TV Show Platforms and Ratings

OIM 454

Alain Duplan

Hoseung Baek

Linus Hsu

Corey Kozlovski

Table of Contents

Project Description

For our project, we decided to use two well-known media databases (IMDb and Rotten Tomatoes) that were categorized by factors of Year | factors of Age, and the data were distributed to platforms based on availability. In recent years, the use of big data for entertainment and media has grown exponentially, and we had a particular interest in this dataset because the four major platforms Netflix | Hulu | Prime | Disney+ are all fighting competitive advantage in the saturated streaming service market. Those in the media industry need to have a general understanding of what movies are in a selected platform to gauge its reach. Additionally, shareholders need a visual representation of how the products they're investing in are faring. Also, they should see where the data is leading and what kind of prediction models can be used to further understand the TV show industry. The goals for our analysis were to figure out if there was a relationship between platform availability and ratings, if there was a relationship between IMDb and Rotten Tomatoes scores, and if it was possible to predict ratings based on platform availability.

This project shows an overview of our early exploratory data analysis, the correlation of variables that are to be used to create a random forest model, and KNN prediction model.

First, we added a column to the initial dataset with an index so that when we identify which movies are clustered together, we could easily pull an ID value.

We also wanted to research deeper into creating the models and decided to create predictive models in R/Rmd. After looking into different models, we decided to use both a random forest and KNN predictive model. By using our factors as predictors, we attempted to use random foresting to predict the IMDb and Rotten Tomatoes scores. The KNN predictive models derived from the dataset shows us classified clusters. Based on the distance of how far out a particular rating a show had, it would be counted within the cluster (if the distance was short) and would be counted in a separate cluster (if the distance was far).

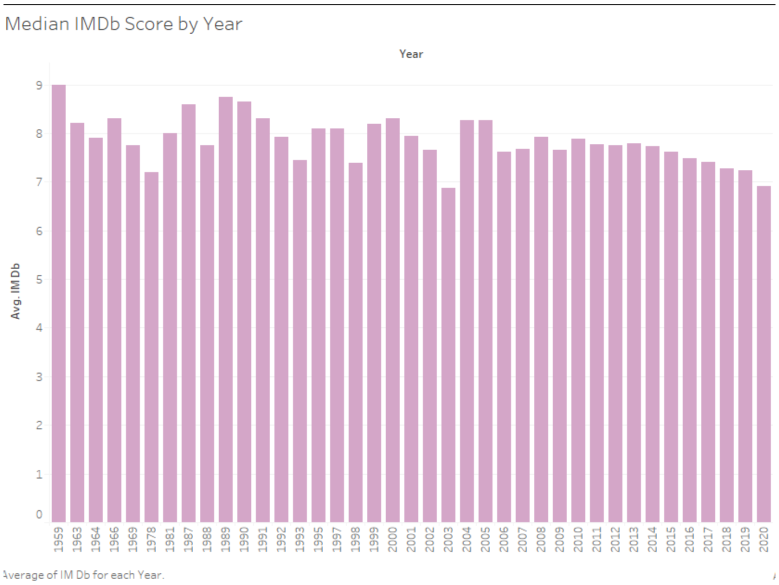
Data Preprocessing

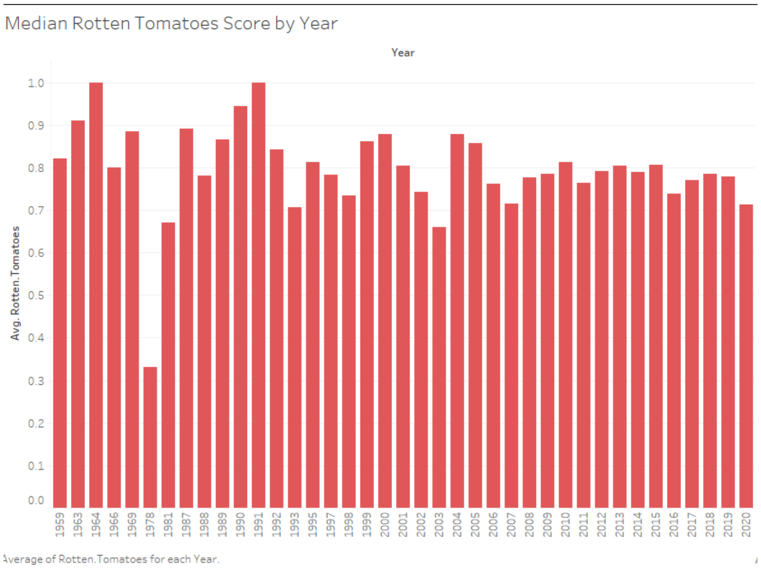
We pulled our raw data from a Kaggle dataset containing 5564 different shows. The variables the data set included was unique show ID and title, the Rotten Tomatoes scores as percents, IMDb scores as a number between 0.0 and 10.0, the year of release ranging from 1901 to 2020, age demographic(7+, 13+, 16+, 18+ and all ages), separate columns for Netflix, Hulu, Prime, and Disney+ containing the binary variable 0 or 1 (1 indicating the property being on the said platform), and a type column holding the binary variable 0 or 1 (where 1 would be a TV show and 0 being a movie). From there, our team set out to understand what data was going to be used for the research analysis. To ensure that we were working on complete cases, we removed rows from our dataset that had missing values. By doing so, all movies were removed from our dataset as movies have a different age rating system than what is listed therefore it was always empty. This left us with only 931 remaining rows to work with. We also dropped our unique show ID and title variables as they are irrelevant to our analysis. After adding an index, we sorted the dataset highest to lowest ratings by scores in the Rotten Tomatoes and IMDb database. There wasn't much to identify within the dataset by sorting ratings, so we decided to find if there was a correlation between the Rotten Tomatoes dataset and IMDb dataset to see if shows that scored higher in R_Tomatoes would also score high in the IMDb dataset. The result showed that there was a high correlation between the datasets, and the higher a score was in one set, the higher a score would be in the other. There were outliers where the result was not a linear relationship, but most of the dataset followed this strongly correlated positive linear relationship.

After this, we wanted to explore the data further for more preprocessing of prediction models. For both of our predictive models, we randomly split our dataset into a training and testing set using a 60/40 split.

Furthermore, in the KNN dataset, we set factor classes to all non-numerical variables so that we would not run into factor variables (KNN requires output variable to be a factor and the test|training sets to be numerical).

To help understand the data, we created graphs in Tableau:





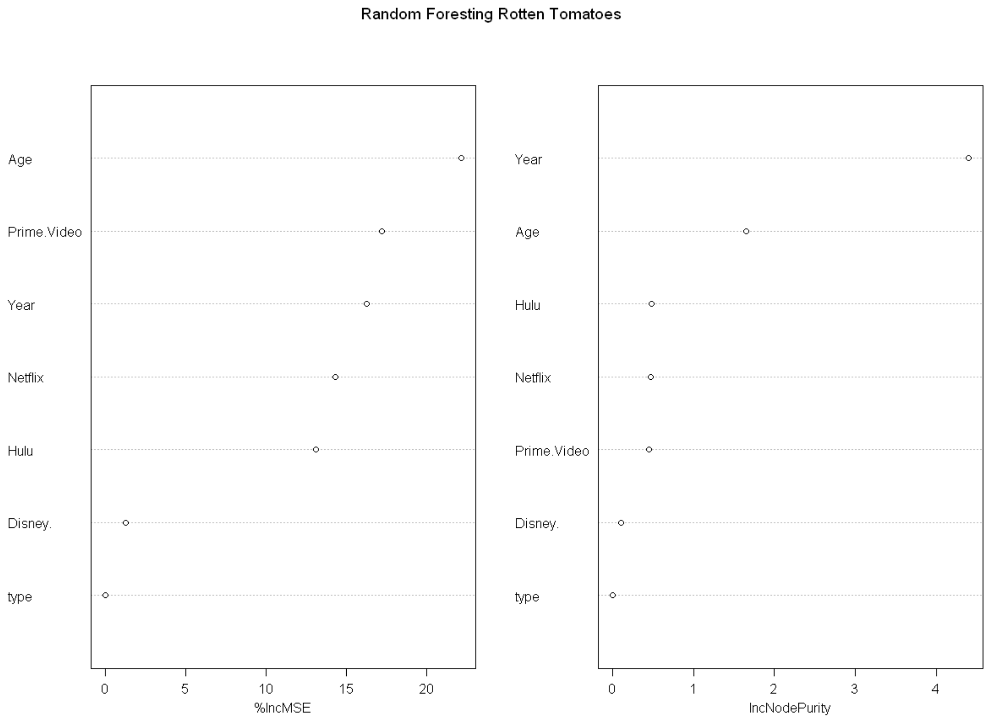
By visualizing the median scores of each movie database we can see how they compare to each other year by year. This provided us with our first glimpse into whether there would be a relationship between the scores of the two different databases.

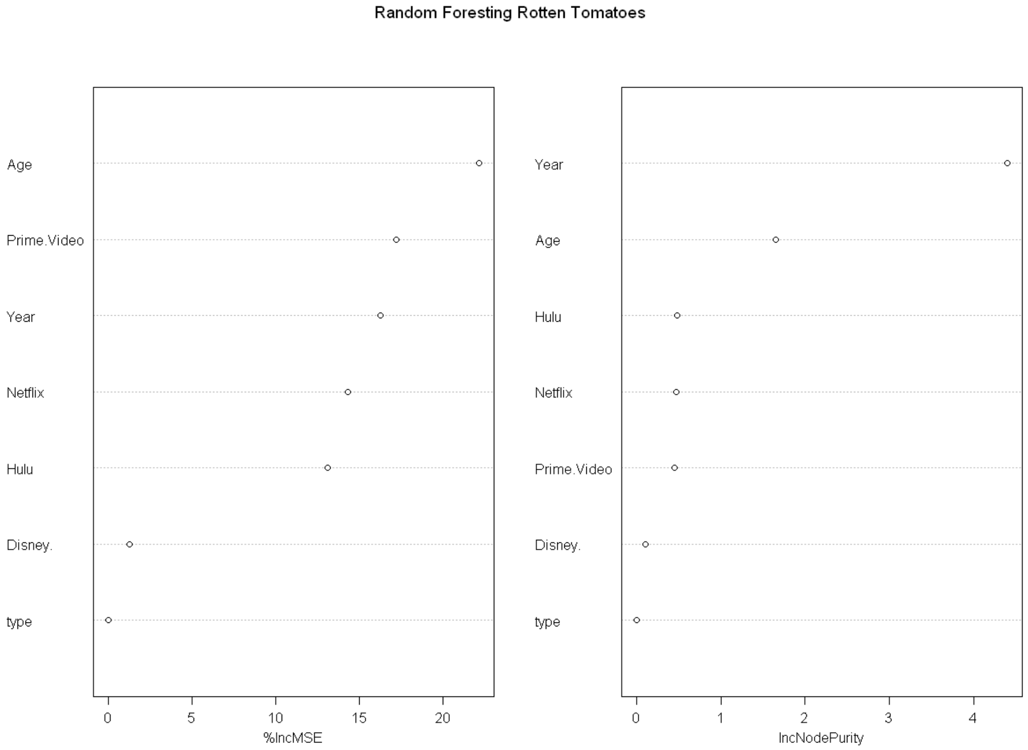
Models

For the data, we decided to create a random forest tree model and multiple KNN prediction models that would help us best predict ratings for platform ownership of newer TV shows. We considered a linear regression model, but after some experimentation we realized that no variables had a linear correlation with the IMDb or Rotten Tomatoes score so we abandoned that model.

By nature of a random forest model, multiple decision trees are created based on the dataset, and the average output of the trees is returned for our prediction. After setting our seed to '1234', two forests were created using our previously established training set: one for IMDb scores and one for Rotten Tomatoes scores. The scores were predicted using the age ratings, year of release, and platform availability variables. We decided to exclude the scores from our predictive variables as we did not want it to skew the results as we already saw a strong positive correlation. Both forests were created using 500 trees as the model. We set four to be the number of variables as randomly selected candidates for each split. Four was selected to avoid overfitting, but also because it happens to be the number of platforms in our dataset. We also set out to test the importance of our predictors.

Once both forests were completed, we plotted the importance of the variables using the percent increase in mean squared error and increase in node purity.

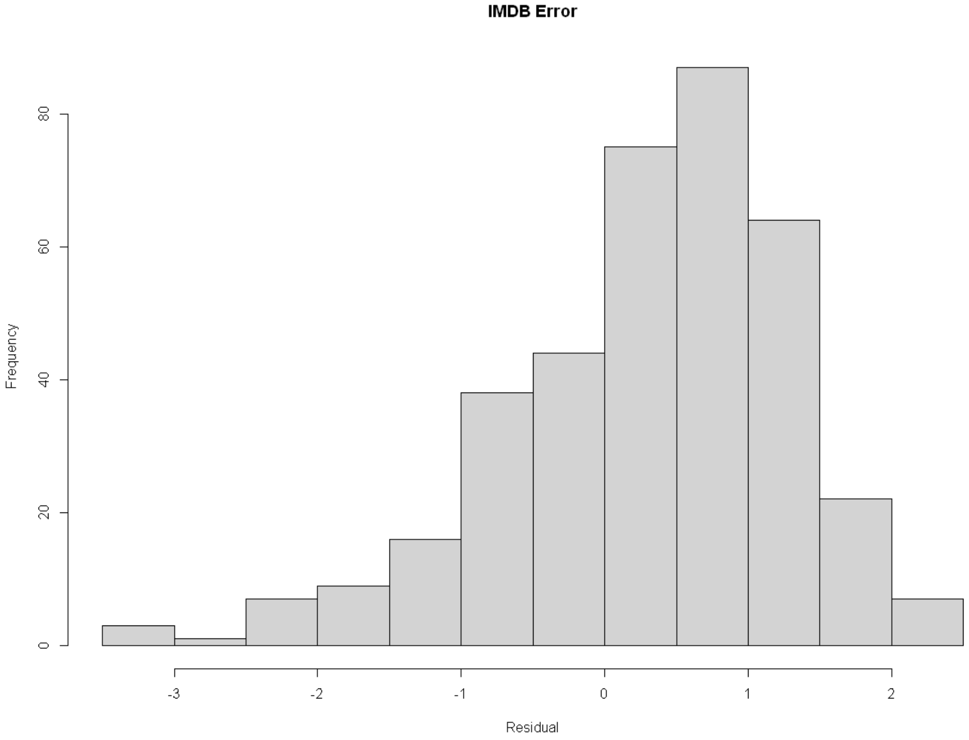


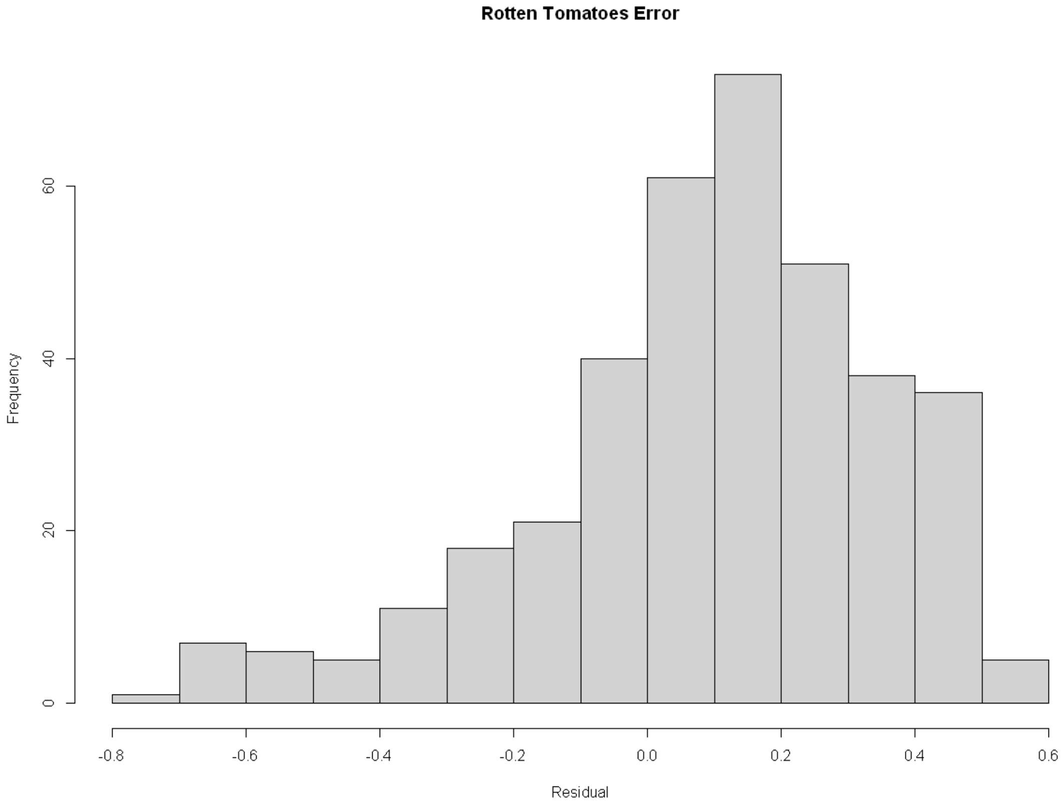


Using both measures we see that in both models, the type was the least important variable. This is due to the fact the type was consistent among all the rows in our dataset. Disney+ is also deemed unimportant which may be caused by the extreme exclusivity of Disney+ and how new and small it is compared to the other platforms. Another noteworthy mention is age and year being the consistent top variable, meaning that there might be a possible relationship.

With our models created, we applied our models onto our testing set and began evaluating our predictions. When we look at our root mean squared error of models, we see that in our IMDb forest, it was 1.036 stars and in our Rotten Tomatoes forest, it was 27.18%. This error was huge and was too big to ignore. To put into context, the 27.18% difference on Rotten Tomatoes is out of a 100% total possible score and could be the difference between a highly rated score like 90% and something that is only subpar being 63%. The ~1 star error of the IMDb model is not as extreme but it is still worth mentioning.

Since our RMSE was so big we decided to plot our residuals onto a histogram.



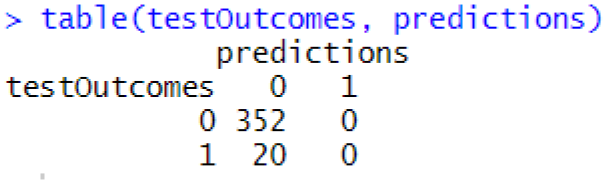


We see in both models, the histograms are skewed to the left, with our modes falling slighting above 0. This means that our models tended to overpredict. To verify, we found the center of our residuals, using the mean of .30112 stars for IMBd and 9.385% for Rotten Tomatoes. In an ideal model, both the averages would be 0, however, as shown through the histogram and the average of residuals, most of our predictions fell above the actual values, showing signs of a bias.

Due to the large RMSE that was calculated and the bias exhibited by the histogram, we were not able to call our random forest an accurate model to use for predicting our IMDb and Rotten Tomatoes scores. We could not conclusively determine if there was a relationship between the platform availability and the ratings of the TV shows.

For the KNN model, there are 4 observable models similar to the table made below, that each predicts the availability of show within the platform it was tested against. After doing a

60/40 preset determined by the group, we determined that the performance of the predicting models was not accurate; the number could be high, but without a larger dataset or more variables to work with, a general KNN prediction ended up being not such an effective model when observing and predicting platform availability.



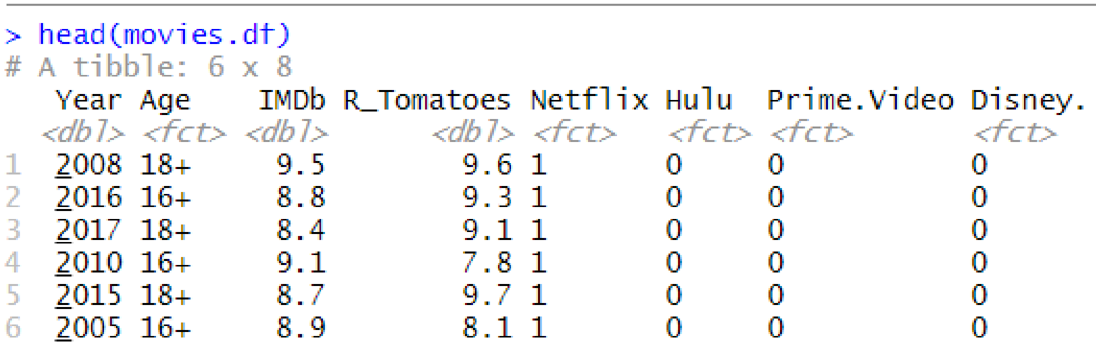
```
> table(testOutcomes, predictions)
           predictions
testOutcomes   0   1
           0 352   0
           1  20   0
```

Results and Discussion

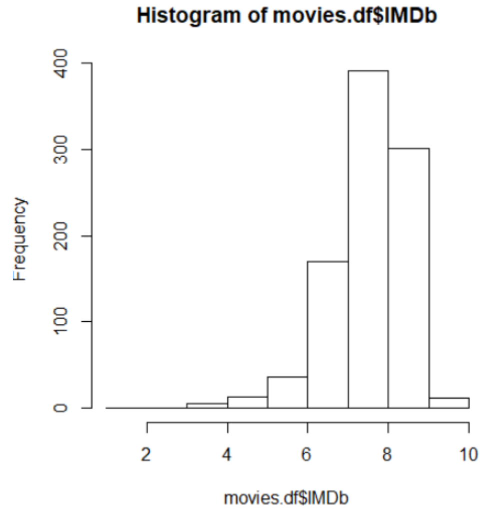
Our team believes that the early data exploration process was an integral part of the research process as a whole, as a deep discussion of understanding the dataset in the initial stages of this project helped create a vision for the resulting prediction models. We assumed that there would be high correlation between the two movie databases, and assumed correctly that our prediction models had a low accuracy rate. The random forest tree model shows a huge jump from 3 stars to 4 stars, for example. This may not seem like much for a larger database where there are thousands of data entries, but with such a limited dataset and not enough variables to put them into a prediction model, we concluded that there was no relationship between platform availability and show rating. From the KNN model, 4 separate tables were made to identify prediction accuracy; we concluded that a higher accuracy rate does not mean reliability, some platforms simply had a wider category to choose from than others, and ultimately the recall output was low for all 4 models.

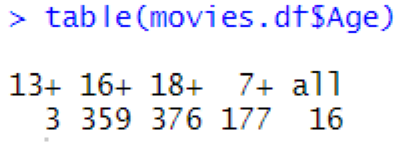
Summary

Through this process, we ultimately found out that data retrieved from the real world has major gaps and limitations, and it is important to correctly mine the right data to make a reliable prediction model. For example, linear regression could not be performed on this dataset. Our random foresting and KNN models were also limited by the data. One piece of advice we would share with any future students is to look for a dataset that will be compatible with many different data mining models. Many prediction models can be used, as random foresting and KNN are just 2 of a variety of prediction models to choose from.



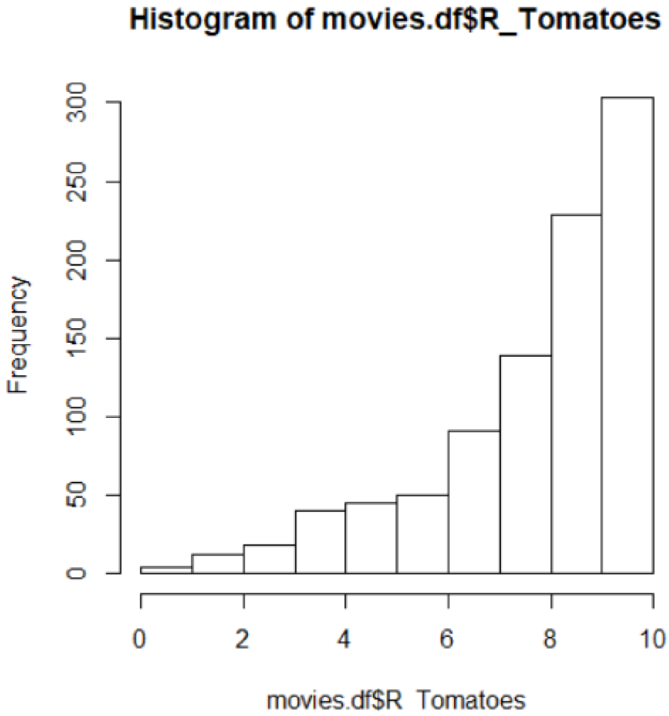
```
> head(movies.dt)
# A tibble: 6 x 8
   Year Age    IMDb R_Tomatoes Netflix Hulu  Prime.Video Disney.
  <dbl> <fct> <dbl>      <dbl> <fct>   <fct> <fct>       <fct>
1  2008 18+     9.5        9.6 1       0     0           0
2  2016 16+     8.8        9.3 1       0     0           0
3  2017 18+     8.4        9.1 1       0     0           0
4  2010 16+     9.1        7.8 1       0     0           0
5  2015 18+     8.7        9.7 1       0     0           0
6  2005 16+     8.9        8.1 1       0     0           0
```

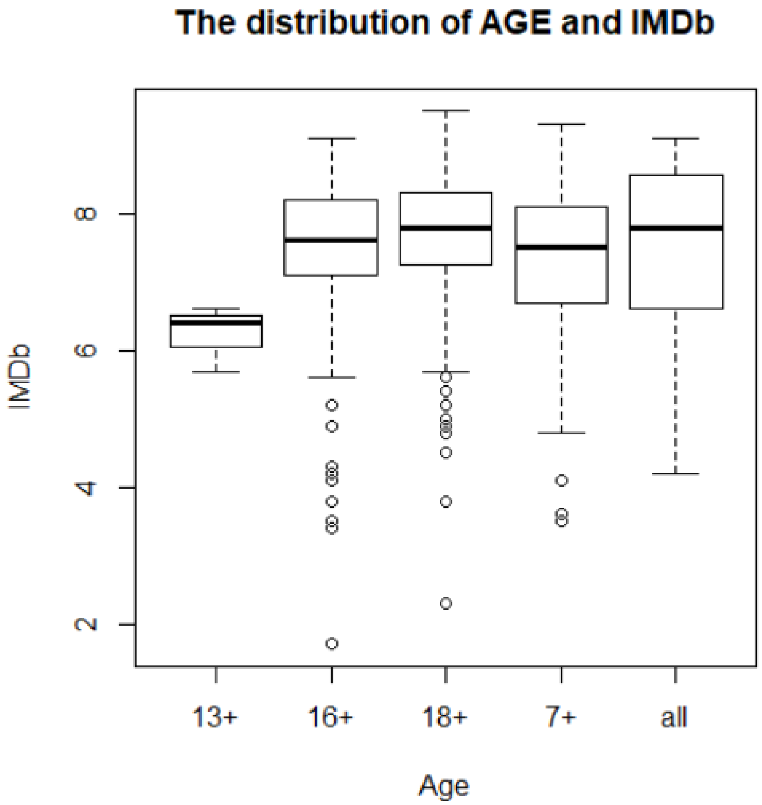


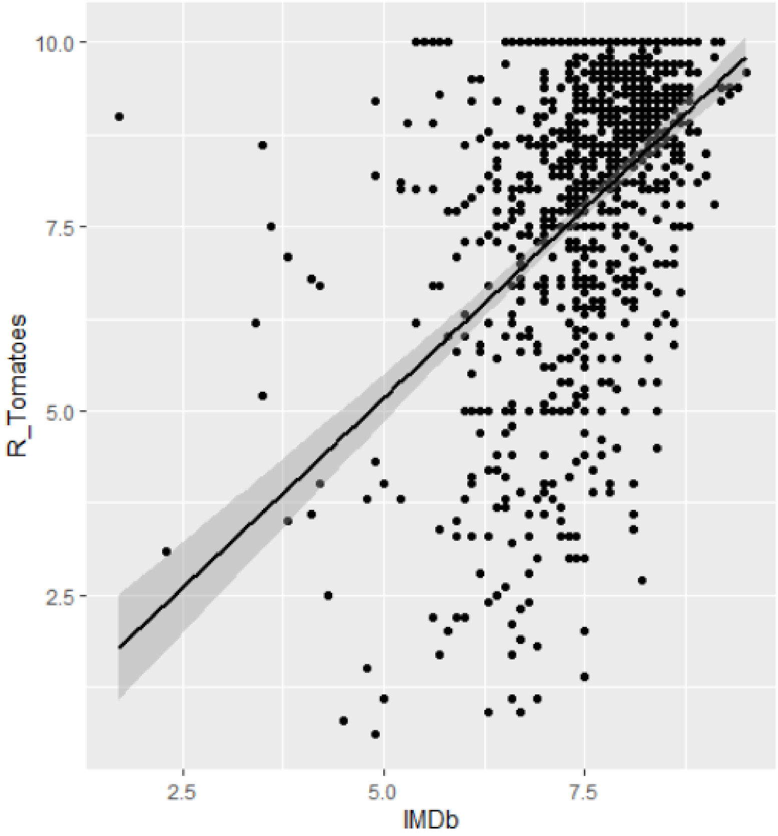


```
> table(movies.dt$Age)

13+ 16+ 18+  7+ all
  3 359 376 177  16
```


Histogram of movies.df$R_Tomatoes
Frequency
0
50
100
150
200
250
300
0
2
4
6
8
10
movies.df$R_Tomatoes


The distribution of AGE and IMDb
IMDb
2
4
6
8
13+
16+
18+
7+
all
Age


R_Tomatoes
2.5
5.0
7.5
10.0
2.5
5.0
7.5
IMDb