MovieLens Project Submission

Craig Haile

August 12, 2019

In partial fulfillment of the requirements of HarvardX: PH125.9x Data Science Capstone

Introduction/Overview/Executive Summary

This purpose of this project is to create a recommendation model to predict movie ratings using the 10M version of the MovieLens dataset. The data contains information related to the movie reviewer, date of review, movie title and year released, and genre(s). The data has been divided into a training set comprised of approximately 90% of the dataset (`edx`) and test set of around 10% of the data (`validation`).

After some initial exploration and visualization, several models are built of increasing complexity. The models are evaluated based on the Root Mean Square Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum(\hat{y} - y)}$$

where $\hat{y}$ is the predicted movie rating and y is the actual movie rating over all N possible combinations of movies and reviewers. The goal is to develop a model with RMSE ≤ 0.8649. The final model developed "Regularized Movie+User+ReleaseYear+Genre+RateYear model" achieved an RMSE of 0.8643.

Methods and Exploratory Data Analysis

Inital data Exploration, Cleaning, and Pre-processing

We will take a brief view of the `edx` dataset structure, which will be the same as `validation`.

```
##   userId movieId rating timestamp                          title                        genres
## 1      1     122      5 838985046               Boomerang (1992)                Comedy|Romance
## 2      1     185      5 838983525                Net, The (1995)         Action|Crime|Thriller
## 4      1     292      5 838983421                Outbreak (1995)  Action|Drama|Sci-Fi|Thriller
## 5      1     316      5 838983392                Stargate (1994)       Action|Adventure|Sci-Fi
## 6      1     329      5 838983392 Star Trek: Generations (1994) Action|Adventure|Drama|Sci-Fi
## 7      1     355      5 838984474       Flintstones, The (1994)       Children|Comedy|Fantasy
```

There are six variables present in both datasets:

1. **userId** is a number that designates each individual movie reviewer.
2. **movieId** identifies each individual movie, since there may be distinct movies with the same title.
3. **rating** shows the rating of the movie by an individual user. Ratings are given from 0.5 to 5 in increments of 0.5.
4. **timestamp** contains the timestamp for the rating provided by a particular user.
5. **title** is the title of each movie including release year.
6. **genres** shows the genre(s) of the movie, with multiple genres separated by `|`.

We will next check for missing values in `edx` and `validation`.

Missing value check for edx dataset.

userId	movieId	rating	timestamp	title	genres
0	0	0	0	0	0

Missing value check for validation dataset.

Table 2: NA check for validation

userId	movieId	rating	timestamp	title	genres
0	0	0	0	0	0

We see there are no missing values.

Next we will alter timestamp to more understandable date, and reduce to the year only to avoid too much granularity. We will save this as the new variable (`yr_rate`). Additionally, we will extract the year released from the movie title and save as the variable `yr_release`. Since the `genres` string is a somewhat complex string with multiple genres separated by the pipe `|`, we will simplify (and perhaps oversimplify) by extracting the first alphaebetically listed genre and save as `genre_single`. Finally we will remove columns that will not be used for our predictive models to help speed processing time.

Reduced dataset format

```
##   userId movieId rating yr_rate genre_single yr_release
## 1      1     122      5    1996       Comedy       1992
## 2      1     185      5    1996       Action       1995
## 3      1     292      5    1996       Action       1995
## 4      1     316      5    1996       Action       1994
## 5      1     329      5    1996       Action       1994
## 6      1     355      5    1996     Children       1994
```

Descriptives and Visualizations

We will now look at some basic descriptives and visualizations of the `edx` dataset.

The number of ratings given in `edx`, grouped by rating.

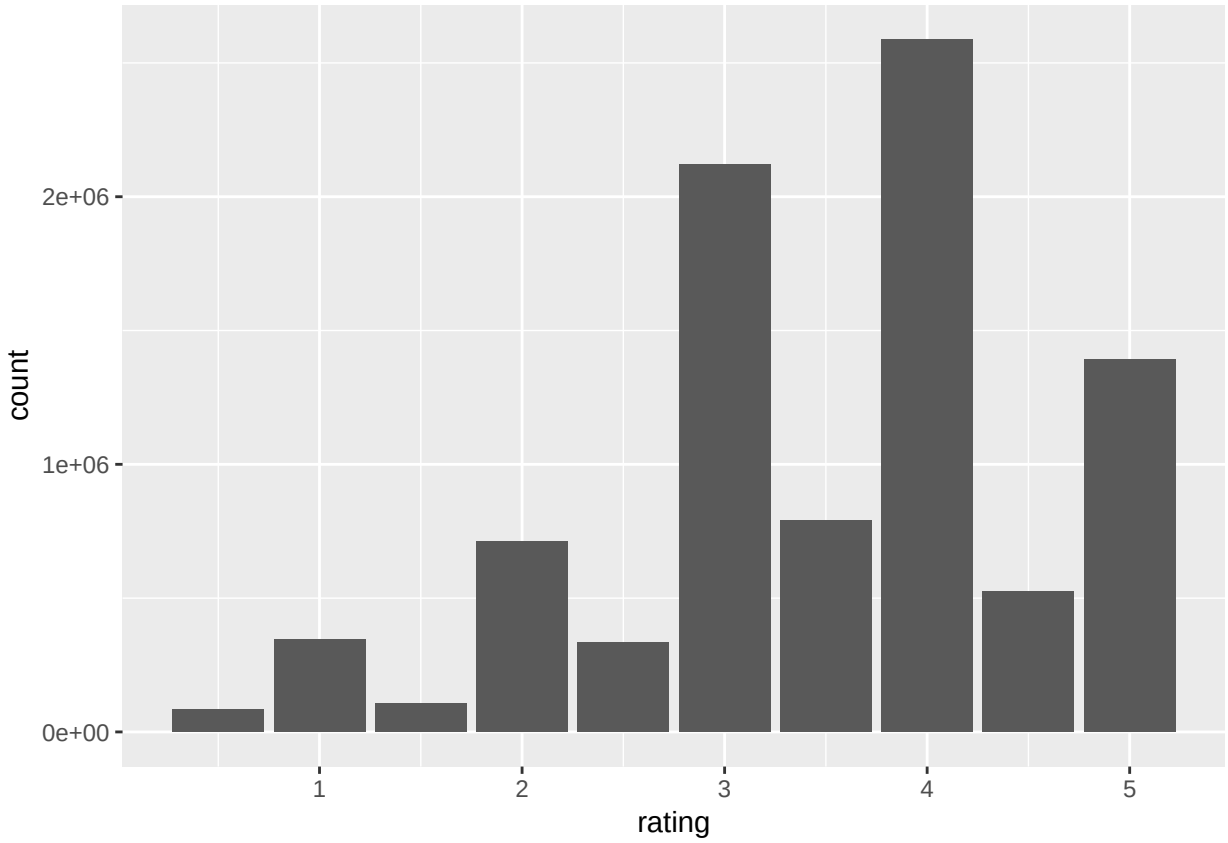
```
## # A tibble: 10 x 2
##    rating       n
##     <dbl>   <int>
##  1    0.5   85374
##  2    1    345679
##  3    1.5  106426
##  4    2    711422
##  5    2.5  333010
##  6    3   2121240
##  7    3.5  791624
```

```
##  8    4   2588430
##  9    4.5  526736
## 10    5   1390114
```

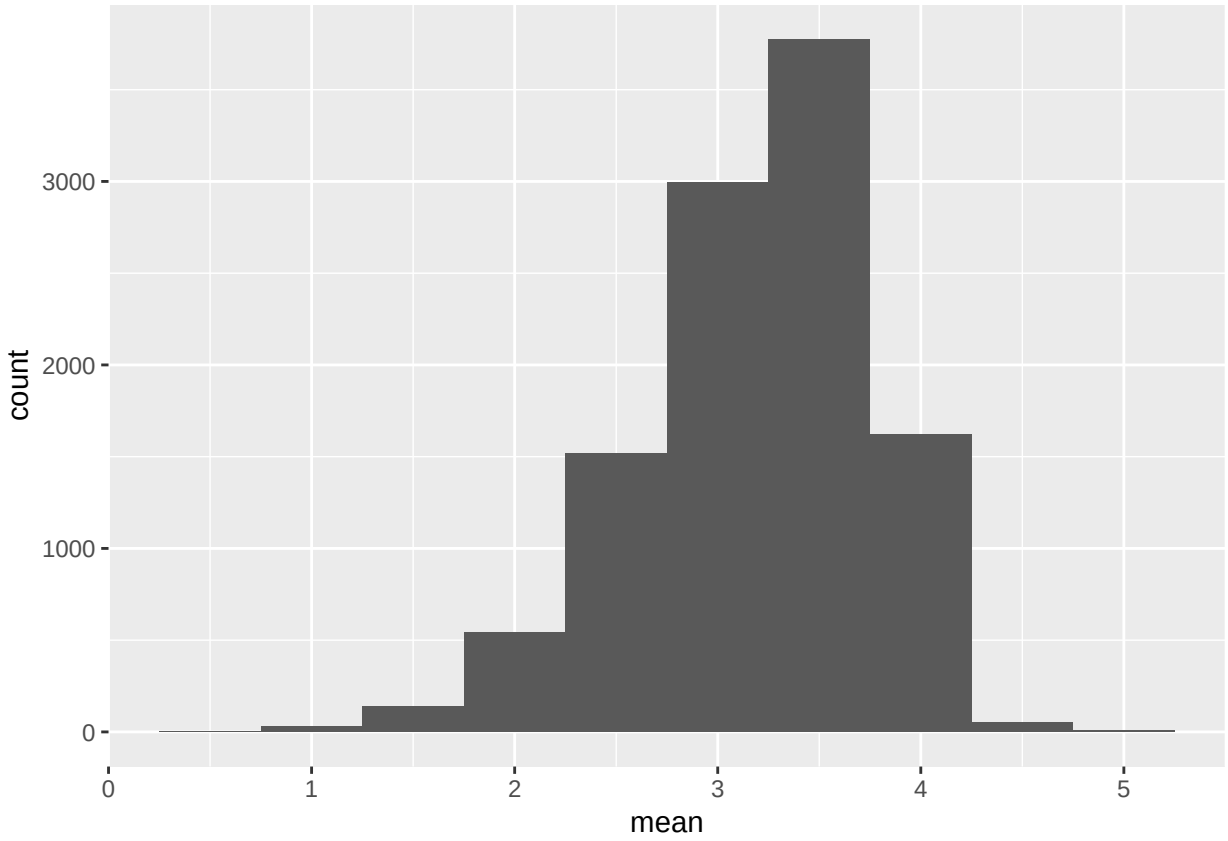
Table showing number of distinct movies and raters

movies	raters
10677	69878

Bar chart of ratings count

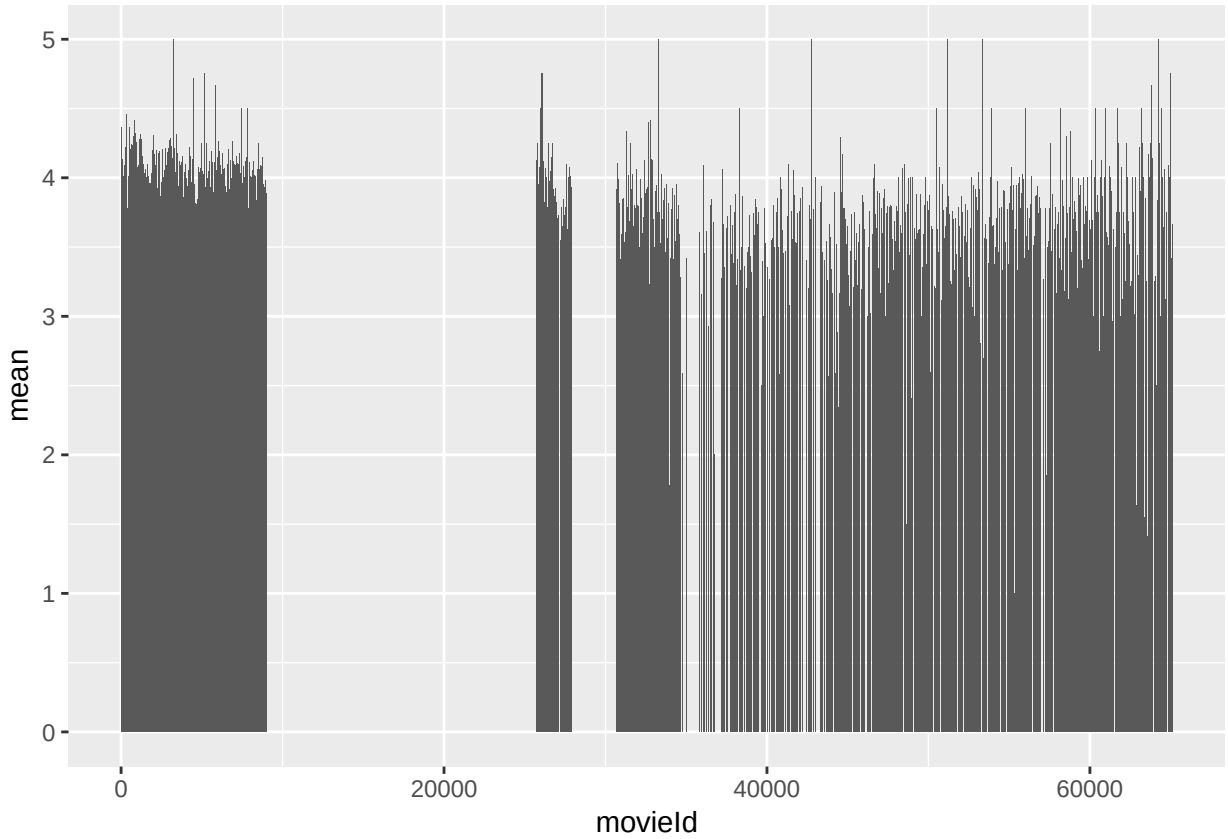


Histogram of frequency of mean ratings of movies

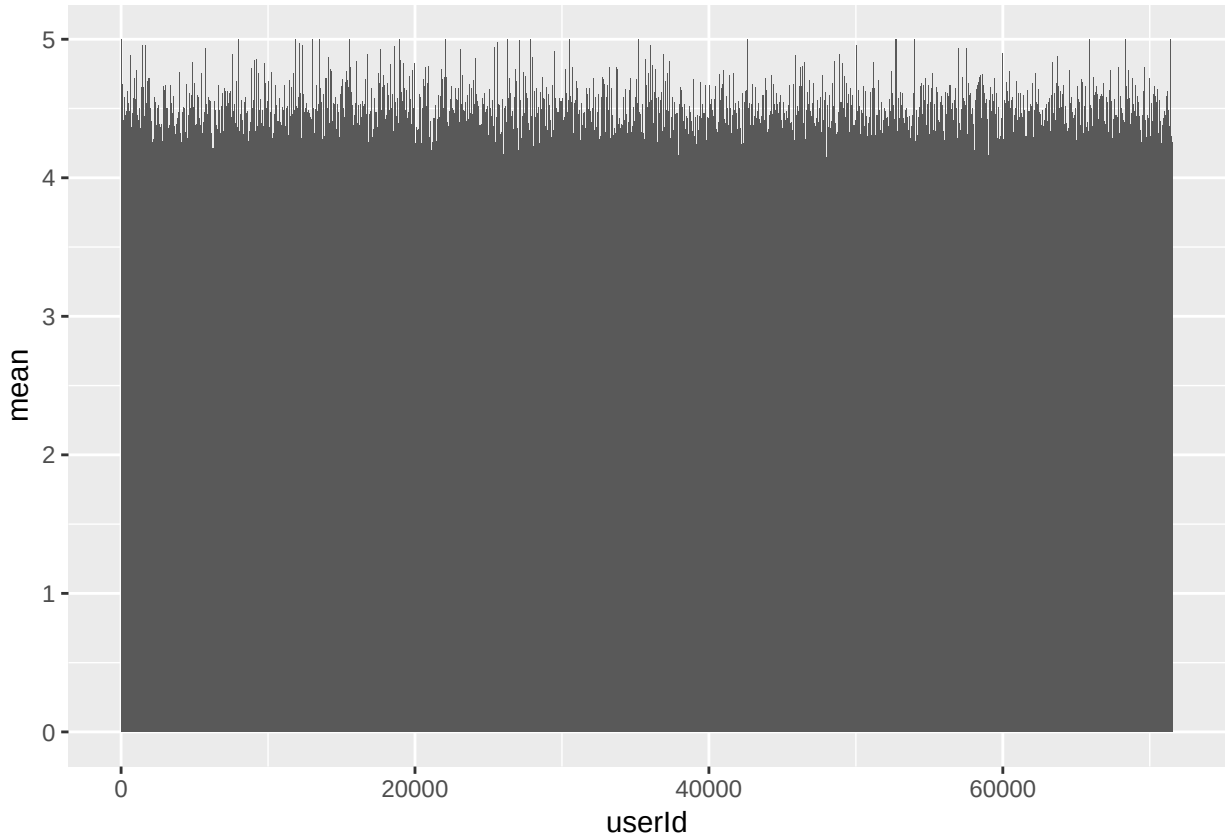


Next we will look at the potential effects of different predictors on the mean movie rating to see which we might want in our model.

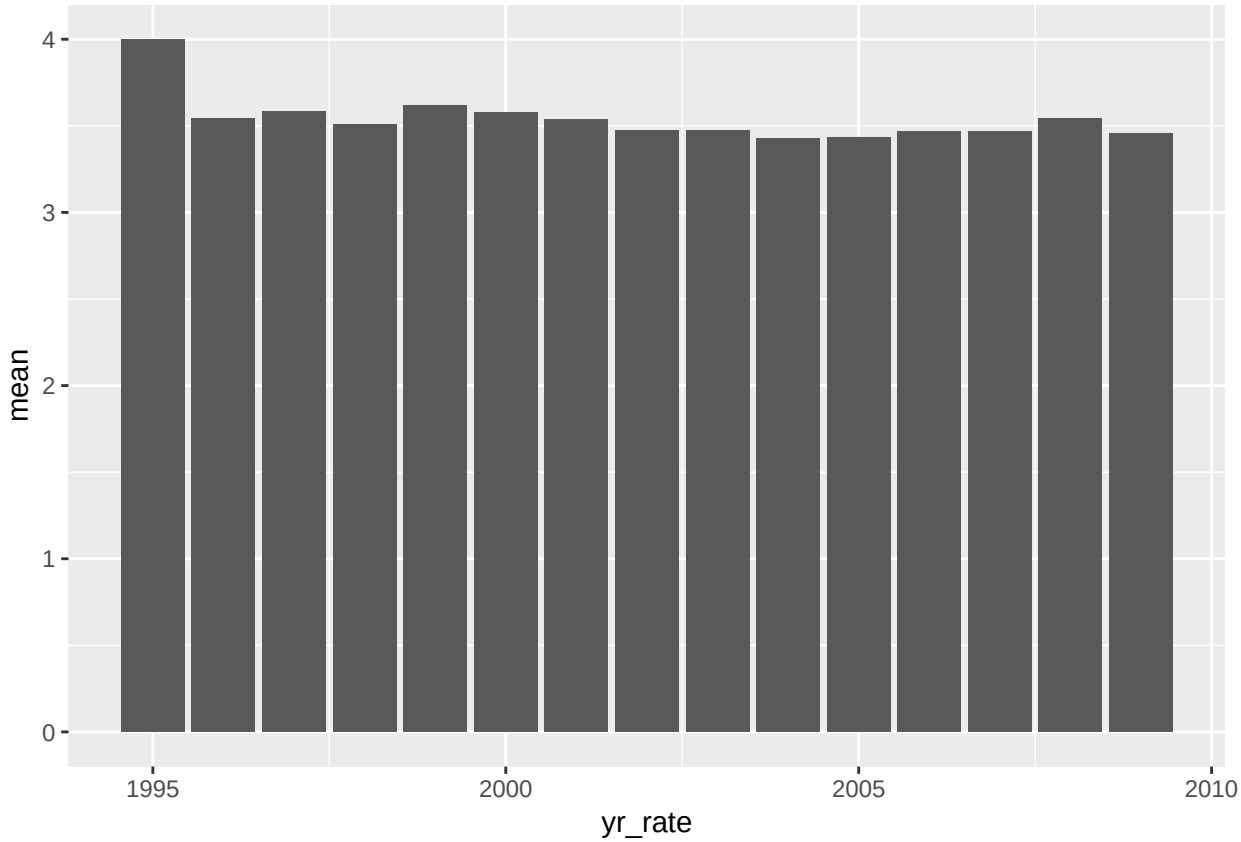
The first effect will be the movie itself. We see quite a bit of variability in the mean ratings, suggesting (as we would expect) that this is a significant predictor of rating.



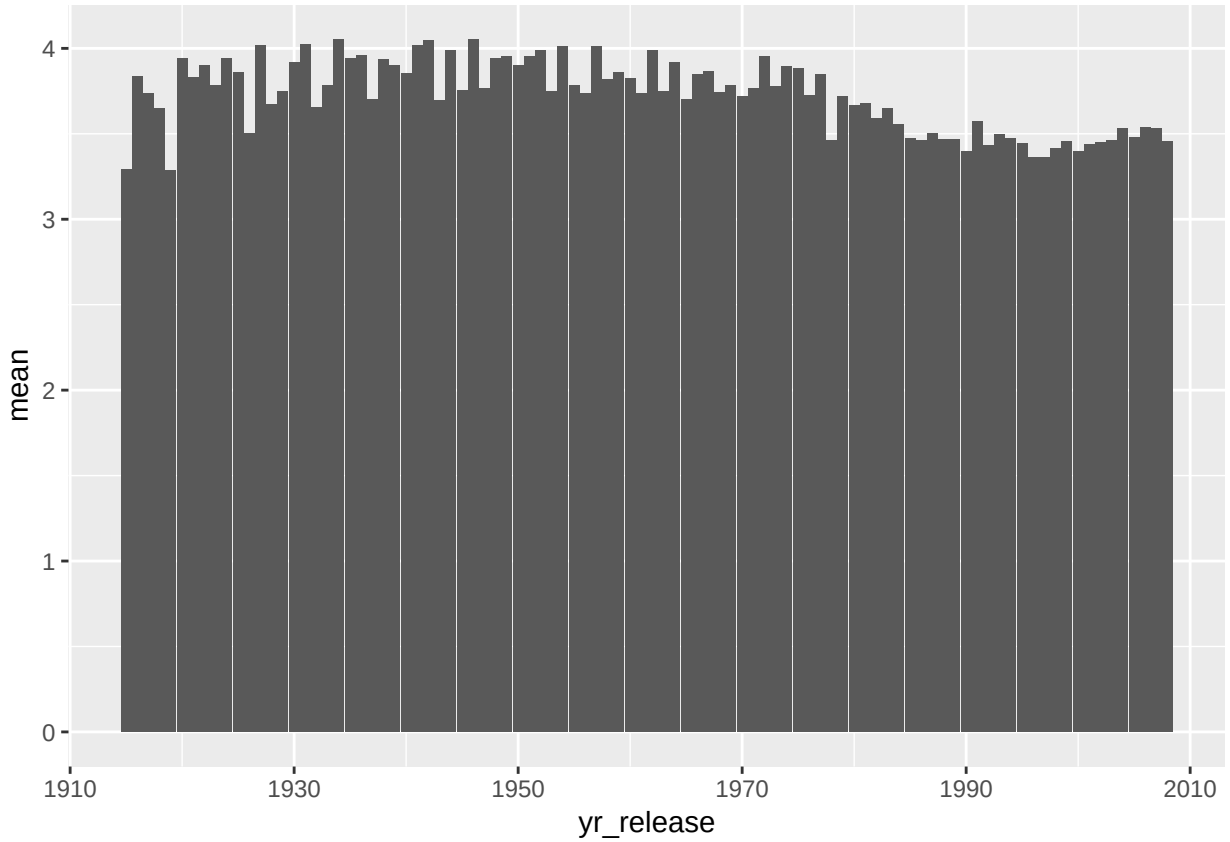
The next effect is the user. We also see significant variability, as again we would expect.



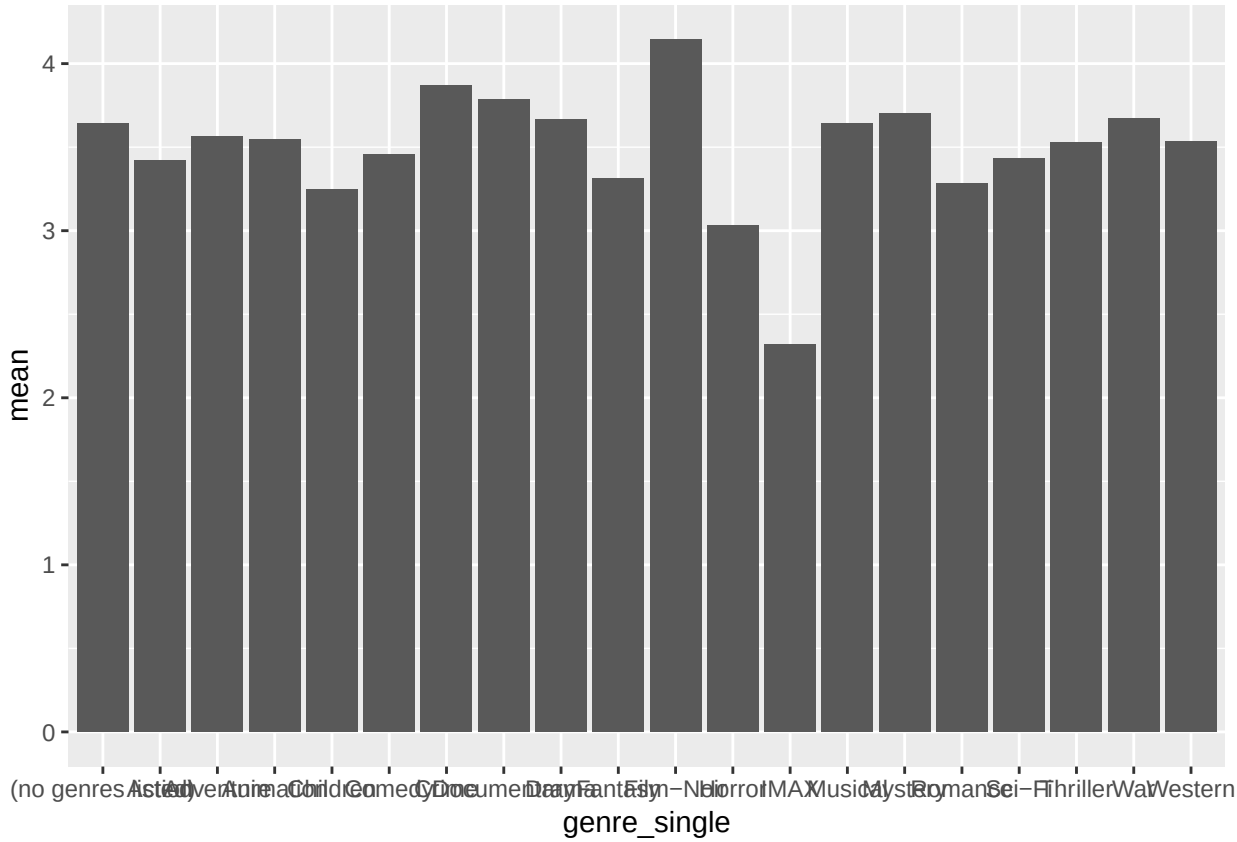
Next to be considered is the year that user rated the movie. Other than the year 1995, the ratings don't seem to vary much by year, suggesting this will have at most a small predictive value.



The year of release of the movie looks to have a small, but fairly consistent effect. In particular, it appears that more recent movies are rated a bit lower. This is perhaps because people tended to only rate older movies that were popular to begin with.



Finally, we look at first listed genre, which has a moderate variability.



Analysis - Model Selection

RMSE Function

We define the RMSE function to be minimized.

```
RMSE <- function(true_ratings,predicted_ratings){
  sqrt(mean((true_ratings-predicted_ratings)^2))}
```

Naive Baseline Model

The simplest model is to choose a constant prediction value, and the most obvious choice for that will be the average (mean) rating from the `edx` dataset, which we will call $\hat{mu}$. This value of $\hat{\mu}$ is

```
## [1] 3.512465
```

Thus the first model is $\hat{y} = \hat{\mu}$, which produces an RMSE of around 1.06 on the `validation` dataset.

Next, we will add the movie itself as a predictor. Assuming movie raters are reasonably consistent this should have the strongest effect on the rating, as presumably better quality movies would receive higher ratings. Thus the Movie Effect model will be $\hat{y} = \hat{\mu} + b_i$, where b_i corresponds to the movie effect for movie i. This reduces the RMSE to around 0.944, better but still short of our goal.

method	RMSE
Naive Model	1.0612018

method	RMSE
Movie Effect Model	0.9439087

The next model will add the user effect, that is, that some users will tend to rate movies higher or lower than other users. This model, Movie+User effects model, will be $\hat{y} = \hat{\mu} + b_i + b_u$, where b_u corresponds to the user effect for user u. This produces an RMSE of 0.8653, very close to the goal.

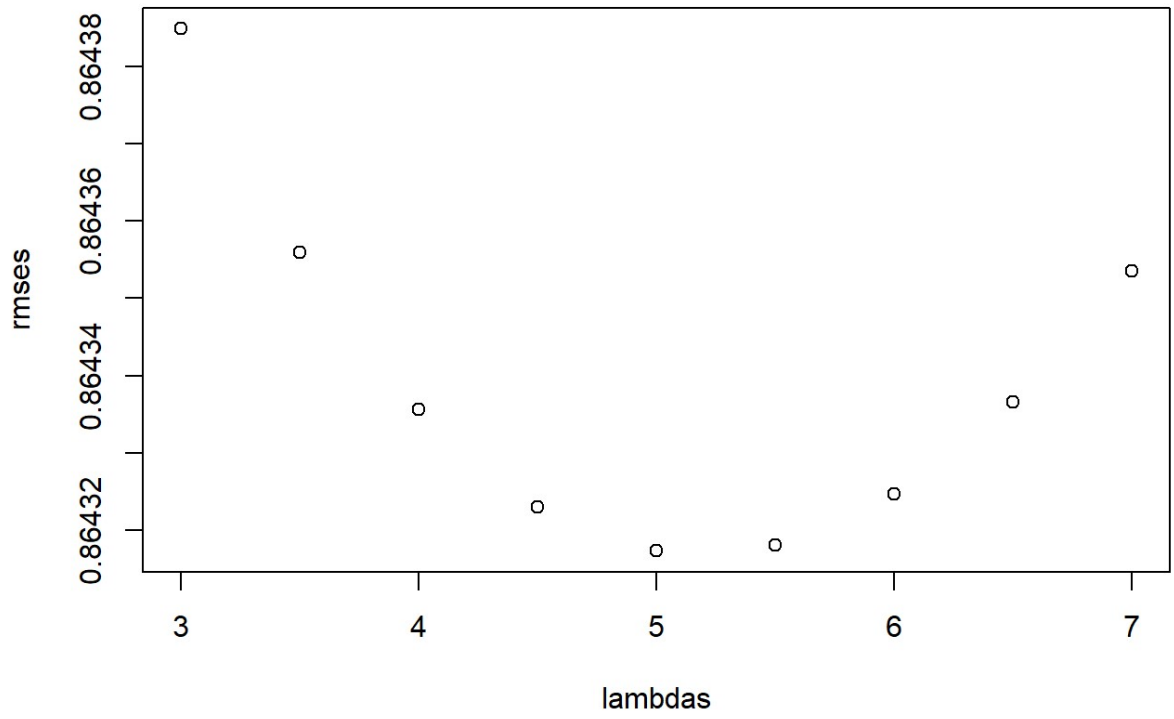
method	RMSE
Naive Model	1.0612018
Movie Effect Model	0.9439087
Movie +User effects model	0.8653488

The next model will add the remaining components (year rated, year released, and first genre), which are expected to have a small but hopefully significant effect. The form of this model will be $\hat{y} = \hat{\mu} + b_i + b_u + b_r + b_w + b_g$, where b_w is the effect of the rating year, b_r corresponds to the release year, and b_g to the genre. The Movie+User+ReleaseYear+Genre+RateYear model actually achieves the required RMSE at 0.8648.

method	RMSE
Naive Model	1.0612018
Movie Effect Model	0.9439087
Movie +User effects model	0.8653488
Movie+User+ReleaseYear+Genre+RateYear model	0.8648283

Our last model will add the feature of "regularization". This idea is attach a penalty, which we will denote λ, in cases where a movie may have a small sample size but a large effect size. We will consider a range of values for λ, from 0 to 10 in steps of 0.5, and choose the lambda that produces the smallest RMSE. Once chosen, the model, Regularized Movie+User+ReleaseYear+Genre+RateYear model, will be $\hat{y}(\lambda) = \hat{\mu} + b_i(\lambda) + b_u(\lambda) + b_w(\lambda) + b_r(\lambda) + b_g(\lambda)$.

After running our model for each value of λ and comparing the RMSE's we graph the values of λ vs RMSE in order to choose the optimal value of λ.



Visually it appears $\lambda = 5$ and we can confirm this.

```
lambda<-lambdas[which.min(rmses)]
lambda
```

```
## [1] 5
```

We run our model again with just this value.

The RMSE for "Regularized Movie+User+ReleaseYear+Genre+RateYear model" on the `validation` dataset is about 0.8643, which is the best performing model of those considered and meets the goal of RMSE < 0.8649.

Results

This is the summary of results for all models trained on `edx` and tested on the `validation` dataset.

method	RMSE
Naive Model	1.0612018
Movie Effect Model	0.9439087
Movie +User effects model	0.8653488
Movie+User+ReleaseYear+Genre+RateYear model	0.8648283
Regularized Movie+User+ReleaseYear+Genre+RateYear model	0.8643173

Conclusion

The initial predictors of `movieId` and `userId` made the most dramatic improvements to the RMSE. Adding regularization and the additional predictors made small but significant reductions in the error which allowed us to meet the desired accuracy.

Limitations and Future Considerations

Other factors that could have been considered but were not were a more specific date of review of the movie (such as incorporating the month of review) and a more detailed genre than just that which was listed first alphabetically. The month of review was viewed as too detailed to be impactful. The `genres` variable consisted of strings of several genres, and extraction would "blow up" the dataset size and strain the computing power that was available. Because of this and that there were so many overlapping genres it was decided to ignore secondary genres. Both features could be considered in future analysis.