Modeling and prediction for movies

Setup

Load packages

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

Part 1: Data

```
#load("movies.Rdata")
load("~/Documents/Stat_Duke_Univ/w3_Linear_Regression/projectw3/movies.Rdata")
```

The data set is comprised of 651 randomly sampled movies produced and released before 2016.

- **IMDB**
  - IMDB: "The Internet Movie Database (abbreviated IMDb) is an online database of information related to films, television programs and video games, including cast, production crew, fictional characters, biographies, plot summaries, trivia and reviews.
  - As of June 2016, IMDb has approximately 3.7 million titles (including episodes) and 7 million personalities in its database,[2] as well as 67 million registered users.
  - Users are invited to rate any film on a scale of 1 to 10, and the totals are converted into a weighted mean-rating that is displayed beside each title, with online filters employed to deter ballot-stuffing. Submitted ratings are filtered and weighted in various ways in order to produce a weighted mean that is displayed for each film"
- **Rotten Tomatoes**
  - It is a website launched in August 1998 devoted to film reviews and news; it is widely known as a film review aggregator. The name derives from the practice of audiences throwing rotten tomatoes when disapproving of a poor stage performance.
  - RT critic aggregate score: collect online reviews from writers who are certified members of various writing guilds or film critic associations. To be accepted as a critic on the website, a critic's original reviews must garner a specific number of "likes" from users. Those classified as "Top Critics" generally write for major newspapers. The staff determine for each review whether it is positive ("**fresh**", marked by a small icon of a red tomato) or negative ("**rotten**". Staff assessment is needed as some reviews are qualitative rather than numeric in ranking.
  - Audience Score and reviews: Each movie features a "user average," which calculates the percentage of users who have rated the film positively, similar to calculation of recognized critics' reviews. The users' score is more detailed, because users rate the movie on a scale of 0???10. (Critic reviews generally use 4-star ratings and are often qualitative). A user score of 7 (equivalent to 3.5 stars on a 5-star scale) or higher is considered positive. Registered and logged-in users can rate and review movies.
- Box Office Mojo: `top200_box` It is a website that tracks box office revenue in a systematic, algorithmic way, founded in 1999. In 2008, Box Office Mojo was bought by the Internet Movie Database, owned by Amazon. The website is widely used within the film industry as a source of data.

Part 2: Research question

1. **Q1: What can the data tell us. Particularities and possible relationships between variables.**
2. **Q2: Can we find some parameter that correlates with high ratings.**
3. **Q3: What attributes make a movie popular.**
4. **Q4: She is also interested in learning something new about movies.**
5. **Q5: We would like to buy films for "our" tv channel that the most audience would like. Can we find a model that gives the rating people will give the movie. It is high rating eneough for determining if the film is going to be popular?**

Part 3: Exploratory data analysis

Ratings

Summaries:

Number of Observation and of variables

```
# Number of observation and variables in the data file movie
dim(movies)
```

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

Max, Min, Median, Mean, QQ

```
# Rottten Tomatoes scores rescaled by 10
a3<-c("RT critics score", summary(movies$critics_score/10))
a1<-c("RT audience score", summary(movies$audience_score/10))
a2<-c("IMDB rating", summary(movies$imdb_rating))
a4<-c("IMDB Number of votes", summary(movies$imdb_num_votes))
summaries=data.frame(rbind(a1,a2,a3,a4)); summaries
```

```
##                        V1 Min. X1st.Qu. Median  Mean X3rd.Qu.   Max.
## a1    RT audience score  1.1      4.6    6.5 6.236        8    9.7
## a2          IMDB rating  1.9      5.9    6.6 6.493      7.3      9
## a3     RT critics score  0.1      3.3    6.1 5.769      8.3     10
## a4 IMDB Number of votes  180     4546  15120 57530    58300 893000
```

Conclusions:

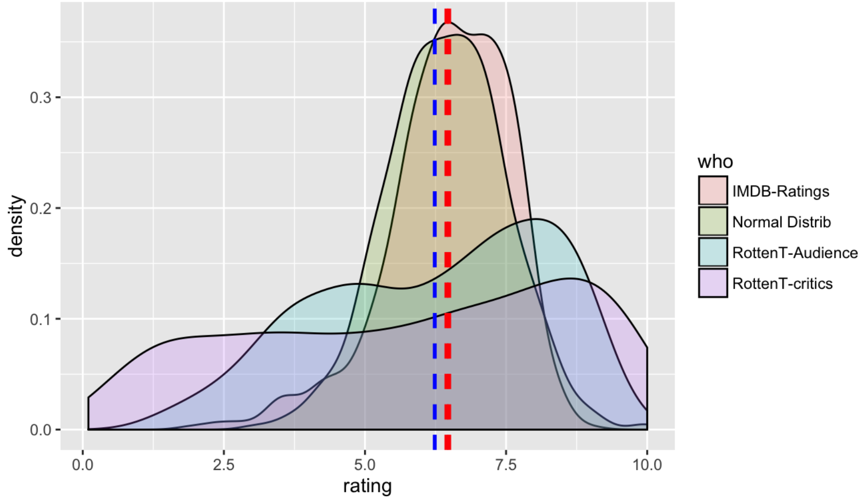
- The **RT audience score** and **IMDB rating** distributions are slightly **left skewed**. Both ratings have a mean close to 6.3 and a higher median around 6.5. The distribution have a queue on the left.
    - Data with an upper bound are often left skewed.

  It is possible that people uses to rate films at IMDB and RT when they consider that a film deserves a rate of at least over 3. However the lowest limit is 1. I suppose that when a film is bad people do not waste their time watching it or rating it. However people rate films when they think it is ok or good and they take their time rating them. Since the upper rating (10 points) is well defined the implication is the left skewness of the distribution.
- The **RT critics score** the median (6.1) is higher than the mean (5.8). Distribution is right skewed. Does the critics rate mostly the bad films?. Or do critics tend to give lower ratings than “common audience”.

Movies’ Rating Densities and Histograms

We check the ratings densities for IMDB-Ratings, RottenT-Audience, RottenT-critics. We compare these densities with a normal distribution with same mean and standard deviation than the IMDB-rating population.

```
# Data preparation for the density plot
movies<-movies %>%
    mutate(top_title=ifelse(imdb_num_votes>1000,title,NA),critics_score1=critics_score/10)
Who<-c(rep("IMDB-Ratings",651),rep("RottenT-Audience",651),rep("Normal Distrib",651), rep("RottenT-critics",651))
mu=mean(movies$imdb_rating); sig=sd(movies$imdb_rating)
normalIMDB=round(rnorm(651,mean=mu, sd=sig),1)
rating<-data.frame(c(movies$imdb_rating,movies$audience_score/10, normalIMDB, movies$critics_score/10))
critics=data.frame(cbind(Who,rating))
names(critics)=c("who","rating")
rating.mean=c(mean(movies$imdb_rating),mean(movies$audience_score/10), mean(normalIMDB), mean(movies$critics_scor
e/10))
# Red->IMDB-rating mean; Blue->RT Audience Mean
```

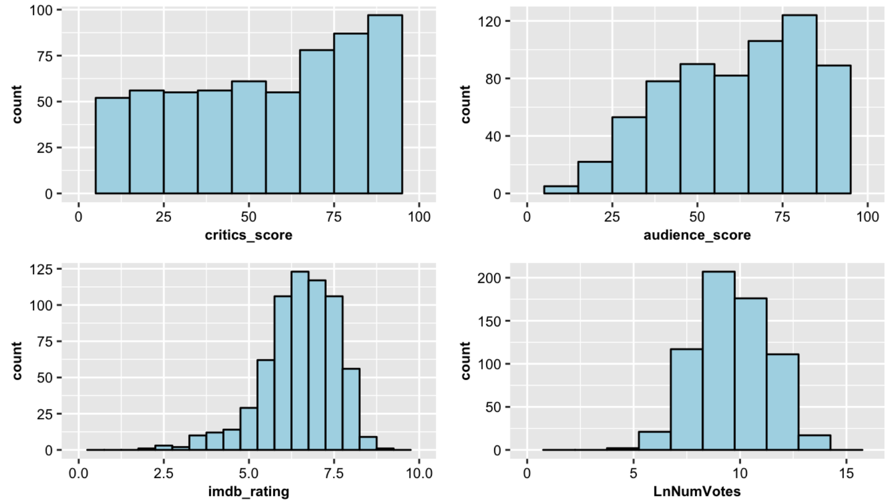


Conclusion:

- Left Skewed
- The **IMDB ratings** densities looks like a **normal distributed** population centered around 6.5.
- The **Rotten Tomatoes audience** ratings distributions shows two peaks one around 4 points and other around 8 points. The pick around 8 has higher density.
- The **Rotten Tomatoes critics** rating distribution looks completely different. The density seems to be approximately constant o slightly linear with highest density at 8.5.
    - We can see that RT critics’ watch and rates approximately as many “bad” films as “good” films. This is different to the RT audience and IMDB voters.

Histograms

Only the IMDB rating looks like a normal distribution. On the other hand Rotten Tomatoes critics and user gives more often high scores than lower. The distributions do not look normal at all.



Movies Rating vs. Genre & Nr. of Votes vs. Genre

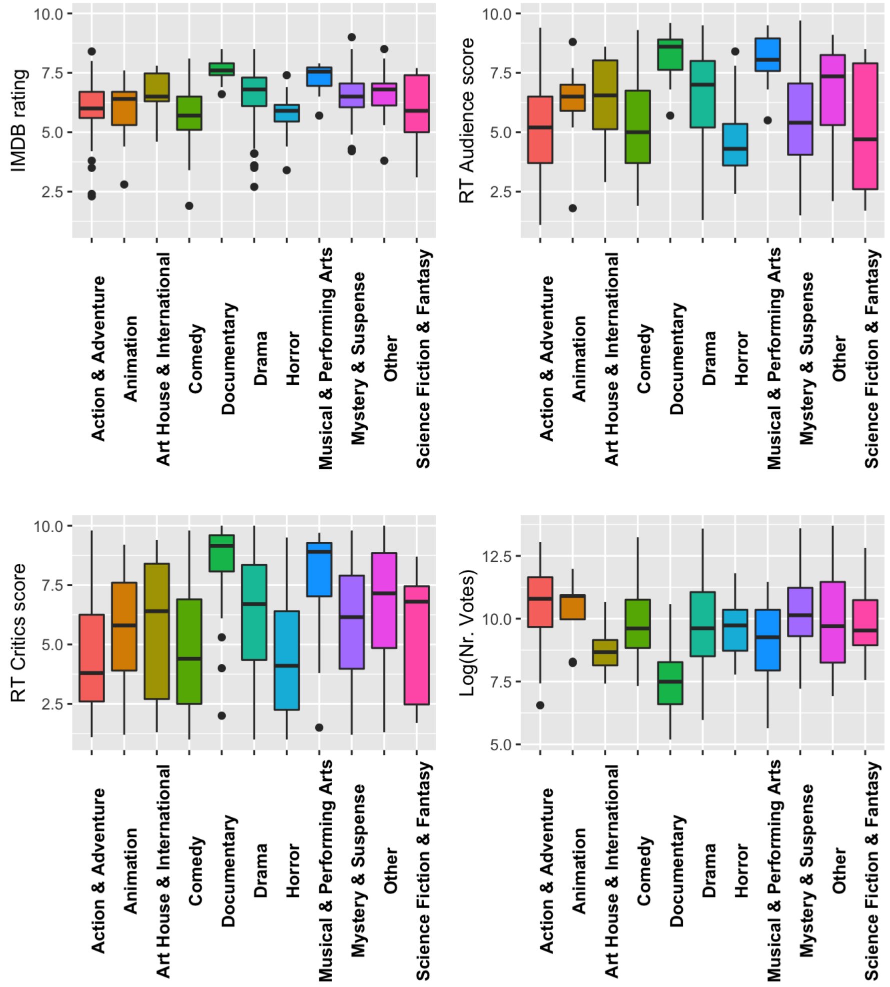
`genre` VS. `imbd_rating`, `audience_score`, `audience_score`

We know by experience that most audience do not watch the same amount of films of all kind of genres. For instance average people do not use to watch as many documentary as drams films.

Also for each genre there is not the same amount of films. For instance there are much less number of documentaries than of comedies. This means that the statistical analysis of this group could not be significant since the size of the sample is too small.

The first three box plots shows the relationships between the Movies Rating (RT Audience, IMDB, RT Critics) and the movies genre.

The 4th box plot shows the relationship between the **natural logarithm** of the IMDB **number of votes** and the movie genre.



```
genre.rating<-movies %>% group_by(genre) %>%
         summarise(IMDB.mean=round(mean(imdb_rating),1), Rt.mean.crit=round(mean(critics_score)/10,1),Rt.mean.aud=round(mean(audience_score)/10,1), LnNumV=round(mean(log(imdb_num_votes)),1))
genre.rating<-genre.rating[order(-genre.rating$IMDB.mean),]
genre.rating
```

```
## # A tibble: 11 x 5
##                        genre IMDB.mean Rt.mean.crit Rt.mean.aud LnNumV
##                       <fctr>     <dbl>        <dbl>       <dbl>  <dbl>
## 1                Documentary       7.6          8.6         8.3    7.6
## 2  Musical & Performing Arts       7.3          7.7         8.0    8.9
## 3                      Drama       6.7          6.2         6.5    9.8
## 4  Art House & International       6.6          5.2         6.4    8.7
## 5                      Other       6.6          6.5         6.7    9.9
## 6         Mystery & Suspense       6.5          5.5         5.6   10.3
## 7         Action & Adventure       6.0          4.1         5.4   10.5
## 8                  Animation       5.9          5.0         6.2   10.4
## 9                     Horror       5.8          4.4         4.6    9.6
## 10 Science Fiction & Fantasy       5.8          5.0         5.1   10.0
## 11                    Comedy       5.7          4.1         5.3    9.8
```

Conclusion:

We observe that the standard deviations of the ratings are much larger for the IMDB ratings probably due to the huge numbers of voters. We have seen that the IMDB rating distribution looks like a normal distribution. This do not happen in the Rotten Tomatoes website.

The standard deviations are even larger for the Rotten Tomatoes critics' rating. Again the reason can be because of the smaller size of the sample.

In the fourth plot is evident how few documentaries are present in the sample. And how the genres as "comedy", "Mystery and suspense", "drama" and "other" are the most abundant in the sample

3 classification Levels. Categorical variables.

Definition of new variables: `IMDB.class` and `Audience.Class`

The new variable makes a new film classification similar to the Rotten Tomatoes classification ("Certified Fresh", "Fresh" and "Rotten").

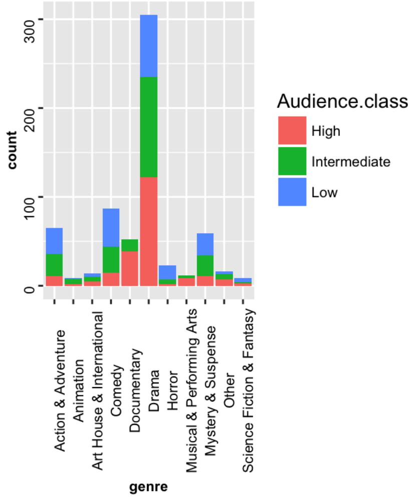
We transform the IMB ratings and Audience score onto "High", "Intermediate" and "Low".

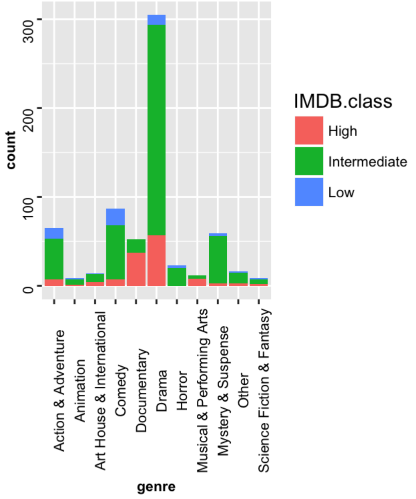
The limits are:

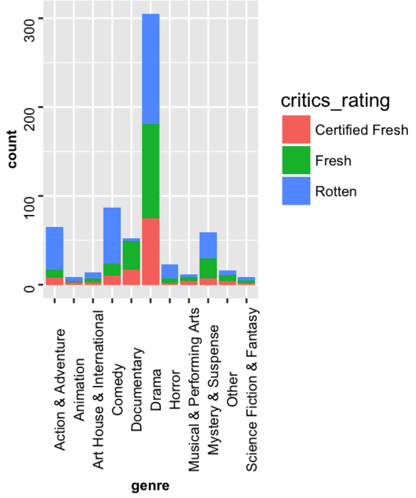
- High: $7.5 >$
- Intermediate: $5 - 7.5$
- Low: < 5

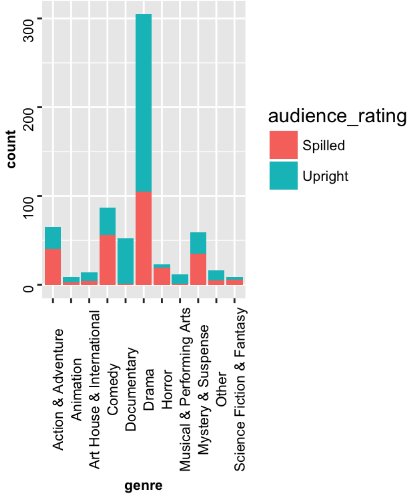
```
movies<-movies %>%
  mutate(IMDB.class=ifelse(movies$imdb_rating<5.0,"Low",ifelse(movies$imdb_rating>=7.5,"High", "Intermediate")))
movies<-movies %>%
  mutate(Audience.class=ifelse(movies$audience_score/10<5.0,"Low",ifelse(movies$audience_score/10>=7.5,"High","In
termediate")))
```

We add this new information to the bar plot of the number of films vs. genre.









Conclusion:

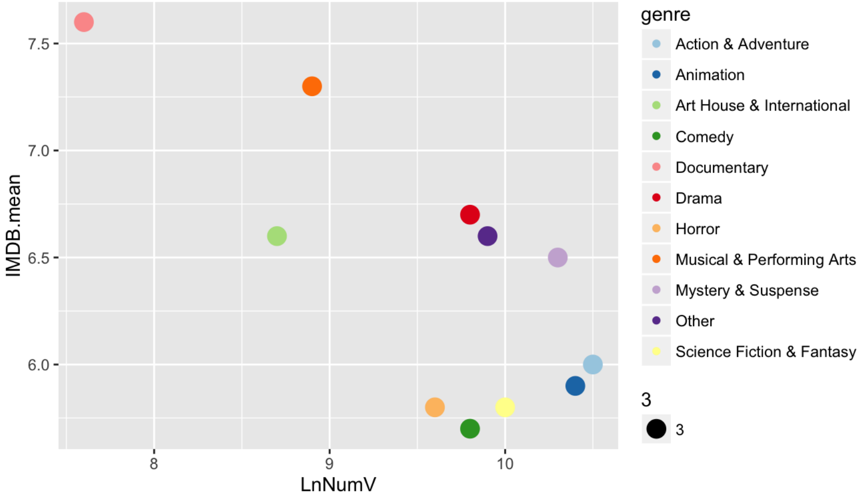
- IMDB rating and Rotten Tomatoes critics and audience differ principally on the negative rating. The Rotten Tomatoes audience and critics seems to punish more often than the IMDB voters
- At the same time, the Rotten Tomatoes audience is more generous with their ratings than IMDB voters and Rotten Tomatoes critics
- In the case of the IMDB most votes are Intermediate as it could be expected since the votes distribution follows a normal distribution as we have seen before.
- IMDB voters give scarcely negative/low votes. Probably due because they do not waste time seen a movie they dislike and hence not voting. Probably the IMDB voters population is a different population than the Rotten Tomatoes audience.

Rating's and Nr. of votes

The genres with less number of votes are the genres with highest rates: **Documentary**, **Musical & Performing Arts** and **Art House & International**

The genres with highest number of votes and more than 22000 (Ln ~ 10) votes have ratings under 6.5 points: **Comedy** is the genre with lowest rating, followed by "Science Fiction & Fantasy" and "Horror" movies.

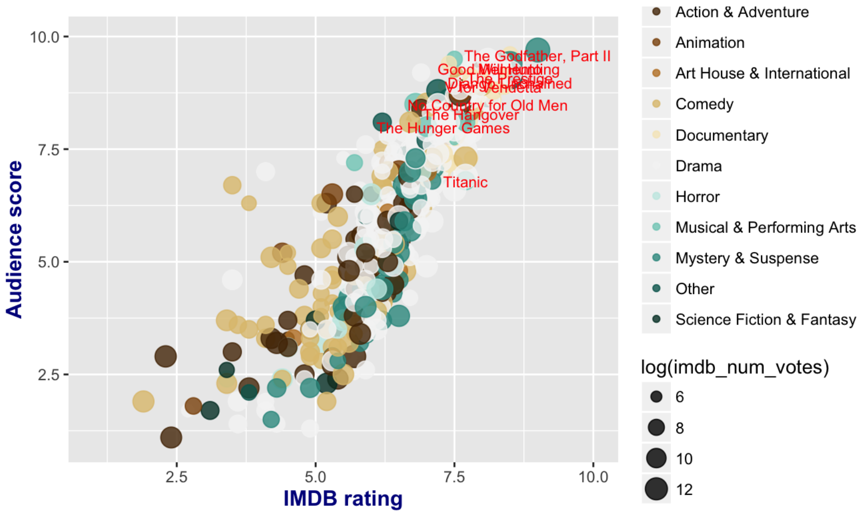
In all cases all genres have average ratings over 5 points.



Conclusion

The most massively voted genres have in general lower average genre ratings.

Audience Score vs. IMDB rating and gender. Top 10 Movies



Conclusion:

The range of the IMDB-rating is from 3 to 8 points. The **relationship between IMDB rating and the RT Audience score is significantly linear**. We will study this linearity in the next paragraph.

The Ln of the number of votes it is also plotted ($\mathrm{scattersize} = \mathrm{Ln}(\mathrm{numberof\ votes})$). Films with a large number of votes fall very often at the lowest rating's areas.

The most rated films are also shown in the graphic. This 10 most rated films are:

```
x<-movies[complete.cases(movies$top10),]; x<-x[c(1,13,18,16,14)]
x[5]<-round(log(x[5]),0)
names(x)<-c(names(x)[1:4], "LnVot")
x[order(-x$imdb_rating),]
```

```
##                      title imdb_rating audience_score critics_score LnVot
## 352 The Godfather, Part II         9.0             97            97    14
## 111       Django Unchained         8.5             91            88    14
## 384                Memento         8.5             94            92    14
## 477            The Prestige        8.5             92            76    14
## 469       Good Will Hunting        8.3             94            97    13
## 208          V for Vendetta        8.2             90            73    14
## 491 No Country for Old Men         8.1             86            93    13
## 140            The Hangover        7.8             84            79    13
## 610                 Titanic        7.7             69            88    14
## 633        The Hunger Games        7.3             81            84    13
```

In IMDB and RT the rates are very similar.

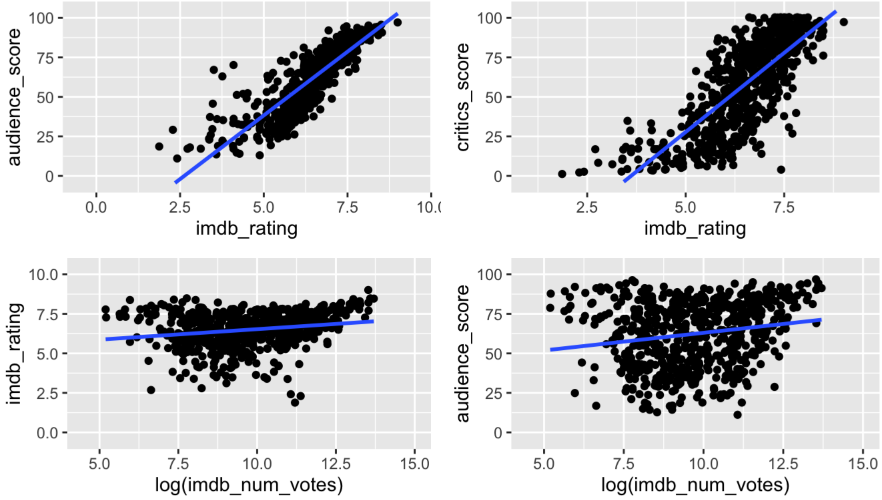
Also these films are among the most voted films.

Logx=14 means that x ~ 1 200 000

Part 2: Linear Regression Modelling

Relationships between variables

```
p2<-ggplot(data = movies, aes(x =imdb_rating, y = audience_score)) +
geom_jitter()+geom_smooth(method = "lm", se = FALSE) + ylim(-5,105) + xlim(-0.5,9.5)
# scatter plot critics_score (Tomatoes) vs. imdb_rating
p1<-ggplot(data = movies, aes(x =imdb_rating, y = critics_score)) +
geom_jitter()+geom_smooth(method = "lm", se = FALSE) + ylim(-5,105) + xlim(1,9.5)
# scatter plot imdb_num_votes ~ imdb_rating
p3<-ggplot(data = movies, aes(x= log(imdb_num_votes), y= imdb_rating)) +
geom_jitter()+geom_smooth(method = "lm", se = FALSE)  + xlim(4.5,15) + ylim(-0.5,10.5)
#Scatter plot imdb_num_votes~critics_score(Tomatoes)
p4<-ggplot(data = movies, aes(x= log(imdb_num_votes), y= audience_score)) +
geom_jitter()+geom_smooth(method = "lm", se = FALSE)  + xlim(4.5,15) + ylim(-5,105)
grid.arrange(p2, p1, p3,p4, ncol=2)
```



```
m<-lm(audience_score~imdb_rating, data=movies) ; m$coefficients[2]
```

```
## imdb_rating
##    16.12344
```

```
sm.m<-summary(m); sm.m$coefficients[2,]
```

```
##      Estimate    Std. Error       t value      Pr(>|t|)
##  1.612344e+01  3.673620e-01  4.388978e+01 2.077630e-196
```

```
m<-lm(critics_score~imdb_rating, data=movies) ; m$coefficients[2]
```

```
## imdb_rating
##    20.03167
```

```
sm.m<-summary(m); sm.m$coefficients[2,]
```

```
##      Estimate    Std. Error       t value      Pr(>|t|)
##  2.003167e+01  6.618978e-01  3.026398e+01 3.743006e-126
```

```
m<-lm(imdb_rating~log(imdb_num_votes), data=movies) ; m$coefficients[2]
```

```
## log(imdb_num_votes)
##           0.1326464
```

```
sm.m<-summary(m); sm.m$coefficients[2,]
```

```
##     Estimate   Std. Error      t value     Pr(>|t|)
## 1.326464e-01 2.488690e-02 5.329969e+00 1.357108e-07
```

In the context of the relationship between the critics' score (Rotten Tomatoes) and the IMDB rating the slope tell us that:

For each additional point on rating scale of the IBMD, the model predicts 16 more points of the Rotten Tomatoes audience score, on average.

For each additional point on rating scale of the IBMD, the model predicts 20 more points of the Rotten Tomatoes critics score, on average.

Every time that the number of voters is multiplied by 10, the model predicts 0.1 more points of the IMDB rating, on average.

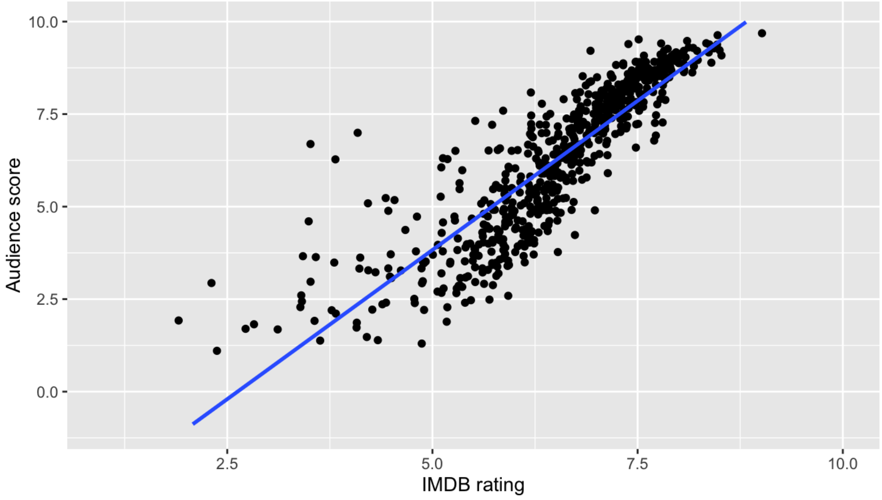
This means that with our model when IMDB rating is 3.6 points the model predicts 0 points for Rotten Tomatoes and for 8.6 IMDB rating points the model gives the maximum of possible critics score (100). The IMDB rating's interval is narrower.

The Rotten Tomatoes critics seems more generous than the IMDB critics

Model: One variable

Scatter Plot: Audience Score vs. IMDB rating.

We know already that there is a linear relationship between both variables. We use `geom_jitter()` command to add some randomness on the plot dots.



```
# Linear Regression
m1<-lm(formula = audience ~ imdb_rating,data=movies)
Sm.m1=summary(m1)
Sm.m1
```

```
##
## Call:
## lm(formula = audience ~ imdb_rating, data = movies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6800 -0.6567  0.0649  0.5689  5.2896
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.23284    0.24183  -17.50   <2e-16 ***
## imdb_rating  1.61234    0.03674   43.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.016 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
```

Visually there is a linear relationship between Rotten Tomatoes audience score vs. IMDB rating The fitted line is:

$$y = 1.612x - 4.233$$

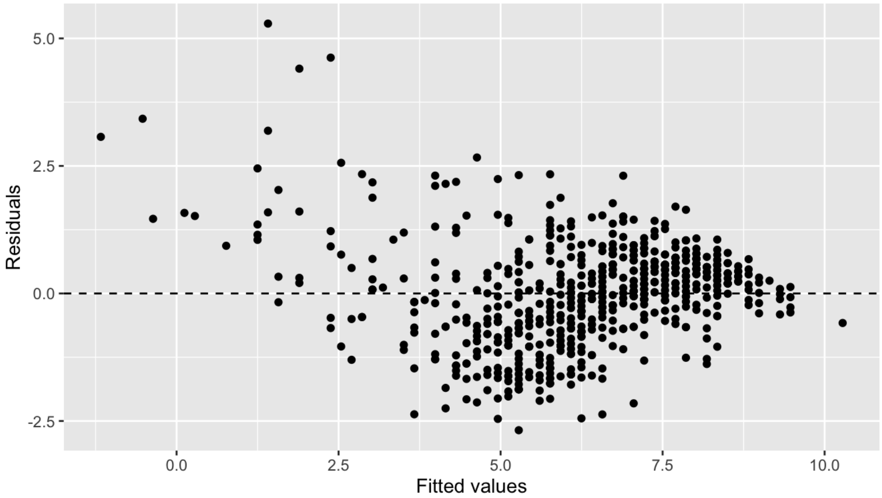
With $x =$ IMDB_rating and $y =$ Rotten Tomatoes audience_score

The adjusted R^2 is 0.74. And the p-value for the slope is ~ 0.

Diagnostic

Linearity: We have checked if the relationship between Rotten Tomatoes audience score and IMDB rating is linear using a scatter plot. We should also verify this condition with a plot of the residuals vs. fitted (predicted) values. We observe that the residuals are distributed around 0 hence the relationship is linear.

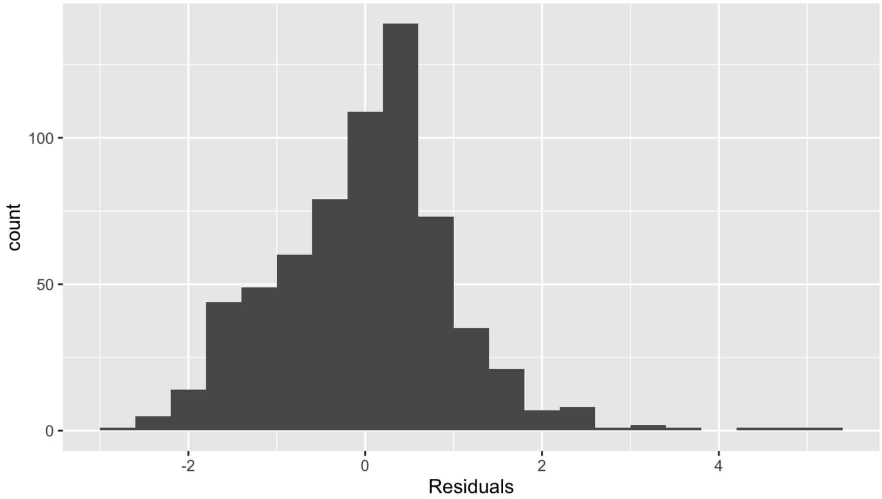
```
ggplot(data = m1, aes(x = .fitted, y = .resid)) +
  geom_point() + geom_hline(yintercept = 0, linetype = "dashed") +
  xlab("Fitted values") + ylab("Residuals")
```



Residuals Histogram: Normally distributed

We observe that the residuals are normally distributed.

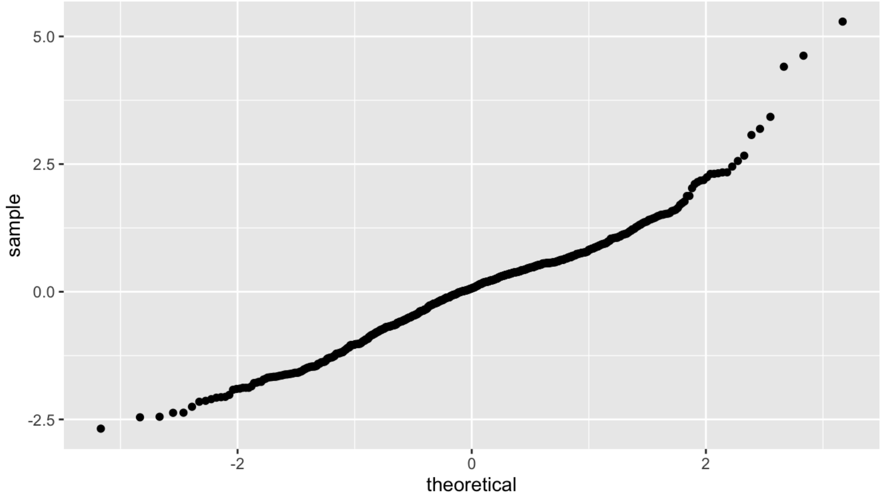
```
ggplot(data = m1, aes(x = .resid)) + geom_histogram(binwidth = 0.4) + xlab("Residuals")
```



Normal Q-Q plot.

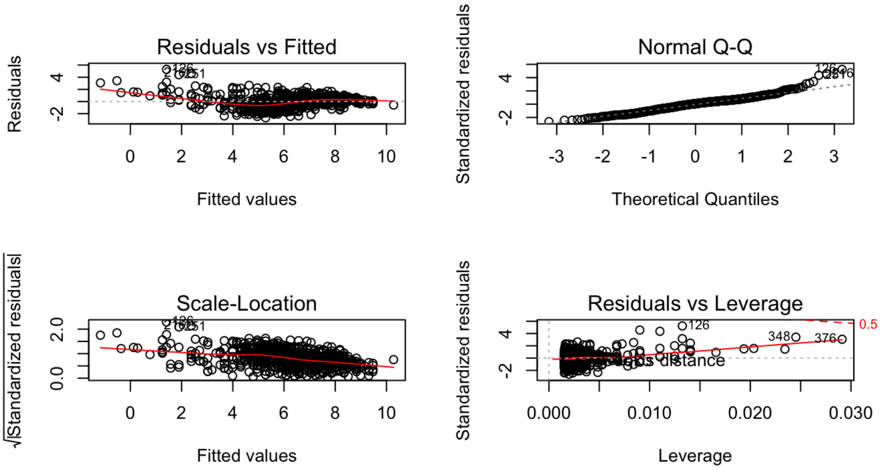
We check also the normal probability plot of the residuals. We see that the fitting is not very accurate at the tails, that is at the highest and lowest ratings.

```
ggplot(data = m1, aes(sample = .resid)) + stat_qq()
```



It is possible to plot all the `lm` analysis using the command `plot(lm)`

```
par(mfrow=c(2,2))
plot(m1)
```

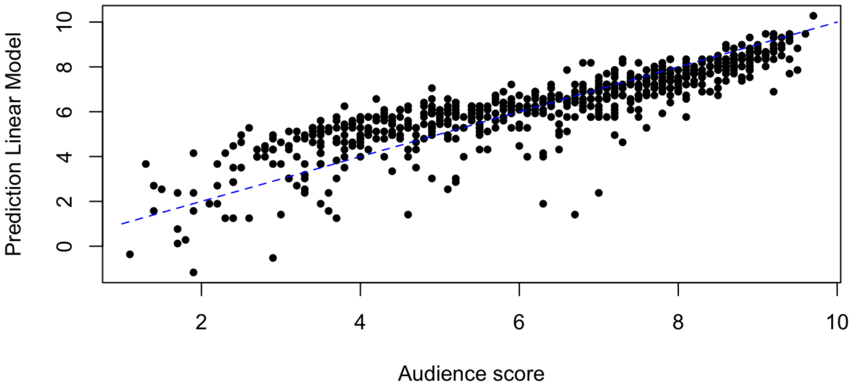


Predictions vs. Audiencience Score

We check also the relationship between the data we want to model `audience_scores` and the predictions from the linear model (based on `imdb_rating`)

The plot should have slope 1 and intersection at 0 (Blue dotted line in the plot)

```
coef1=m1$coefficients
prediction=coef1[1]+coef1[2]*movies$imdb_rating
plot(round(movies$audience,1),round(prediction,3),pch=20, xlab="Audience score", ylab = "Prediction Linear Model"
)
lines(c(1:10),c(1:10),type="l",col="blue",lty=2)
```



The Theoretical Rating from the model in the range between 1-5 are higher ratings than those of the audience scores. On the other hand around the 9 points area, the prediction is slightly lower than the known audience score. But in general the fitting is pretty accurate.

In summary, knowing the IMDB rating provides some information but on its own does not get us very far in predicting what my score would be.

Model: Many variables

Forward Approach: Adding predictors

We add more variables in order to improve the linear fitting. We add one variable at a time until the $R^2_A dj$ remains constant or decreases.

2 predictors `imdb_rating` and `genre`

We add more variables in order to improve the linear fitting. Some playing around shows that among the available candidates as imdb_rating and dummies for a few genres and studios (selected only from those with more than four movies in the data) give any leverage.

`summary(lm(audience~imdb_rating + genre ,data=movies))`

```
m<-lm(formula = audience ~ imdb_rating + genre ,data=movies)
Sm.m=summary(m)
Sm.m
```

```
##
## Call:
## lm(formula = audience ~ imdb_rating + genre, data = movies)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6946 -0.6195  0.0627  0.5615  5.0488
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -4.19520    0.27207 -15.419  < 2e-16 ***
## imdb_rating                     1.60342    0.04068  39.416  < 2e-16 ***
## genreAnimation                  0.97946    0.35155   2.786 0.005493 **
## genreArt House & International -0.01029    0.29240  -0.035 0.971940
## genreComedy                     0.23439    0.16231   1.444 0.149197
## genreDocumentary                0.20711    0.19615   1.056 0.291422
## genreDrama                      0.02961    0.13802   0.215 0.830196
## genreHorror                    -0.45930    0.23996  -1.914 0.056058 .
## genreMusical & Performing Arts  0.50689    0.31523   1.608 0.108326
## genreMystery & Suspense        -0.59951    0.17893  -3.350 0.000854 ***
## genreOther                      0.23126    0.27715   0.834 0.404357
## genreScience Fiction & Fantasy  0.05551    0.35165   0.158 0.874631
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9884 on 639 degrees of freedom
## Multiple R-squared:  0.7651, Adjusted R-squared:  0.7611
## F-statistic: 189.3 on 11 and 639 DF,  p-value: < 2.2e-16
```

```
#zz<-round(data.frame(Sm.m$coefficients[,4]),6) ; names(zz)="P-val"; zz
```

We take the 19 variables that seems to have some influence when modelling the data `audience_score` .

We check which variable increases the most R^2_Adj and we pick up this variable to the next step, that is, we add this variable to the model. Also we remove the variables whose P-values are extremely large. We end up with 10 possible candidates for the next step.

After this we loop again over the 10 variables left. And we repeat this process iteratively until R^2_Adj do not continue increasing.

We only show the three 1st loop and last ones.

Loops: Calculate all possible R^2

1 variable

`lm(audience ~ get(var_names[i]), data = movies)`

```
i<-1
R.square1<-NA
for (i in 1:19) {
  m<-lm(audience ~ get(var_names[i]), data = movies)
  x<-summary(m)
  v<-as.character(var_names[i])
  R2<-c(v,x$adj.r.squared)
  R.square1=rbind(R.square1,x$adj.r.squared)
}
R.square1=R.square1[2:20]
R.square=data.frame(cbind(var_names),R2.adj=R.square1)
R.square<-R.square[order(-R.square$R2.adj),]
var.ord=as.character(R.square[,1])
```

R^2 adjusted `audience ~ imdb_rating`

```
## [1] 0.7476034
```

2 variables

`lm(audience ~ imdb_rating  + get(var.ord.2[i]))`

```
##                V1            R2.adj
## 9   best_pic_win 0.747216950923816
## 4 imdb_num_votes 0.747227048678774
## 7   best_pic_nom 0.747866325386437
## 6    mpaa_rating 0.749397607042518
## 8        runtime 0.750104840141201
## 1  critics_score 0.751608179151888
## 5         studio 0.753858502904563
## 2 critics_rating 0.754313357613094
## 3          genre 0.761106503455103
```

R^2 adjusted `audience ~ imdb_rating + genre`

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

Since `genre` gives me the highest R^2_Adj. I chose `genre` as the variable to add to my linear regression model. I'll repeat this process until R^2_Adj doesn't cahnge or decreases.

3 variables

`lm(audience ~ imdb_rating + genre + get(var.ord.2[i]))`

R^2 adjusted `audience ~ imdb_rating + genre + critics_rating`

```
## [1] 0.7666122
```

4 variables

`lm(audience ~ imdb_rating + genre + critics_rating + get(var.ord.2[i]))`

R^2 adjusted `audience ~ imdb_rating + genre + critics_rating + studio`

```
## [1] 0.7722975
```

5 variables

`lm(audience ~ imdb_rating + genre + critics_rating + studio + get(var.ord.2[i]))`

R^2 adjusted `audience ~ imdb_rating + genre + critics_rating + studio+ runtime`

```
## [1] 0.7750826
```

6 Variables

`lm(audience ~ imdb_rating + genre + critics_rating + studio + runtime+get(var.ord.5[i]))`

R^2 adjusted `audience ~ imdb_rating + genre + critics_rating + studio+ runtime+ best_pic_nom`

```
## [1] 0.7756985
```

7 Variables

`lm(audience ~ imdb_rating + genre + critics_rating + studio + runtime + best_pic_nom + get(var.ord.5[i]))`

8 Variables

R^2 adjusted `audience ~ imdb_rating + genre + critics_rating + studio+ runtime+ best_pic_nom + thtr_rel_month`

```
## [1] 0.7773151
```

`lm(audience ~ imdb_rating + genre + critics_rating + studio + runtime + best_pic_nom + thtr_rel_month)`

9 Variables

R^2 adjusted
`audience ~ imdb_rating + genre + critics_rating + studio+ runtime+ best_pic_nom + thtr_rel_month + critics_score`

Adding this new variable decreases the $R^2_A dj$ value

```
## [1] 0.7770868
```

`lm(audience ~ imdb_rating + genre + critics_rating + studio + runtime + best_pic_nom + thtr_rel_month, thrt_rel_year + critics_scor`

9 Variables

We use another variable `director`

R^2 adjusted
`audience ~ imdb_rating + genre + critics_rating + studio+ runtime+ best_pic_nom + thtr_rel_month + critics_score + director`

```
## [1] 0.8267465
```

`lm(audience ~ imdb_rating + genre + critics_rating + studio + runtime + best_pic_nom + thtr_rel_month + thtr_rel_year+critics_score`

Addidg more TOO MANY variables does not increase the accuracy of the fit BUT DIMINISH it. The $R^2.adj$ increases at most 0.0001. However the plot Standardized residuals vs. Theoretical Quantiles (Normal Q-Q plot) shows no improvement.

Diagnostic Multiple Linear Regression: Model with 6 and 9 variables.

6 variables: `imdb_rating + genre + critics_rating + studio + runtime + best_pic_nom`

9 variables: `audience ~ imdb_rating + genre + critics_rating + studio + runtime+best_pic_nom +thtr_rel_month + thtr`

We show the 10 dummy variables with the lowest P-value together with the adjusted R^2 of this regression model.

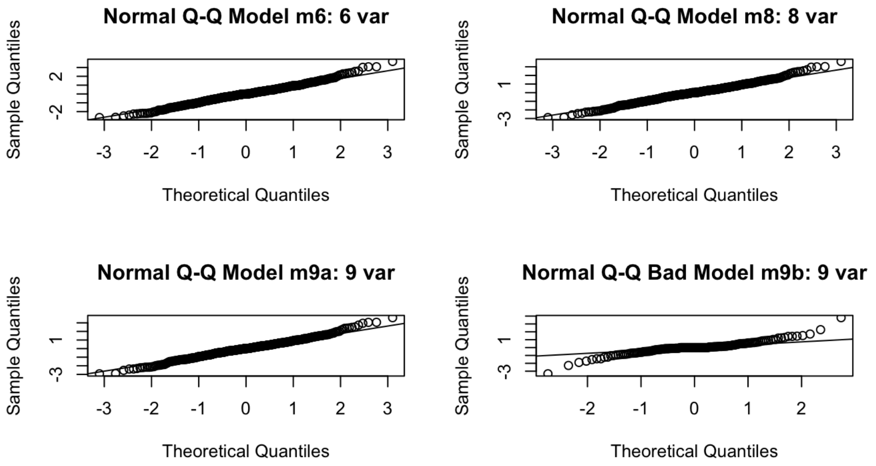
```
##                                      Var P_val
## 1                            (Intercept) 0.000
## 2                            imdb_rating 0.000
## 3                genreMystery & Suspense 0.010
## 4                    critics_ratingRotten 0.001
## 5  studioAmerican International Pictures 0.007
## 6                studioChloe Productions 0.008
## 7                      studioFirst Look 0.000
## 8               studioGenius Productions 0.002
## 9                       studioLionsgate 0.003
## 10                              runtime 0.008
```

We control the linear regerssion through Normal QQ-Plots.

We check haw the theretical quantiles fits the prediction and we campare among the different models with different amount of predictors

Normal Q-Q plot

```
# zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz
# We apply the lm function to a formula that describes the variable audience_score
# by the multiple variable waiting, and save the linear regression model in a new variable
# called m. Then we compute the standardized residual with the rstandard function.
par(mfrow=c(2,2))
m6.stdres = rstandard(m6)
qqnorm(m6.stdres,main="Normal Q-Q Model m6: 6 var")
qqline(m6.stdres)
m8.stdres = rstandard(m8)
qqnorm(m8.stdres,main="Normal Q-Q Model m8: 8 var")
qqline(m8.stdres)
m9.stdres = rstandard(m9)
qqnorm(m9.stdres,main="Normal Q-Q Model m9a: 9 var")
qqline(m9.stdres)
m10.stdres = rstandard(m10)
qqnorm(m10.stdres, main="Normal Q-Q Bad Model m9b: 9 var")
qqline(m10.stdres)
```



```
par(mfrow=c(1,1))
```

Conclusion:

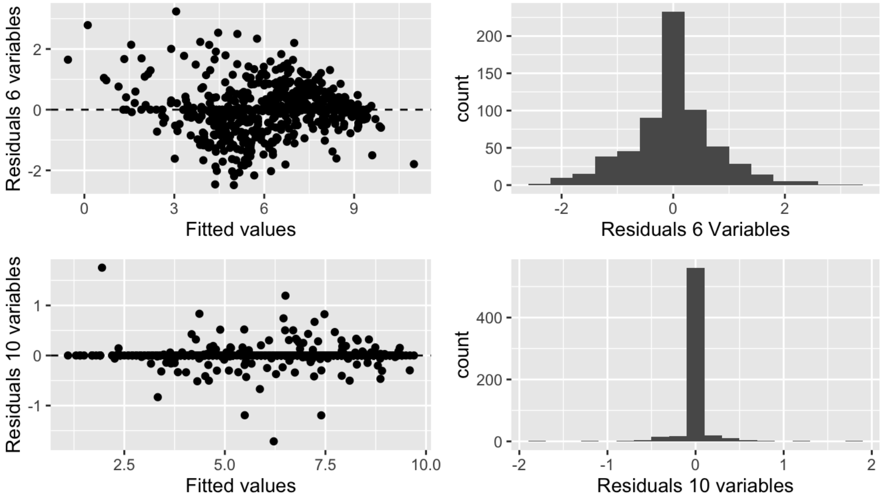
We can see how adding more variables does not improves the model that much.

In fact, when we use 10 variables we can see how most part of the fitting get worse at the tails. At the same time the residues plot, Residuals vs. Fitted, shows how the distance residue-model decreases considerably. Hence is an error to include that many variables.

Diagnostic: Residuals' Scatter Plots and Histograms

`audience ~ imdb_rating + genre + critics_rating + studio+ runtime+ best_pic_nom`

Residuals vs. Fitted residues values –> We check the ramdomness Histograms -> We check if the residues are normally distributed



The residues of both models, with 6 and 10 variables, show randomness and normality. But as explained before the Normal Q-Q curve for the model with 10 varibables does not shows linearity.

Part 4: Summary:

Problem: Find a method to determine how much audience likes a movie. Or which rate will general audience rate a film. It is like we want to buy films for a TV channel and we want that audience likes the films we decide to show.

Then our task is to find parameters from our databases that can predict the rating audience will give to a particular movie.

We want to make a linear model. We need to find the most important variables that can predict audience rating using a linear regression model

Data

Q1: What can the data tell us: Particularities and possible relationships between variables.

Q2: Can we find some parameter that correlates with high ratings.

Q3: She is also interested in learning something new about movies

- The variables, `imdb_rating` (IMDB), `audience_score` (Rotten Tomatoes), `critics_score` (Rotten Tomatoes) shows large similarities. However standard variations of the Rotten Tomatoes ratings are much larger than the ratings of IMDB.
- IMDB ratings distribution is almost normal distributed, probably because the large amount of voters has produced this normality (central limit theorem)
- Ratings are left skewed since the upper bound is well defined (equal to 10)

It is possible that people uses to rate films at IMDB and RT when they consider that a film deserves a rate of at least 3. The highest bound is 10. However although the effective limit is 3 the lowest limit is 1.

- The ratings depend strongly on the movies genre.
- The highest rating are the most "uncommon" genres as for instance "Documentaries" and Musical & Performing Arts
- The lowest rated films are 'Comedy', 'Mystery' and 'suspense' and 'drama' (lowest to highest)
- IMDB votes are often in the middle of the scale, as expected from the normal distribution of the votes.
- Rotten Tomatoes ratings are more negatives than IMDB's ratings, principally critics' votes. At the same time Rotten Tomatoes audience rating are also more generous (The density plot shows two picks, one at low rates and other at high rates)
- Amount of voters and genre rating mean are highly related. Large amount of votes means in general low ratings.
- I suppose that when a film is bad people do not waste their time watching it or rating it. However people rate films when they think it is ok or good and they take their time rating them. At the same time the larger choice of films as comedies and dramas will produce larger amounts of votes, hence approaching more to normality.

Modeling: Linear Regression

Q4: What attributes make a movie popular.

Q5: We would like to buy films for "our" tv channel that the most audience would like. Can we find a model gives us what rating people will give the movie. It is high rating enough for determining if the film is going to be popular?

- **Linearity** there is a linear relationship between Audience score and the IMDB rating.

The diagnosis of the fitting corroborates this.

- We use the forward model for the linear regression model. First we choose the variables that gives P values under 0.05. After this we add one variable at each time, the one that increases the adjusted R^2.
- After 6 variables the gain is not big.
- We build a model with 10 variables. The diagnosis of this model shows that this is not a good model.

Limitations:

- The way IMDB and Rotten Tomatoes as well as the method used to determine the ratings is not very clear and seems to be confusing or non realistic in many cases.
- In all cases people voting has to be registered. This means that people that vote can be a differentiated group of the whole population of a country, estate, etc.
- Rotten tomatoes information missed the data about the quantity of votes. At any case Rotten Tomatoes exist shorter time than IMDB so the number of votes has to be much lower than in IMDB
- It can also be big differences between rotten tomatoes voters and IMDB.
- IMDB rating is completely numeric but RT is categorical. This can lead to differences on how people decide if a film is good or bad
- More information is needed about the way data is recorded, as well as the dates the data is recorded.

Final Conclusion:

- The IMDB rating is the variable that is going to help us more to guess the rating general audience will give a movie. The genre is also important and slightly the critics rating, the studio that produce the film, the time the movie has been on the theaters as well as if the film has been nominated as best movie or not.
- In the context of the relationship between the critics score (Rotten Tomatoes) and the IMDB rating the slope tell us that:
    - For each additional point on rating scale of the IBMD, the model predicts 16 more points of the Rotten Tomatoes audience score, on average.
- We construct a linear model with these 6 variables.
- However high ratings, in general, are not related with the amount of people that watch-rate the movie. A documentary and give very high rates but however very few people is going to watch it.
- Data shows that bad movies have very few votes or not voted at all.
- If we want to a lot of people watching a film and liking it we would need more data, as for instance amount of Rotten Tomatoes voters or the money that a particular movie has gotten since it was released.