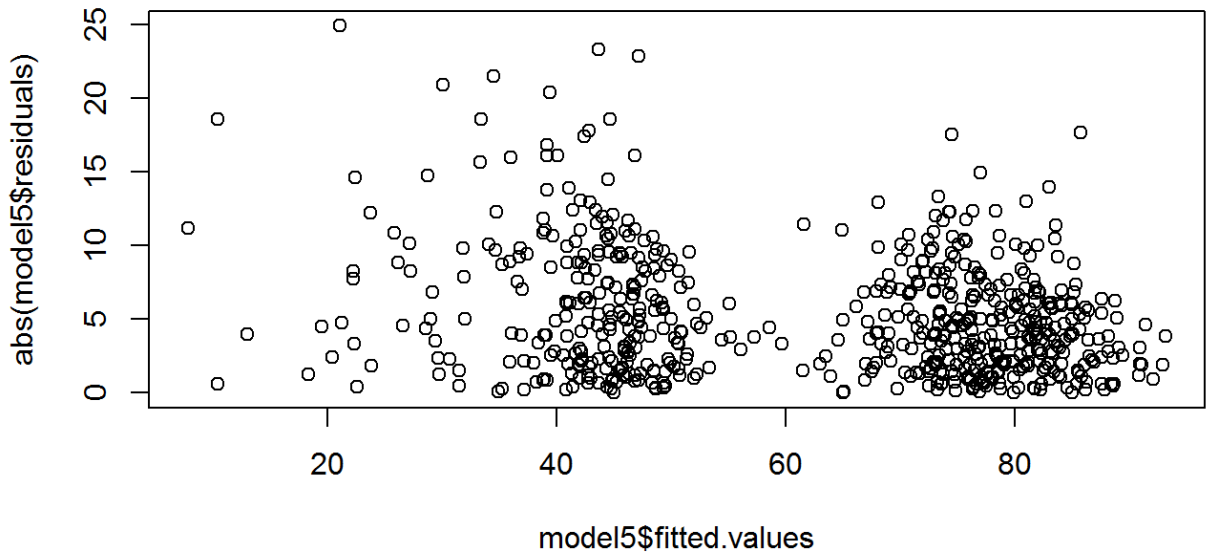

```
qqnorm(model5$residuals)
qqline(model5$residuals)
```



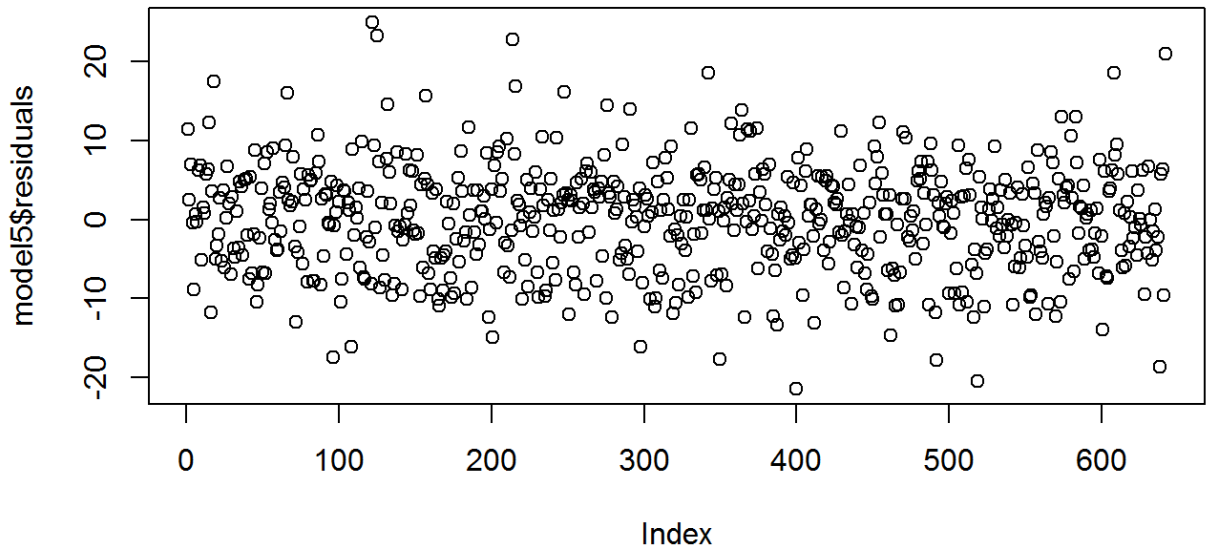

```r
# 3
plot(model5$residuals ~ model5$fitted.values)
```




```r
plot(abs(model5$residuals) ~ model5$fitted.values)
```




```
# 4
plot(model5$residuals)
```



The model dignastic plots suggest a strong linear relationship as demostrated by the residual plot which shows the residuals randomly scattered around 0 showing normal distribution of residuals centered at 0. This is also confirmed by the normal proabiity plots (points falling along the linear line). Residuals vs. predicted plots show random scatter and confirm constant variability of residuals.

Part 5: Prediction

Now I'll test the model with a movie from 2016 as required in the assignment. I chose the movie "Ghostbusters". The audience score in rottentomateos.com is 52% and now we'll see whether the model will predict it correctly.

```
newdata <- data.frame(genre = "Action & Adventure", runtime = 105, imdb_rating = 6.4, audi
ence_rating = "Spilled", best_pic_nom = "no" )

predict(model5,newdata, interval = "prediction", level = 0.95)
```

```
##        fit      lwr      upr
## 1 49.95722 36.45758 63.45686
```

Our model predicts, with 95% confidence, that the movie Ghostbusters is expected to have an audience score between 36.4 and 63.5. Yes, the prediction of 50 is quite close to the actual value of 52 and for certanly within the lower and upper boundry of our 0.95 confidence level.

Part 6: Conclusion

Using MLR we were able to identify a 5 variables that accurately were able to predict the audience score of a film from the year 2016 that was not included in the movies data set. shortcomings: - this model's predictive power is limited because the sample data is not representative.