# MovieLens Project

Richa Gautam

4/11/2020

## Utilizing the Movielens Database to Predict Movie Preferences

### Introduction

Why do we watch the movies we do? In the era of binge-watching, the answer to this question has largely been "because we watched a similar movie before." Streaming services' algorithms curate our dashboard to recommend movies that they find most likely to appeal to us. This appeal could be based on your bias towards certain genres, or an era, or an actor or a director. I for example will watch anything with Andy Samberg in it. My partner on the other hand is a fan of bad horror movies.

We can therefore hypothesize that people tend to like things based on certain foundational preferences. This insight is hugely useful when trying to predict human behavior. Imagine you are inviting friends over for a movie. What movie should you suggest that will please most of the crowd?

### Overview

In this report, I will outline the steps taken to create a recommendation system given the genre, year, and 9000055 user ratings of 10677 unique movies by 69878 unique users from the Movielens database.

### Executive Summary

I hypothesized that this problem will require a regression machine learning algorithm, as the dependent variable (variable being predicted; ratings) is continuous, and the independent variables (variable being used to predict; genre, and users, movies) are categorical. Using the xxx machine learning method, I was able to predict movie ratings with xx% accuracy. The root mean squared error (RMSE) was xx.

### Methods

The dataset was obtained from http://grouplens.org using code provided by the Data Science Capstone course being taught by Harvard Extension school. The final dataset had 9000055 observations of 6 variables - userId, movieId, rating, timestamp, title, and genres.

Of the variables, our independent variables (IVs) were "genres," a character vector, "year," which was extracted from the "title" column, and "userId," an integer vector. Our dependent variable (DV) was "rating," a numeric vector.

#### Pre-processing

The code provided by edx led to a dataset without any incomplete rows, and another validation dataset with 10% of the full dataset's observations. I began with the full dataset.

The "genres" column could contain one or multiple genres, separated by "|". So my first task was to separate the responses in that column. However, working with dataframes is more time consuming, so I decided to convert the separated genre data into a matrix.

As you can see, the genre data from the large dataset has been dummy-coded:

```
##   userId movieId rating timestamp                         title
## 1      1     122      5 838985046              Boomerang (1992)
## 2      1     185      5 838983525               Net, The (1995)
## 3      1     292      5 838983421               Outbreak (1995)
## 4      1     316      5 838983392               Stargate (1994)
## 5      1     329      5 838983392 Star Trek: Generations (1994)
## 6      1     355      5 838984474       Flintstones, The (1994)
##                          genres Action Adventure Animation Children Comedy
## 1                Comedy|Romance      0         0         0        0      1
## 2          Action|Crime|Thriller      1         0         0        0      0
## 3  Action|Drama|Sci-Fi|Thriller      1         0         0        0      0
## 4       Action|Adventure|Sci-Fi      1         1         0        0      0
## 5 Action|Adventure|Drama|Sci-Fi      1         1         0        0      0
## 6       Children|Comedy|Fantasy      0         0         0        1      1
##   Crime Documentary Drama Fantasy Film_noir Horror Musical Mystery Romance
## 1     0           0     0       0         0      0       0       0       1
## 2     1           0     0       0         0      0       0       0       0
## 3     0           0     1       0         0      0       0       0       0
## 4     0           0     0       0         0      0       0       0       0
## 5     0           0     1       0         0      0       0       0       0
## 6     0           0     0       1         0      0       0       0       0
##   Sci_Fi Thriller War Western year rating_datetime
## 1      0        0   0       0 1992            1996
## 2      0        1   0       0 1995            1996
## 3      1        1   0       0 1995            1996
## 4      1        0   0       0 1994            1996
## 5      1        0   0       0 1994            1996
## 6      0        0   0       0 1994            1996
```

### Preliminary Predictions

#### User Effects

Movie ratings will be impacted by the user, as users have different tastes and some users are likely to rate movies more highly on average than others. Table 1 shows the mean rating and standard deviation for the first 6 userIds.

```
## # A tibble: 6 x 4
##   userId Users_Rated Mean_Rating SD_Rating
##    <int>       <int>       <dbl>     <dbl>
## 1      1          19        5        0
## 2      2          17        3.29     0.920
## 3      3          31        3.94     0.761
## 4      4          35        4.06     1.16
## 5      5          74        3.92     1.11
## 6      6          39        3.95     1.10
```

Table 1: Mean rating and standard deviation for the first 6 userIds.

Figure 1 visualizes the number of ratings for the first 100 userIds.

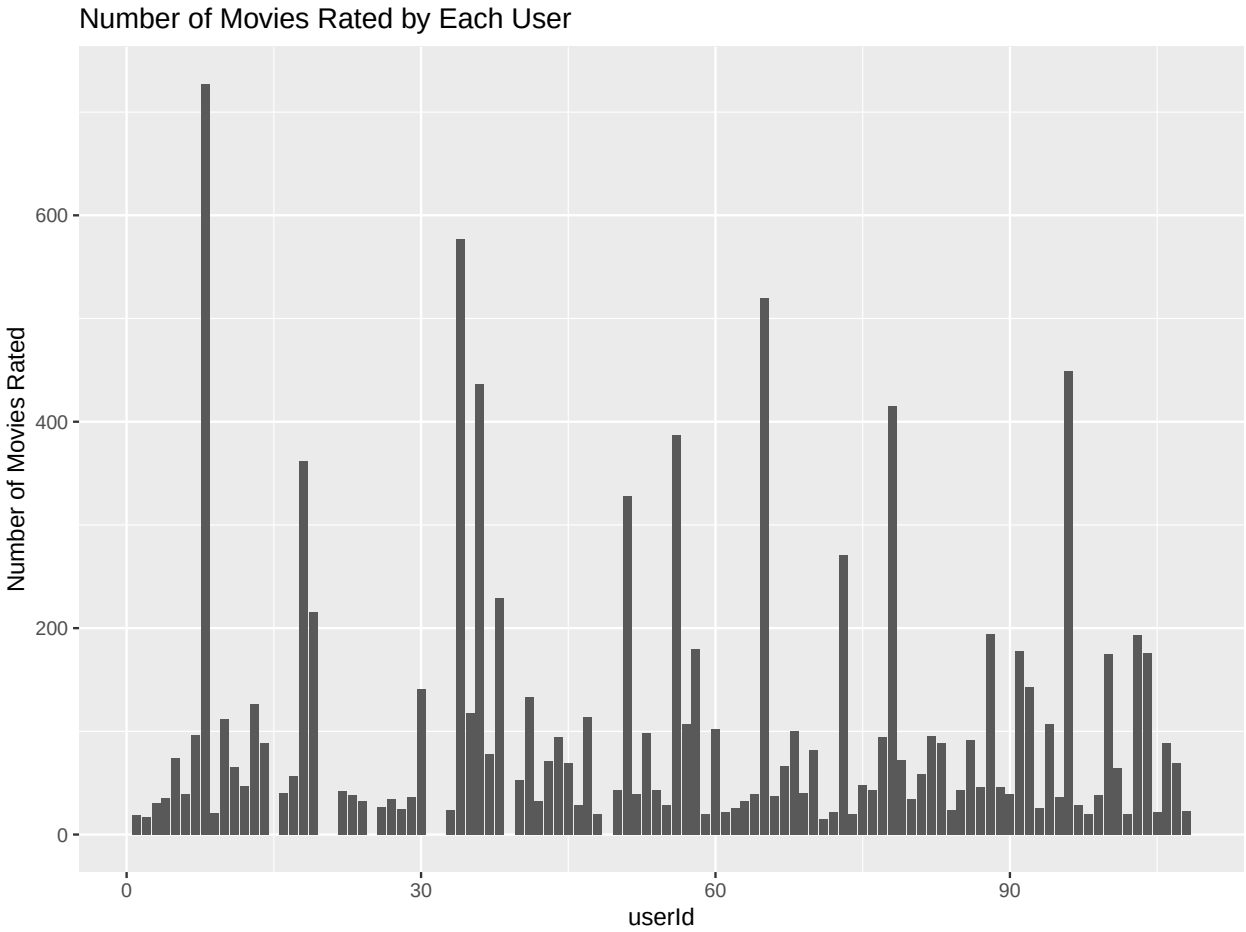
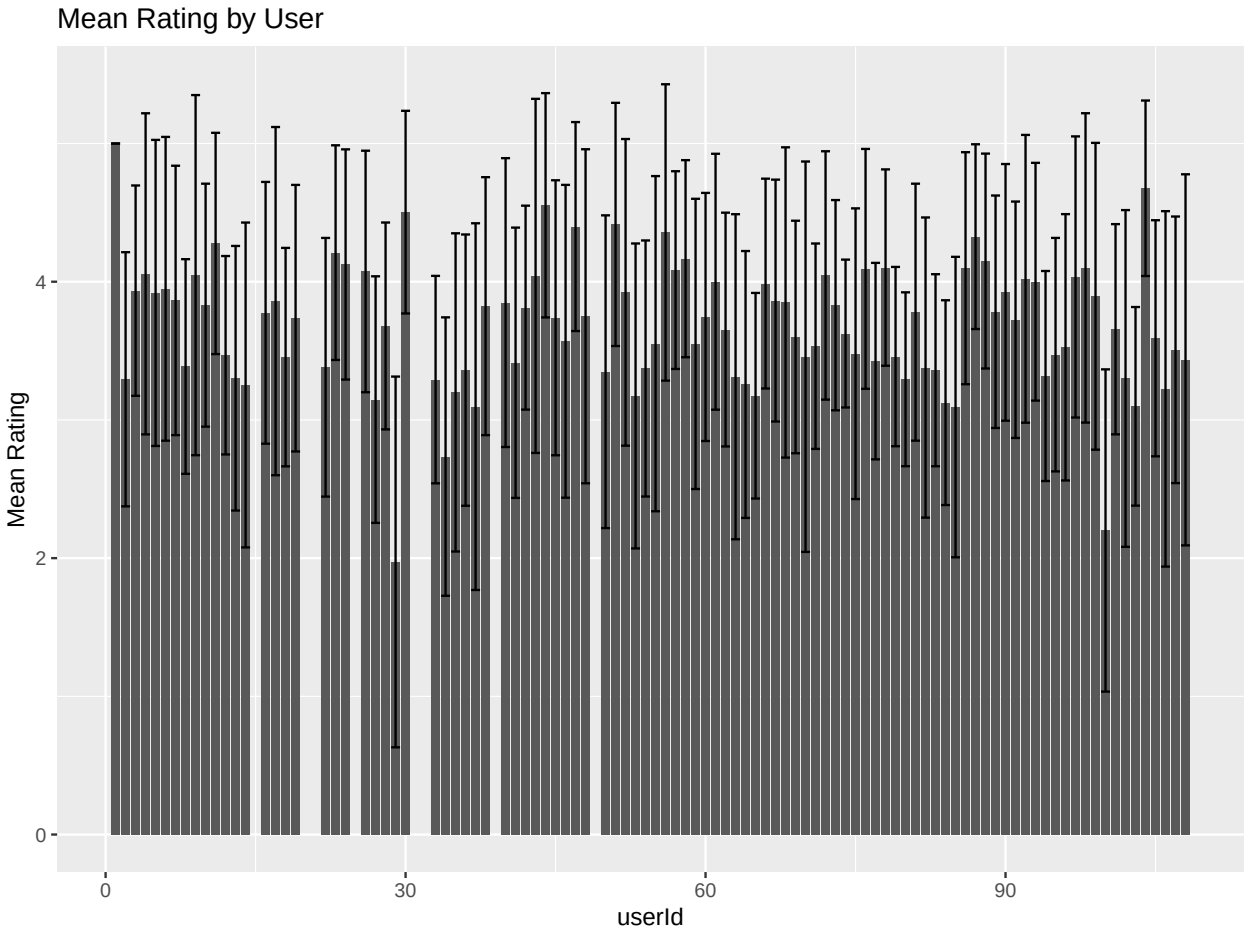



Figure 1: Number of Movies rated by first 100 userIds.

In figure 2, we can see the change in mean rating, and the standard deviation (as errorbars) in ratings, for the first 100 userIds.




```
## Warning in rm(user_counts, user_ratings): object 'user_ratings' not found
```

Figure 2: Mean rating by first 100 userIds.

### Movie Effects

Movie ratings will also be impacted by the movie itself, as some movies are better than others. Table 2 shows the mean rating and standard deviation for the first 6 movieIds.

```
## # A tibble: 6 x 4
##   movieId Movies_Rated Mean_Rating Standard_Deviation
##     <dbl>        <int>       <dbl>              <dbl>
## 1       1        23790        3.93              0.897
## 2       2        10779        3.21              0.951
## 3       3         7028        3.15              0.999
## 4       4         1577        2.86              1.09
## 5       5         6400        3.07              0.964
## 6       6        12346        3.82              0.885
```

Table 2: Mean rating and standard deviation for the first 6 movieIds

Figure 3 visualizes the number of ratings for the first 100 movieIds

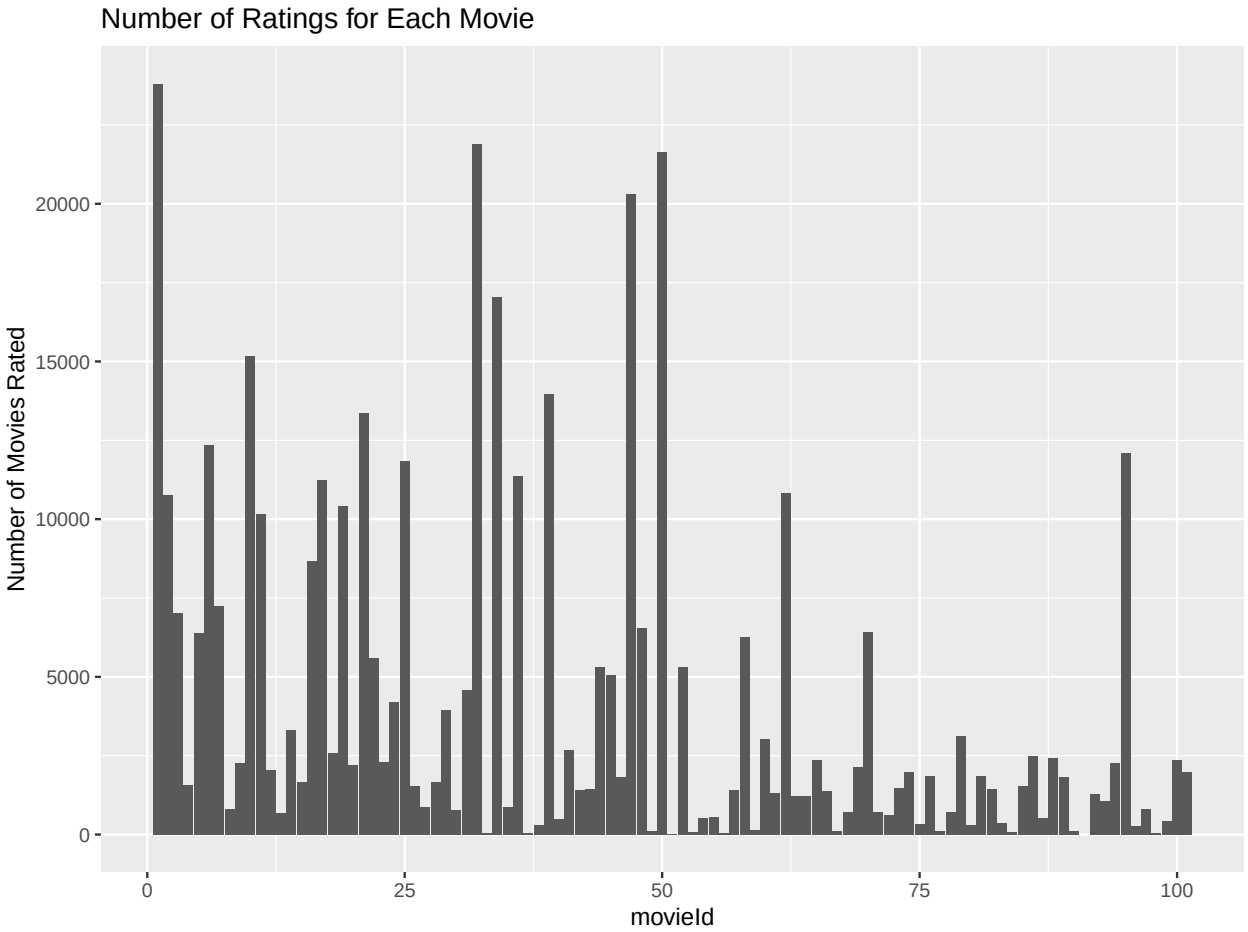
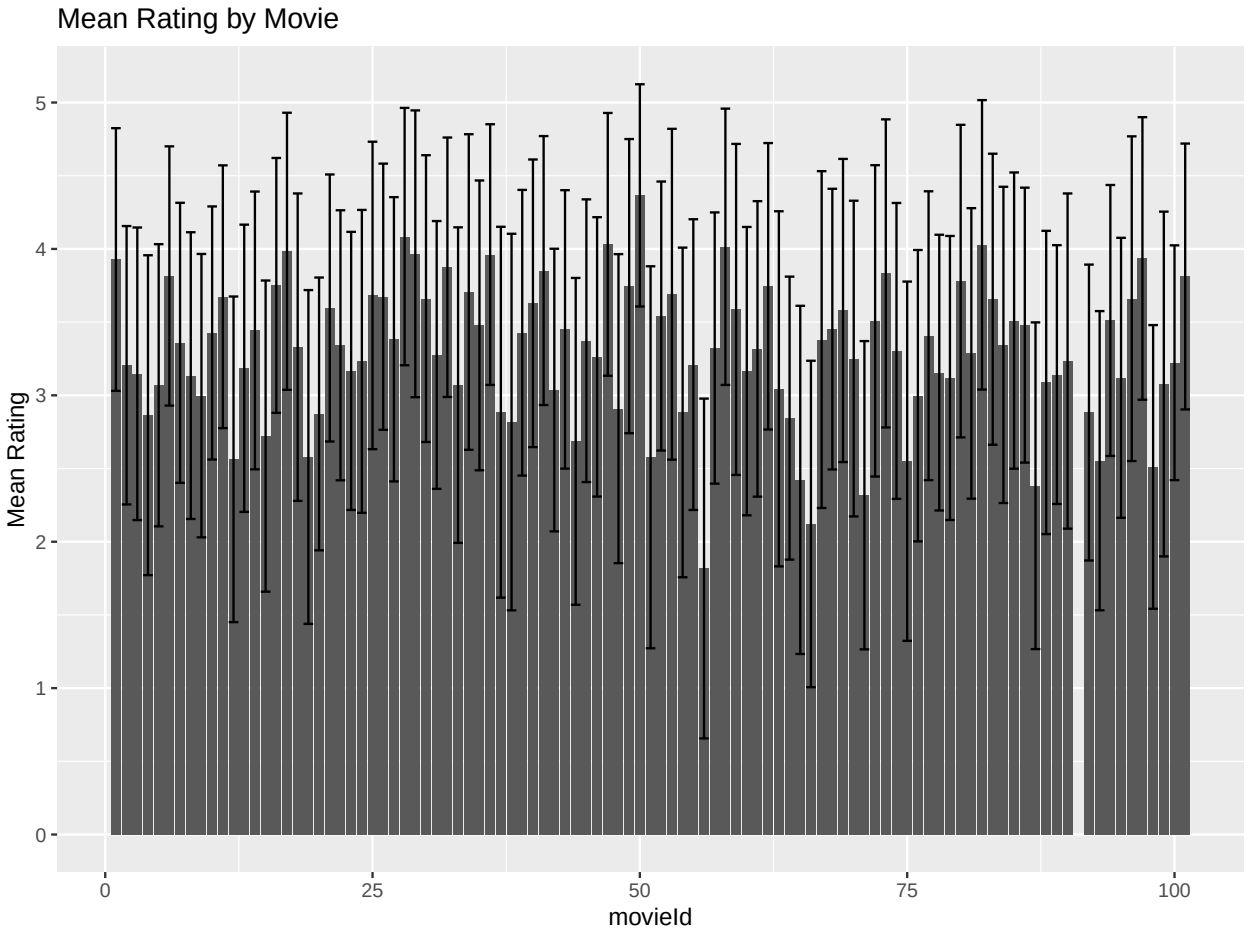



Figure 3: Number of Movies rated by first 100 movieIds

In figure 4, we can see the change in mean rating, and the standard deviation (as errorbars) in ratings, for the first 100 movieIds




```
## Warning in rm(movie_counts, movie_ratings): object 'movie_ratings' not found
```

Figure 4: Mean rating by first 100 movieIds

### Genre Relationships

Movie ratings may be impacted by the genre of the movies. However, each movie may belong to multiple genres. To see if there was a correlation between the genres, I did a Pearson-r correlation on the genre matrix. The correlations are visualized in Figure 5.

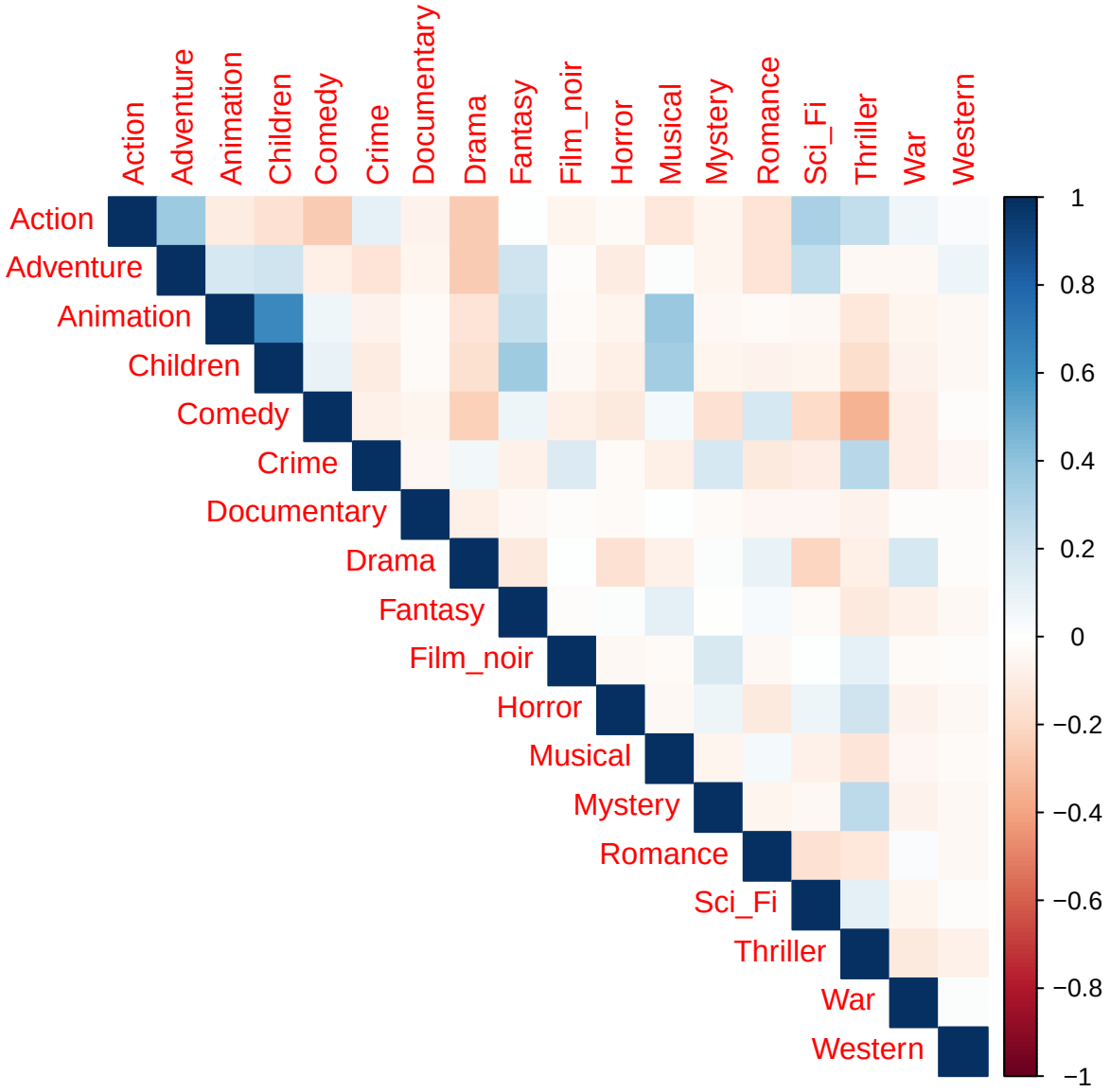



Figure 5: Significance of correlation between the genres

As we can see, the correlations are quite weak.

For correlation lower than -0.1975513 (mean-sd), we have Comedy-Drama, Horror-Drama, and Comedy-Thriller, meaning these genres overlap the least.

For correlation greater than 0.2929194 (mean + sd), we have Animation-Children, meaning these genres overlap the most.

**Genre Effects**

Given the correlation of genres with ratings, movie ratings will also be impacted by the movie's genres. Table 3 shows the mean rating and standard deviation for each genre. Plotting correlation between ratings and genres, we see no major interactions (Figure 6) - none of the correlation coefficients exceed 0.2 in either direction. This suggests that individual genres do not have an impact on ratings.

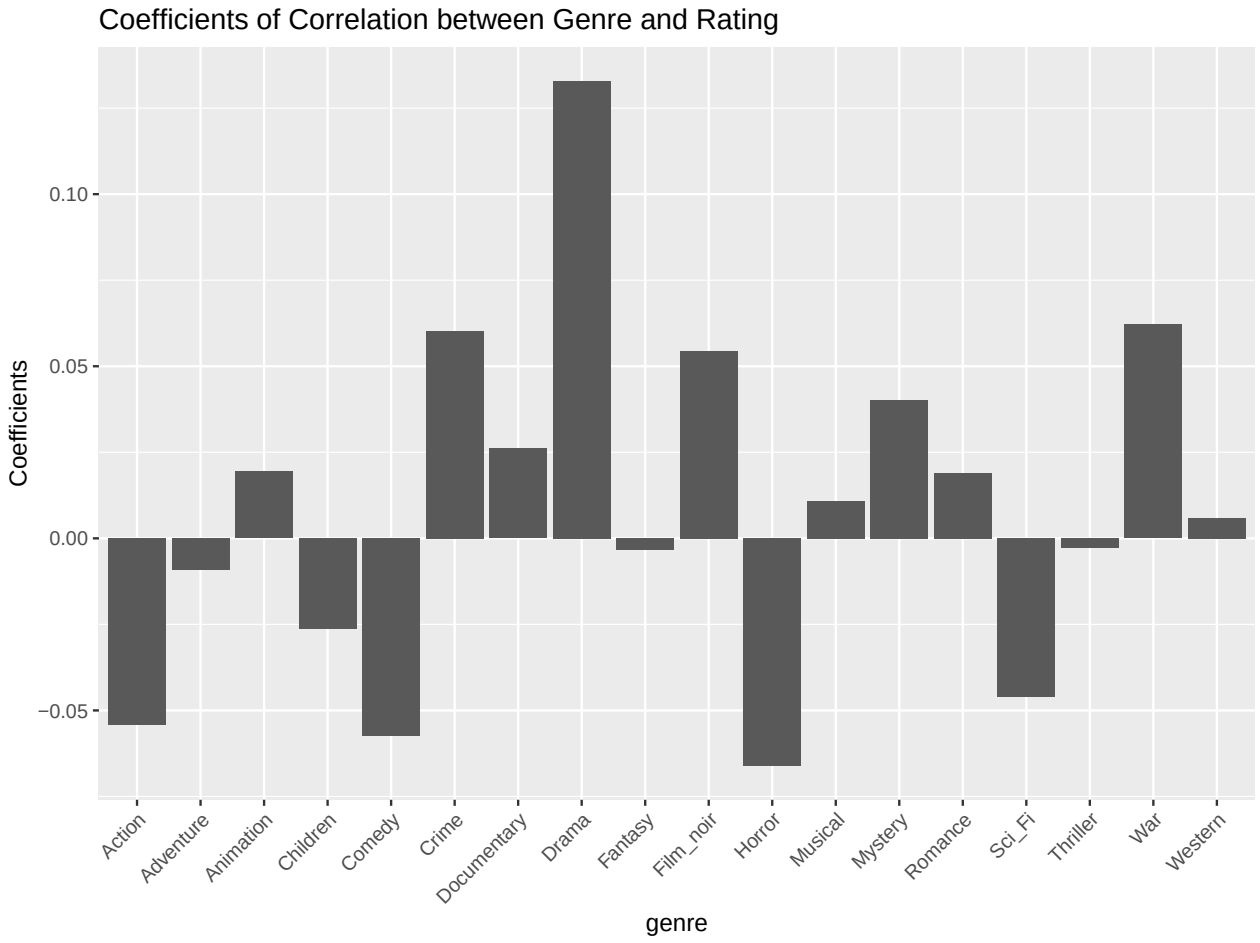



Figure 6: Correlation between genres and ratings.

And plotting mean ratings for genre, we see bias by genres as well.

```
## # A tibble: 18 x 4
##    genre       Movies_Rated Mean_Rating Standard_Deviation
##    <chr>              <int>       <dbl>              <dbl>
##  1 Action           2560545        3.42               1.07
##  2 Adventure        1908892        3.49               1.05
##  3 Animation         467168        3.60               1.02
##  4 Children          737994        3.42               1.09
##  5 Comedy           3540930        3.44               1.07
##  6 Crime            1327715        3.67               1.01
##  7 Documentary        93066        3.78               1.00
##  8 Drama            3910127        3.67               0.995
##  9 Fantasy           925637        3.50               1.07
## 10 Film_noir         118541        4.01               0.887
## 11 Horror            691485        3.27               1.15
## 12 Musical           433080        3.56               1.06
## 13 Mystery           568332        3.68               1.00
## 14 Romance          1712100        3.55               1.03
## 15 Sci_Fi           1341183        3.40               1.09
## 16 Thriller         2325899        3.51               1.03
## 17 War               511147        3.78               1.01
## 18 Western           189394        3.56               1.02
```

Table 3: Mean rating and standard deviation for each genre.

Figure 7 visualizes the number of movies in each genre.

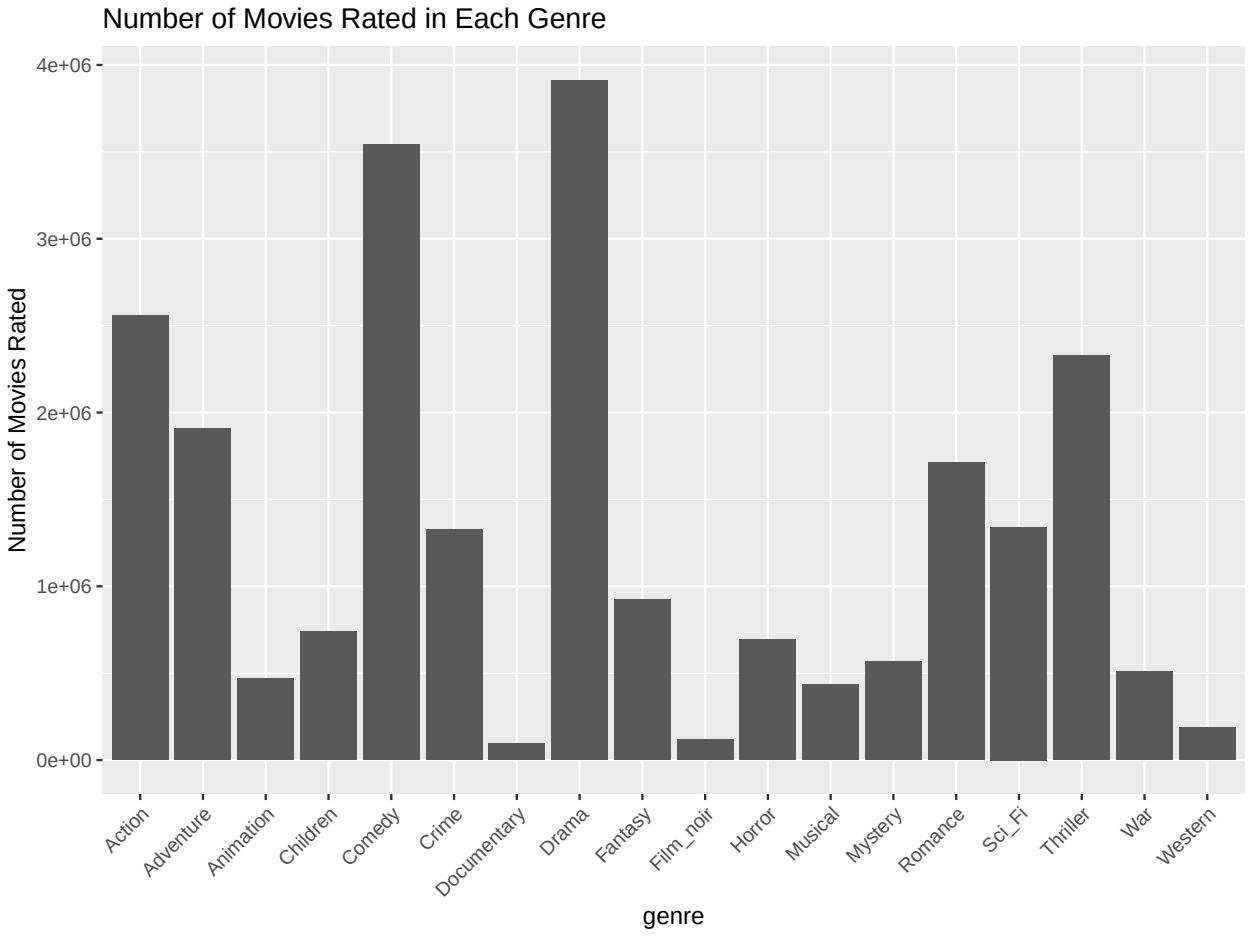



Figure 7: Number of Movies rated in each genre.

In figure 8, we can see the change in mean rating and the standard deviation (as errorbars) in ratings for all genres.

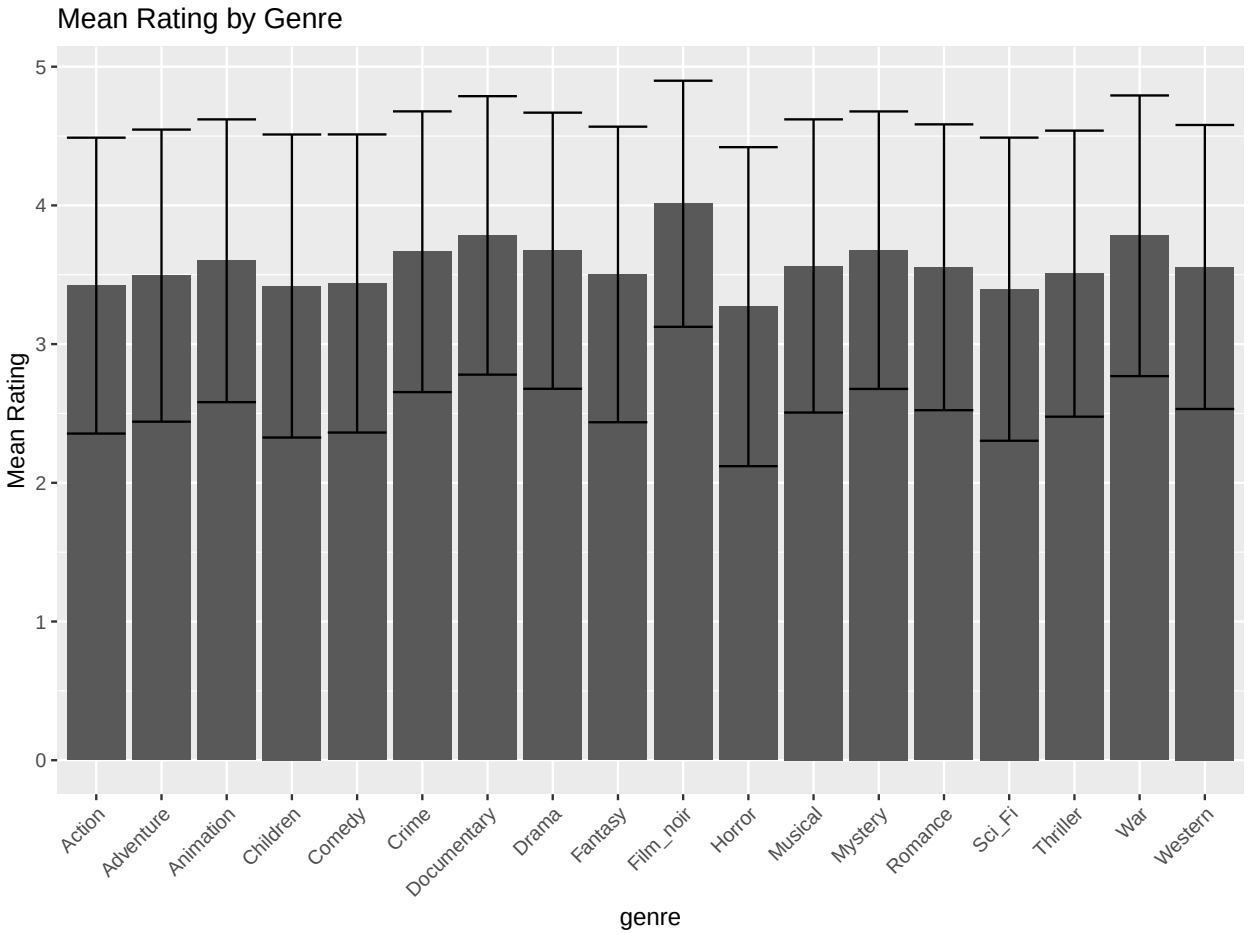



Figure 8: Mean rating and standard deviation for all genres.

### Genre Combination

However, the combination of genres a movie falls into may be more important than the discrete genres it is part of. Figure 9 visualizes the mean rating by genre combination for 50 top-rated genre combinations.

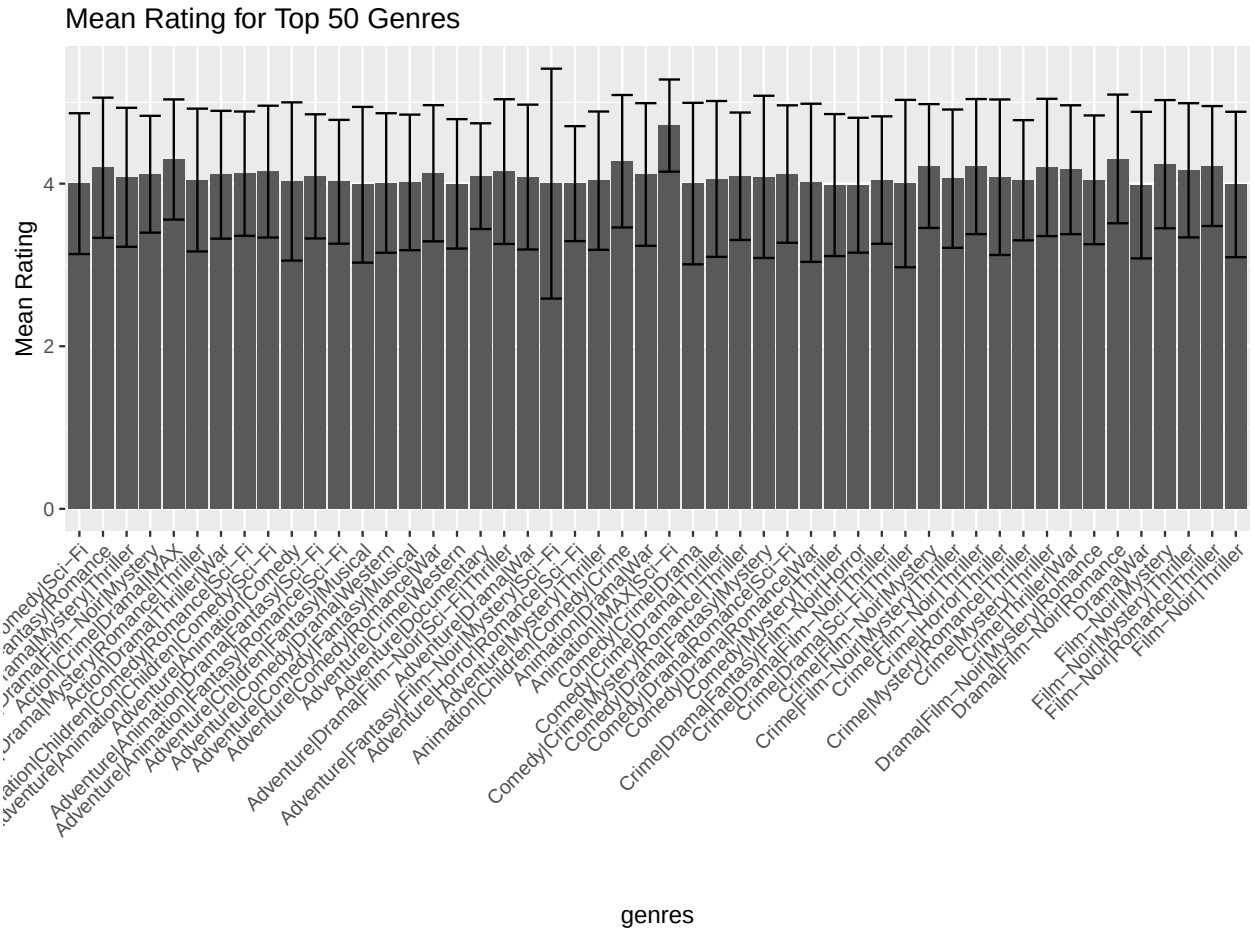



Figure 9: Mean Rating by genre combination for 50 top-rated genre combinations.

In Figure 10 we see the distribution for the 50 least rated genre combinations

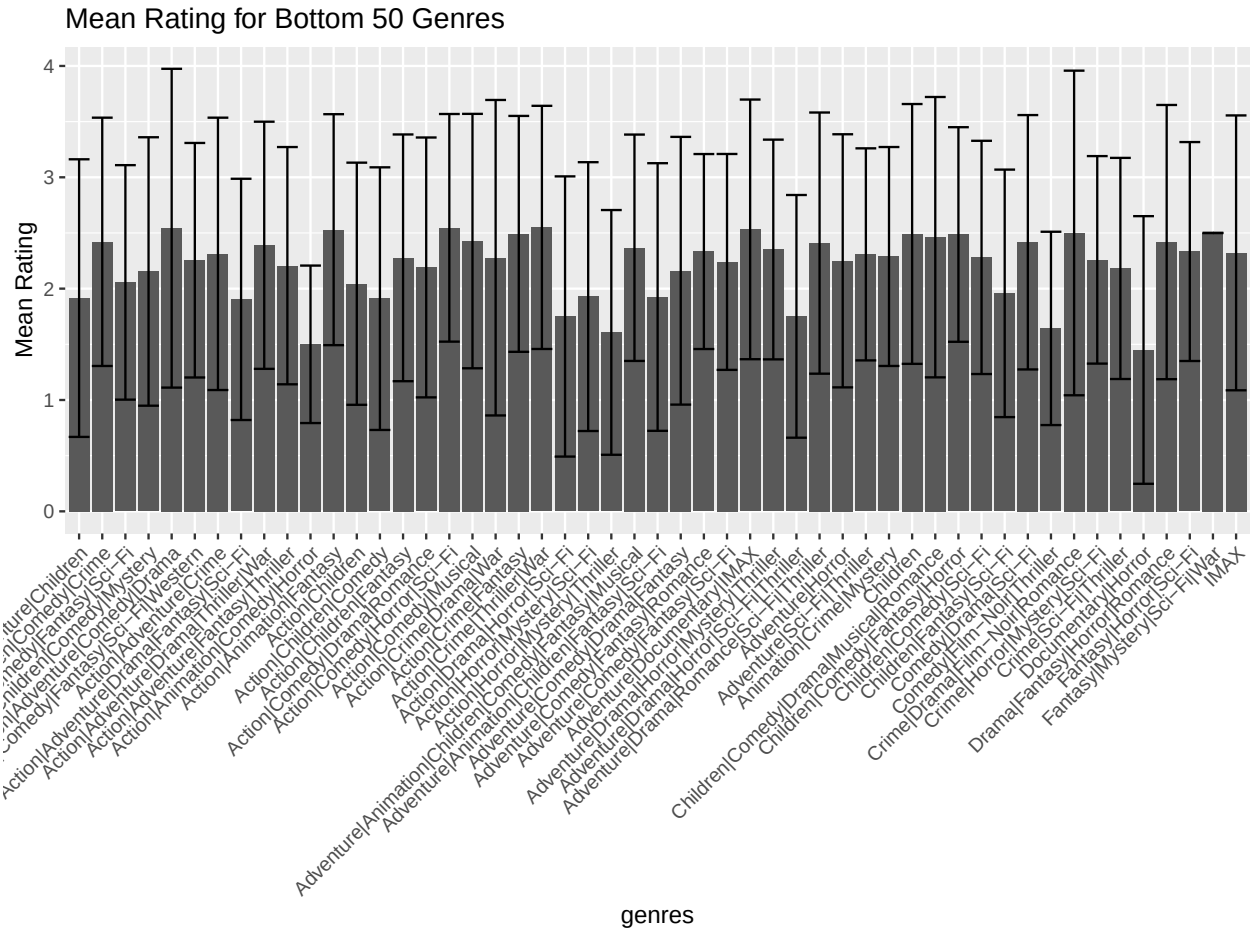



Figure 10: Mean Rating by genre combination for 50 least-rated genre combinations.

Correlating the genre combinations with ratings, we see

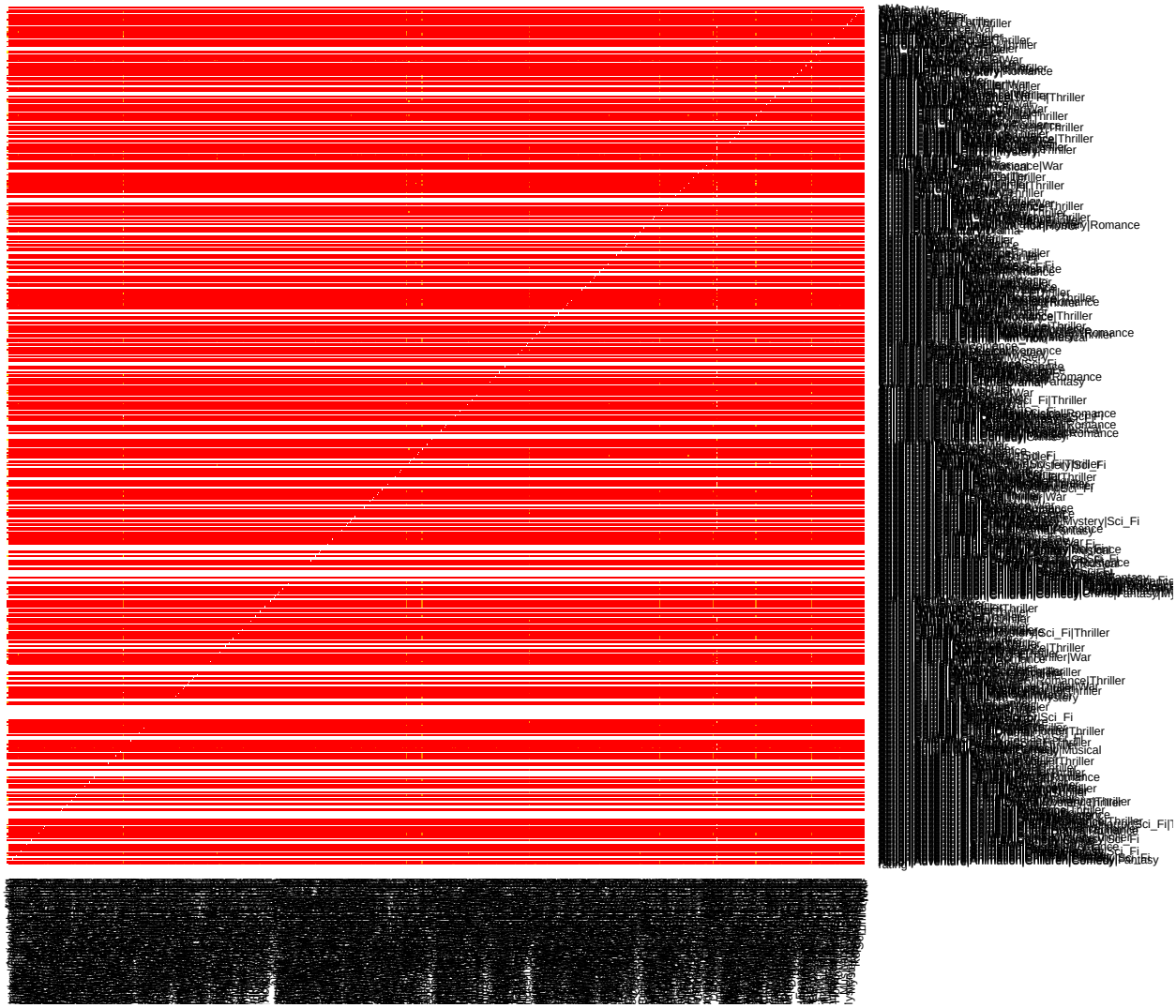


Figure 11: Correlation of Genre Combination with Ratings.

Of the 797 unique genre combinations, 588 of them were significantly correlating with ratings. This suggests a much stronger relationship between genre combinations and rating. Therefore, I will use genre bias from the genre combinations in the regression.

### Year of Release Effects

The year of a movie's release may affect ratings as well. Old movies may be rated more highly because they are considered classics. On the other hand, there will also be fewer movies in the early years, which could lead to the ratings having higher variance. Table 4 shows the mean rating and standard deviation for the first 10 movieIds.

```
## # A tibble: 6 x 4
##    year Number_of_Ratings Mean_Rating SD_Rating
##   <dbl>             <int>       <dbl>     <dbl>
## 1  1915               180        3.29     1.19
## 2  1916                84        3.83     0.916
## 3  1917                32        3.73     0.959
## 4  1918                73        3.65     1.17
## 5  1919               158        3.28     1.09
```

```
## 6  1920                575        3.94      0.843
```

Table 4: Mean rating and standard deviation for movies by Year

Figure 12 visualizes the number of movies rated for each year of release.

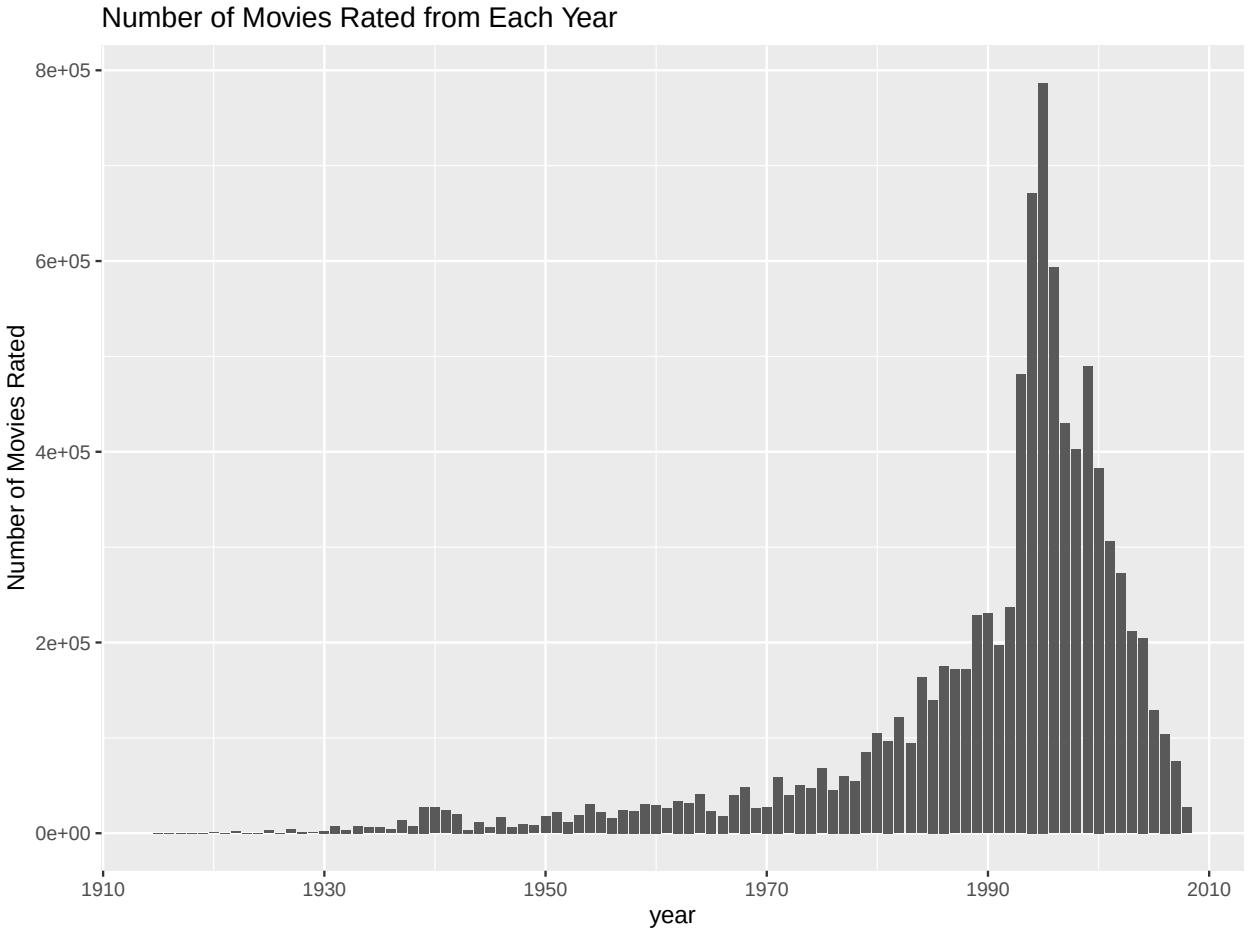



Figure 12: Number of Movies rated for each year of release.

In figure 13, we can see the change in mean rating, and the standard deviation (as errorbars) in ratings, for each year of rellease.

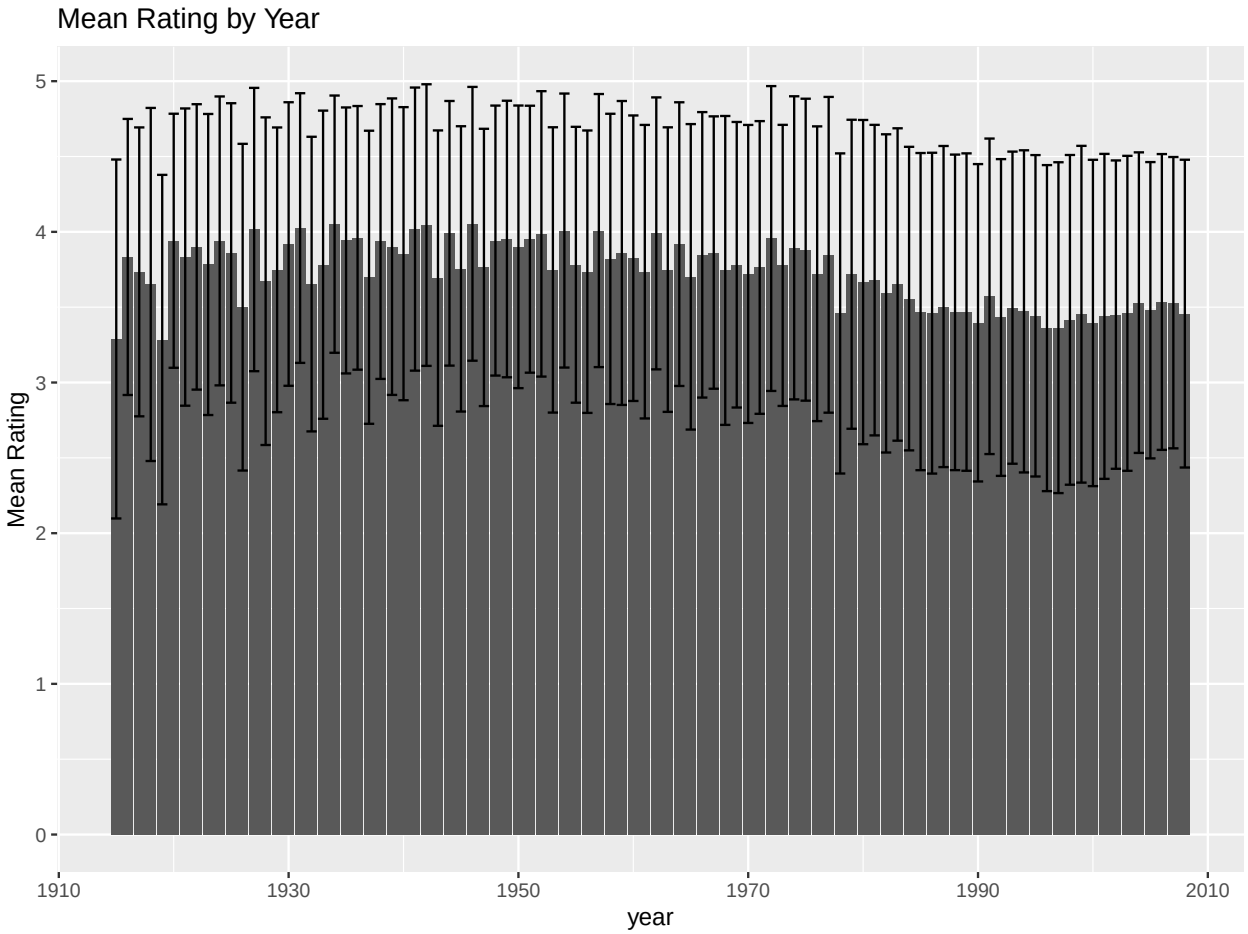



Figure 13: Mean rating by first 1000 userIds.

**Time of Rating Effects**

Are ratings affected by the year of rating? Table 5 shows the mean rating and standard deviation for the first 6 years.

```
## # A tibble: 6 x 4
##   rating_datetime Movies_Rated Mean_Rating SD_Rating
##             <dbl>        <int>       <dbl>     <dbl>
## 1            1995            2        4        1.41
## 2            1996       942772        3.55     0.995
## 3            1997       414101        3.59     1.01
## 4            1998       181634        3.51     1.13
## 5            1999       709893        3.62     1.12
## 6            2000      1144349        3.58     1.12
```

Table 5: Mean rating and standard deviation for the first 6 years of rating

Figure 14 visualizes the number of ratings for each year of rating.

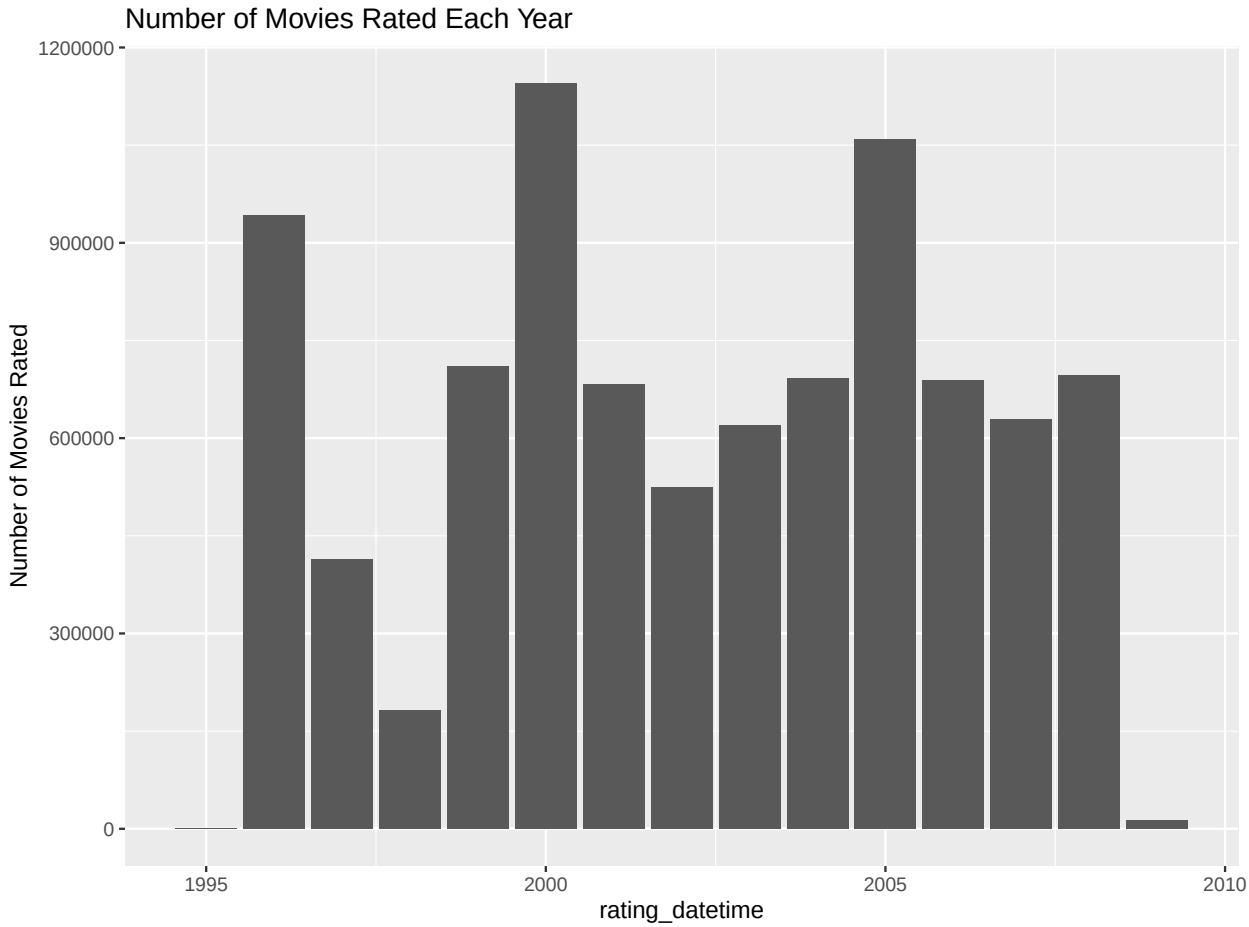



Figure 14: Number of Movies rated Year of Rating.

In figure 15, we can see the change in mean rating, and the standard deviation (as errorbars) in ratings, for each year of rating.

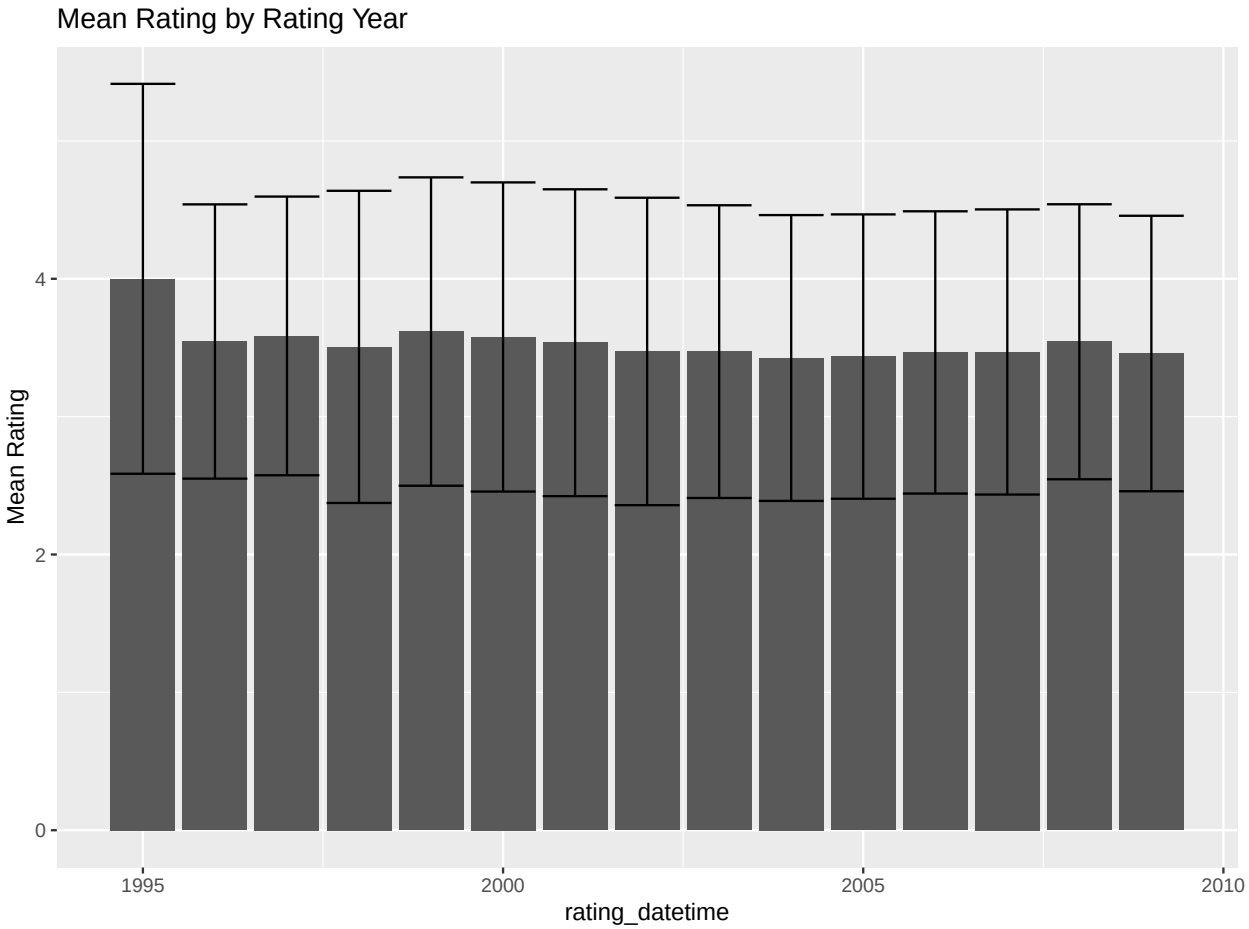



Figure 15: Mean rating by Year of Rating.

The figures who little impact of year. Digging in deeper, I created another variable - distance_from_release - that calculated how long after the release of a movie was it rated by a user. Plotting the mean and standard deviation for this, we see similar plateauing (Figure 16).

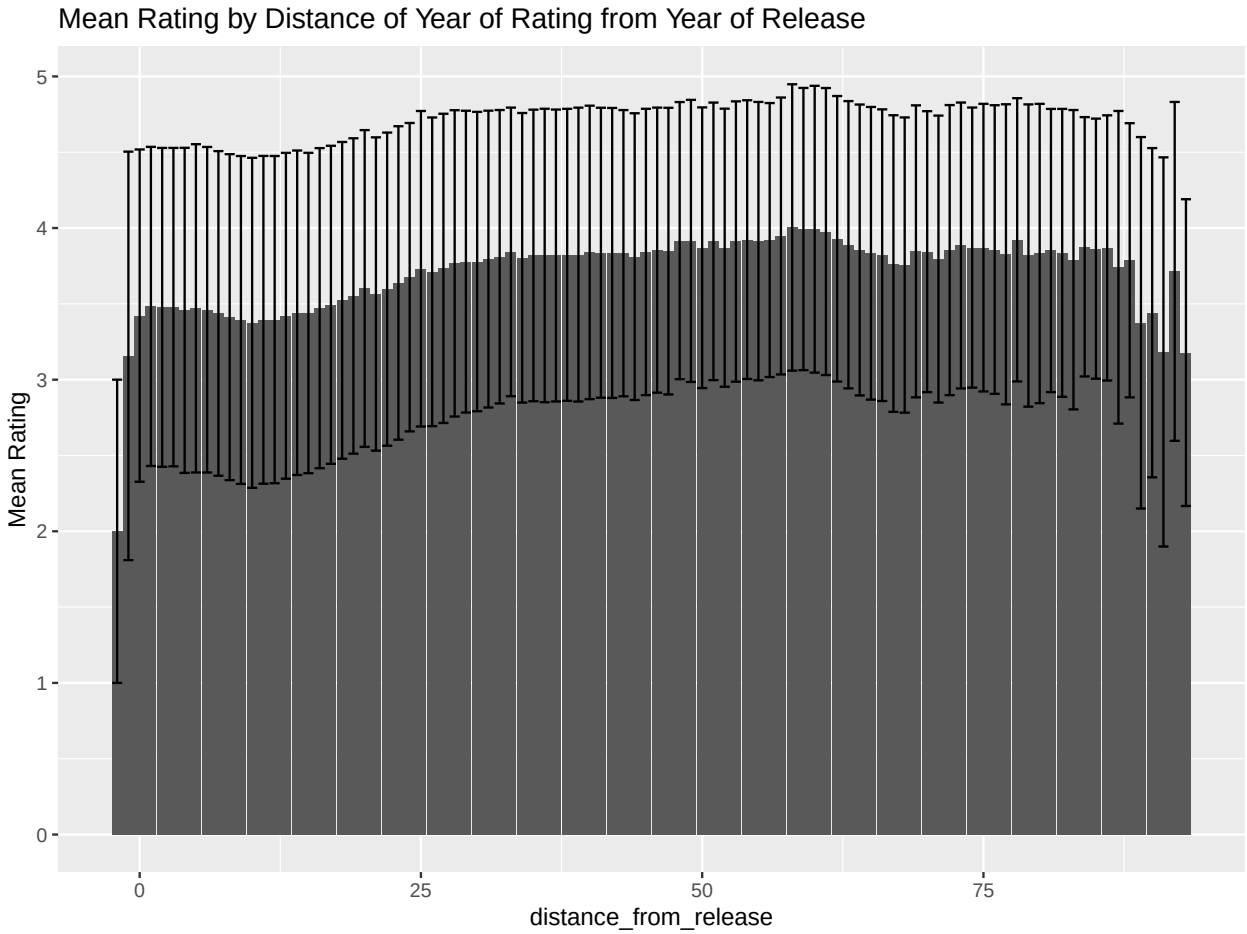



Figure 15: Mean rating by Distance of Year of Rating from Year of Release.

## Results

My analysis plan was to test the regression with each variable, and then to combine the most successful variables. I then planned to run this combination model with standardization and regularization.

### Regressions Accounting for Movie Bias

My first variable was the movie itself - we know some movies are better than others, so it would follow that ratings for movies would be biased by the movie.

The resulting model had an RMSE of 0.9439087.

### Regressions Accounting for User Bias

My next variable was the user themselves - we know that we have different tastes in movies, and are therefore likely to rate certain movies better than others.

The resulting model had an RMSE of 0.978336.

### Regressions Accounting for Genre Bias

### Regressions Accounting for Year Bias

The resulting model had an RMSE of 1.0500259.

**Regressions Accounting for Time Bias**

The resulting model had an RMSE of 1.0594609.

**Regressions Accounting for Distance between Year of Release and Year of Rating Bias**

The resulting model had an RMSE of 1.0523932.

The best models were the ones that took into account movie bias, user bias, and genre bias.

**Regressions Combining Movie and User Bias**

Of all the variables that impact ratings, movie bias and user bias are the most essential. Therefore, I added these to our regression model

The resulting model had an RMSE of 0.8653488.

**Regressions Combining Movie, User and Genre Bias**

Adding genre to the mix, we see the RMSE worsen to 0.8649469.

**Regressions Combining Movie, User and Year of Release Bias**

So we took out genre and added year of release to the mix, and we saw the RMSE better to 0.8650043.

**Regressions Combining Movie, User and Time of Rating Bias**

So we took out genre and added year of release to the mix, and we saw the RMSE better to 0.8653369.

**Regressions Combining Movie, User and Difference between Time of Rating and Year of Release Bias**

This was the most successful model so far, and the RMSE bettered to 0.8649038. Therefore I decided to use only this model for standardization and regularization tests.

**Regressions with Standardization**

What happens if we standardize ratings within users and movies, such that ratings by each user fell on a normal curve, as did ratings for each movie?

The RMSE with standardization of ratings was quite high. One explanation can be that standardization is not helpful for predicting real ratings because real ratings are not standardized. However, for modeling movie recommendations (i.e., categorical recommendations), standardization may be helpful.

**Regressions with Regularization**

We then checked the RMSE with regularization. When keeping year in the mix, the RMSE did not improve with regularization (Figure 16).

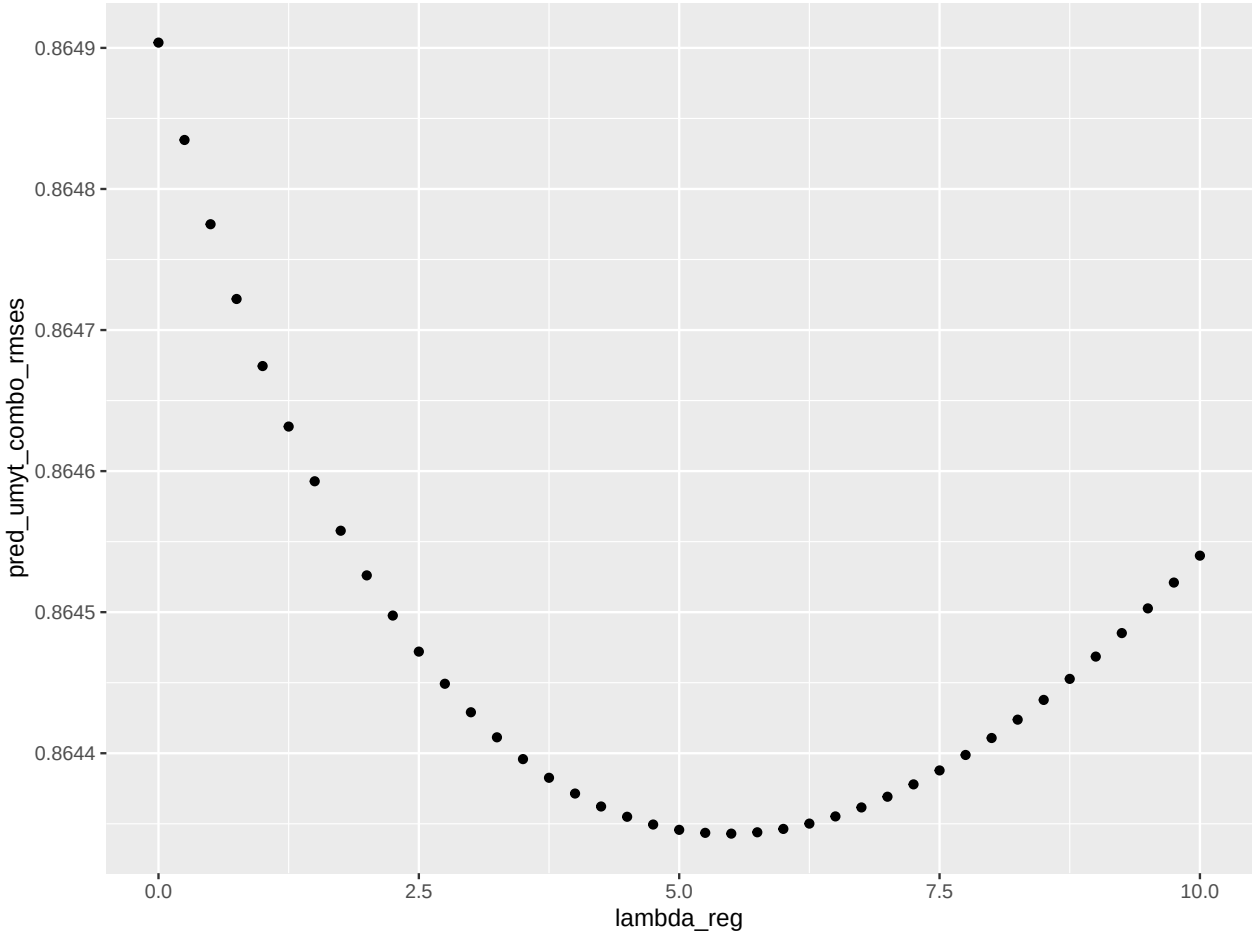



Figure 16: RMSE at different lambdas for the User-Movie-Distance between Year of Rating and Release Model.

```
r_error = pred_umyt_combo_rmses[which.min(pred_umyt_combo_rmses)]
r_error_lambda = lambda_reg[which.min(pred_umyt_combo_rmses)]

rm(pred_umyt_combo_rmses)
```

The best RMSE was 0.864343 at a lambda value of 5.5. Therefore, regularization improved the combined user-movie-distance-from-release model's predictive ability.

### RMSE Table

All RMSEs are listed in table 6.

```
##                                     model      RMSE
## 13                          Regularization 0.8643430
## 11 User-Movie-Distance from Release Combo 0.8649038
## 8                   User-Movie-Genre Combo 0.8649469
## 9         User-Movie-Year of Release Combo 0.8650043
## 10         User-Movie-Time of Rating Combo 0.8653369
## 7                         User-Movie Combo 0.8653488
## 12                         Standardization 0.9058226
## 1                               Movie Bias 0.9439087
## 2                                User Bias 0.9783360
## 3                               Genre Bias 1.0184056
```

```
## 5                                Year Bias 1.0500259
## 6  Year of Release vs Time of Rating bias 1.0523932
## 4                                Time Bias 1.0594609
```

Table 6: RMSEs across all models from lowest RMSE to highest RMSE.

## Conclusion

I conclude that movie, user and distance from release are the best predictors of ratings. By predicting ratings, we can effectively and successfully predict target audience of a movie as well as movie recommendations for a viewer.