HarvardX DataScience Capstone

Naren Kanteti

2023-01-26

Contents

1 Overview & Objectives:

The objective of this project is to predict movie ratings using the MovieLens dataset.The version of movielens we will use in our project is just a small subset of a much larger dataset available with millions of ratings. We will, however, use the smaller dataset to make the computation a little easier. We will explore, analyze the data using analysis and visualization techniques, then move on to build various machine learning models, compare and contrast those until we find the most optimal model for the business case.

2 Install and Load Packages

Let us begin by installing and loading the required packages. This may take a while depending on network bandwidth and computing power being deployed.

2.1 Download data files

Now that the packages have been installed, let us download the required data sets. If operating over slower network speeds, it is recommended to pre-download the data into the project directory.

All went well thus far. No missing data in our data set. Let us answer a few initial questions for the quiz. This will also help us to get a feel for the overall data

3 Data Analysis

Let's start by looking at the Edx dataset.

Number of total rows and columns in the Edx data which will be used for training.

```
## [1] 9000055       6
```

How many zeros were given as ratings in Edx dataset?

```
## [1] 0
```

How many 3's were given as rankings in Edx dataset?

```
## [1] 2121240
```

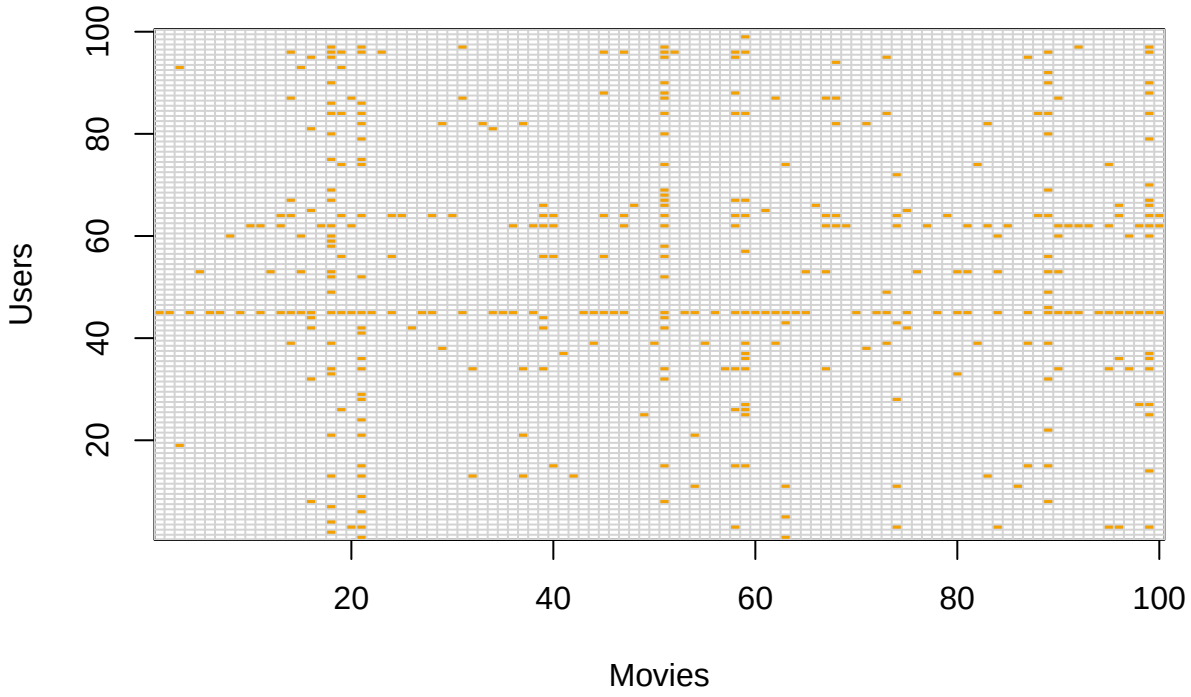
How many unique movies and unique users exists in the dataset?

Table 1: Unique Movies and Unique Users

Unique # of Users	Unique # of Movies
69878	10677

Let us sample the data set visually to get a better understanding of the data. Let us start with Users and Ratings data. Not every user will provide rating for every movie and vice versa. Sample a set of 100 random rows from Edx data set.

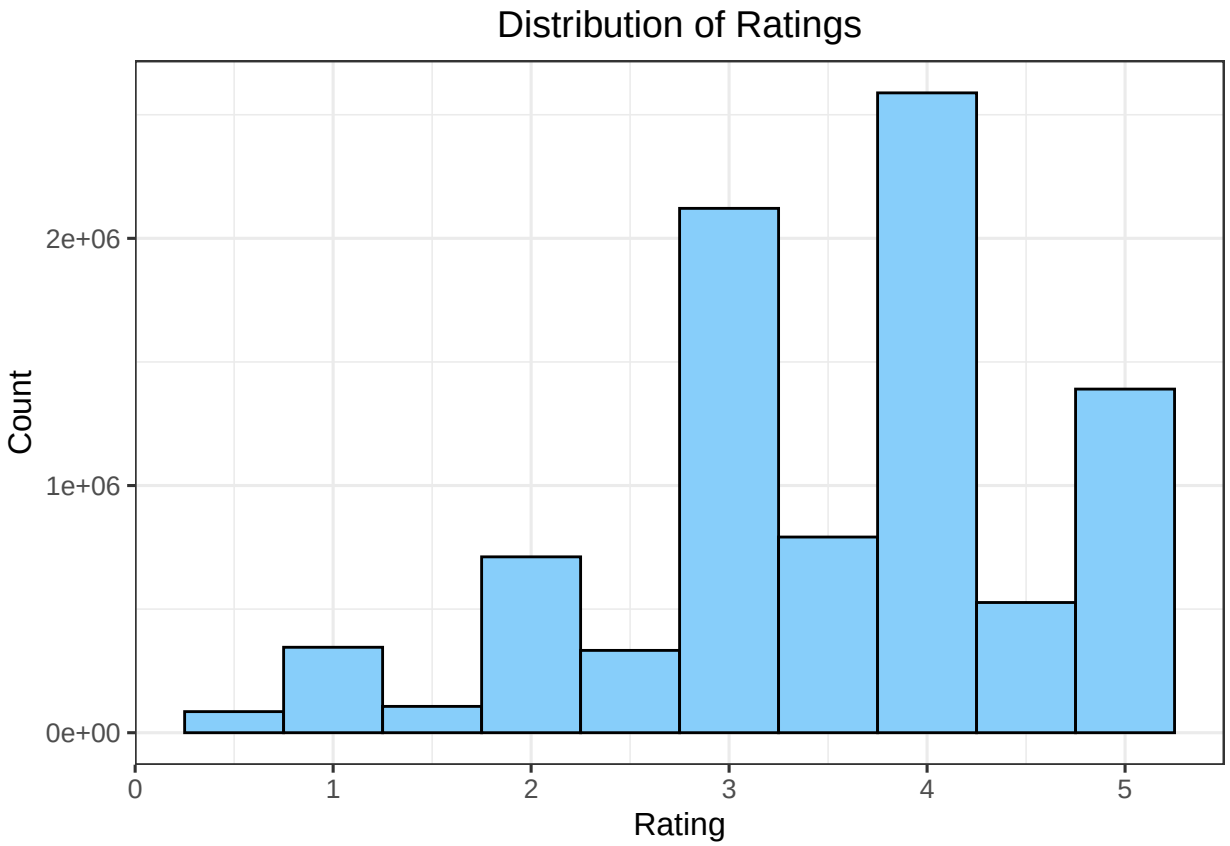
	userId	movieId	rating	timestamp	title	genres
1	1	122	5	838985046	Boomerang (1992)	Comedy|Romance
2	1	185	5	838983525	Net, The (1995)	Action|Crime|Thriller
4	1	292	5	838983421	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller
5	1	316	5	838983392	Stargate (1994)	Action|Adventure|Sci-Fi
6	1	329	5	838983392	Star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



The solid dots are where a user provided a rating for a particular movie. The gaps are where there is no corresponding data point at the intersection of a particular user and a particular movie. Ideally, our algorithm should, towards the end, be able to predict those values.

3.1 Ratings

Let us start by looking at the ratings data in detail. Plot the ratings distribution on a histogram to get a view on data distribution.



Summary of the total number of ratings available.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   85374  336177  619079  900006 1240492 2588430
```

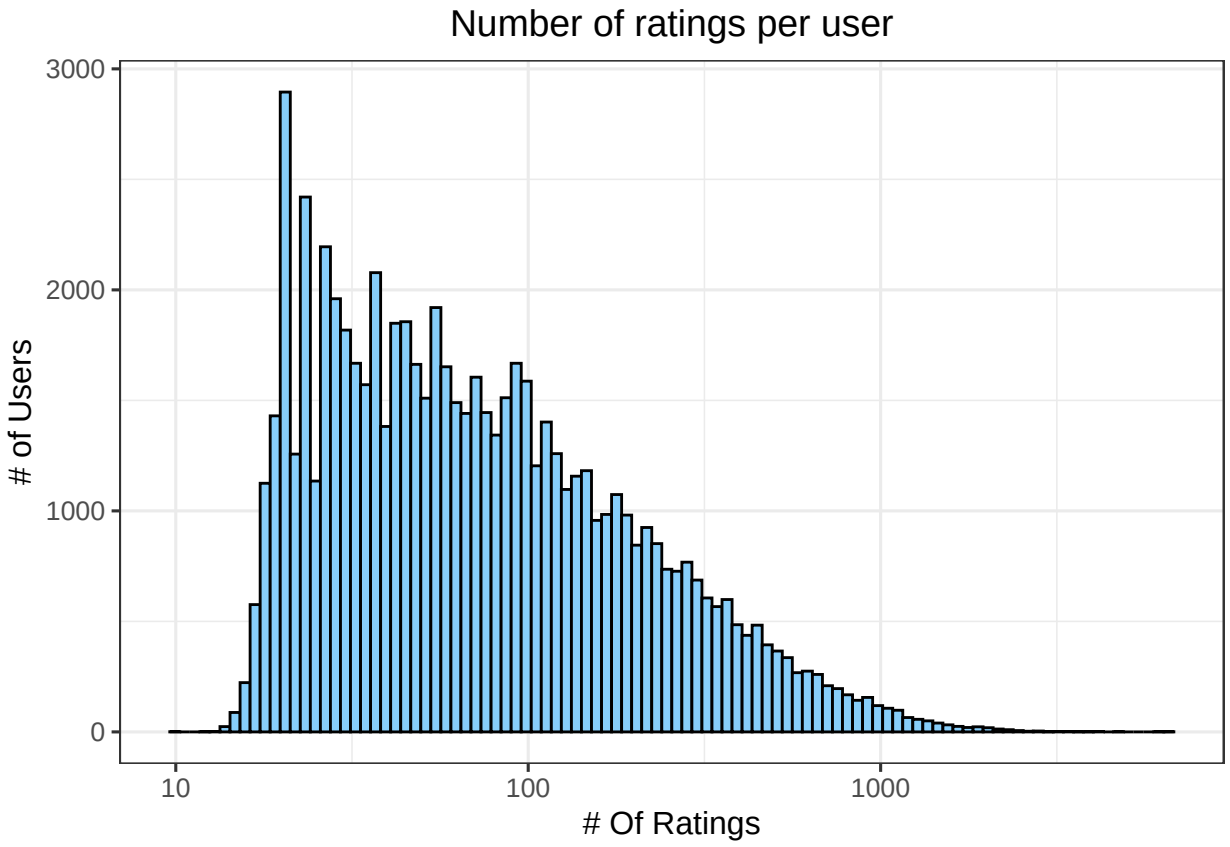
Table 2: Ranking of the ratings given and number of times they were given

Rating	Total Count
4.0	2588430
3.0	2121240
5.0	1390114
3.5	791624
2.0	711422
4.5	526736
1.0	345679
2.5	333010
1.5	106426
0.5	85374

With 2,588,430 ocuurances, rating of 4 occurs the most. The median of the rating distribution is 619079 indicating that data might be skewed. The more number of ratings a movie received, the most likely it is to get a rating of 3 of higher. The movies on the either extreme ends of the ratings curve(0, and 5) received relatively fewer ratings.

3.2 User Distribution

Let us look at the user distribution. Start with plotting general user distribution to see how many times a given user has provided ratings




```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    10.0    32.0    62.0   128.8   141.0  6616.0
```

Over 50% of the users provided 62 ratings or fewer, and a third of the user base provided ratings for less than 140 movies, with the one user providing the most number of views (6616) and on the low side we have a user with 10 total ratings.

3.2.1 Users by number of ratings provided

Let us analyze the relation between number of ratings given by a user and how they tend to rate.

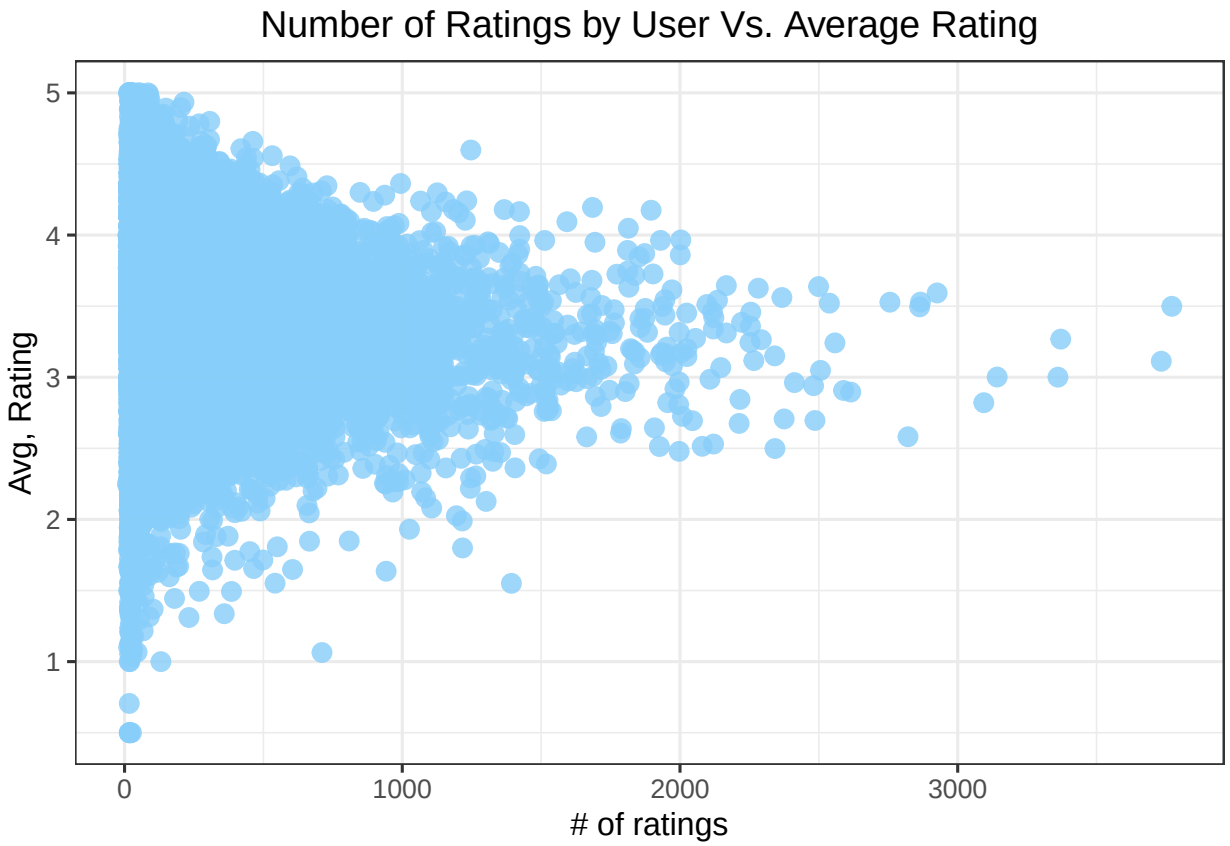


Table 3: Users with highest number of ratings given (top 10)

Id	Ratings	Avg. Rating
59269	6616	3.26
67385	6360	3.20
14463	4648	2.40
68259	4036	3.58
27468	4023	3.83
19635	3771	3.50
3817	3733	3.11
63134	3371	3.27
58357	3361	3.00
27584	3142	3.00

Let us rank the user data by their average ratings.

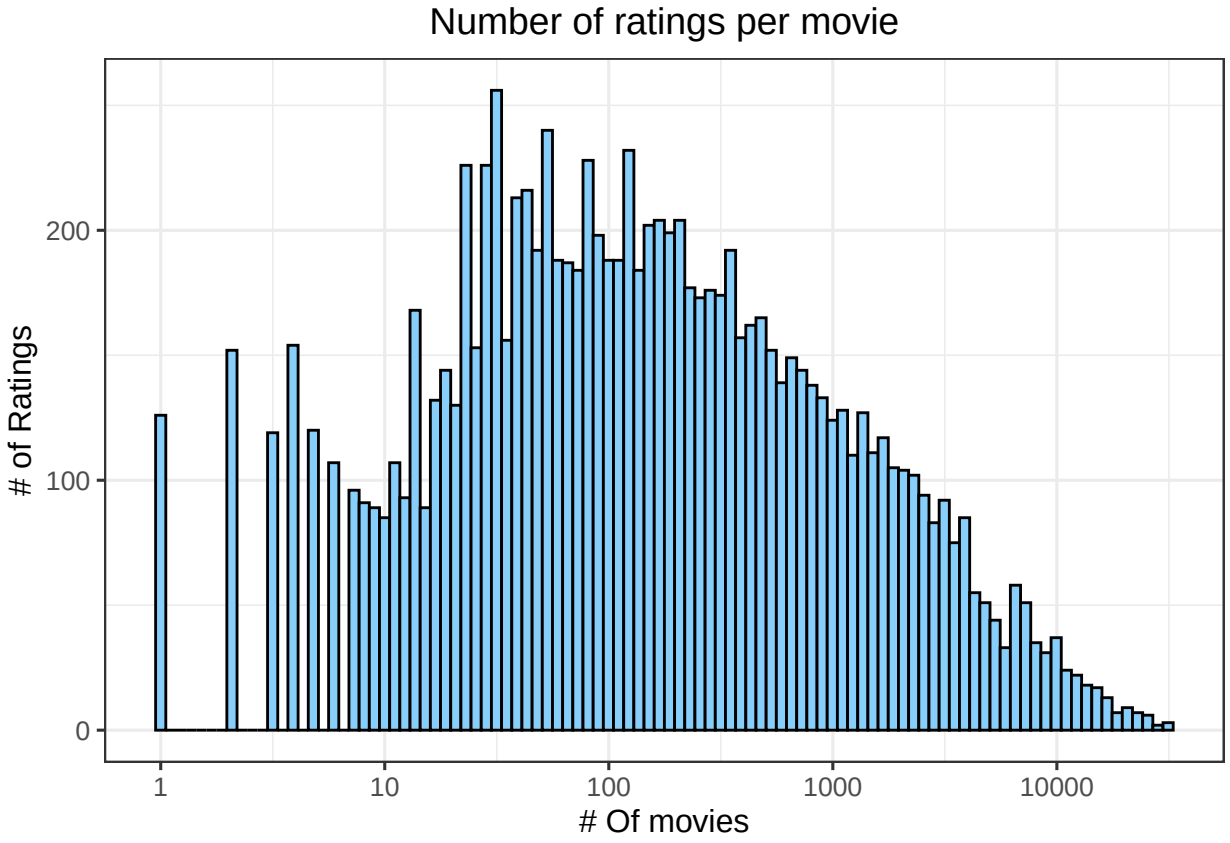
This clearly indicates that the more number of ratings a user submits, the more balanced their ratings are. The users with few number of ratings tend to skew the rating to the higher side.This was also evident when we plotted a rating distribution earlier.

3.3 Movie Distribution

Let us look at just the movie data and how its distribution is.

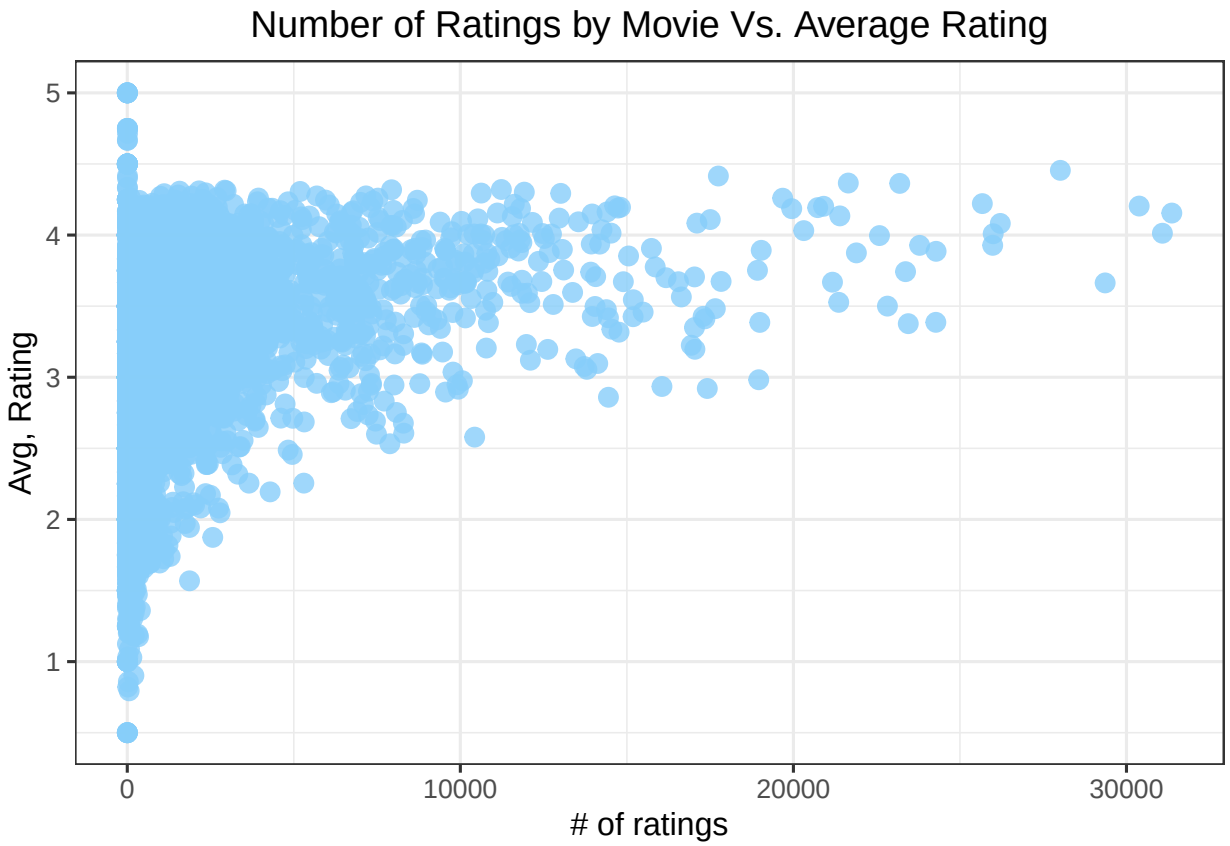
Table 4: Users with least number of ratings given

Id	Ratings	Avg. Rating
57894	14	4.36
62317	14	3.71
63143	14	3.93
68161	14	3.43
68293	14	4.36
71344	14	3.21
15719	13	3.77
50608	13	3.92
22170	12	4.00
62516	10	2.25

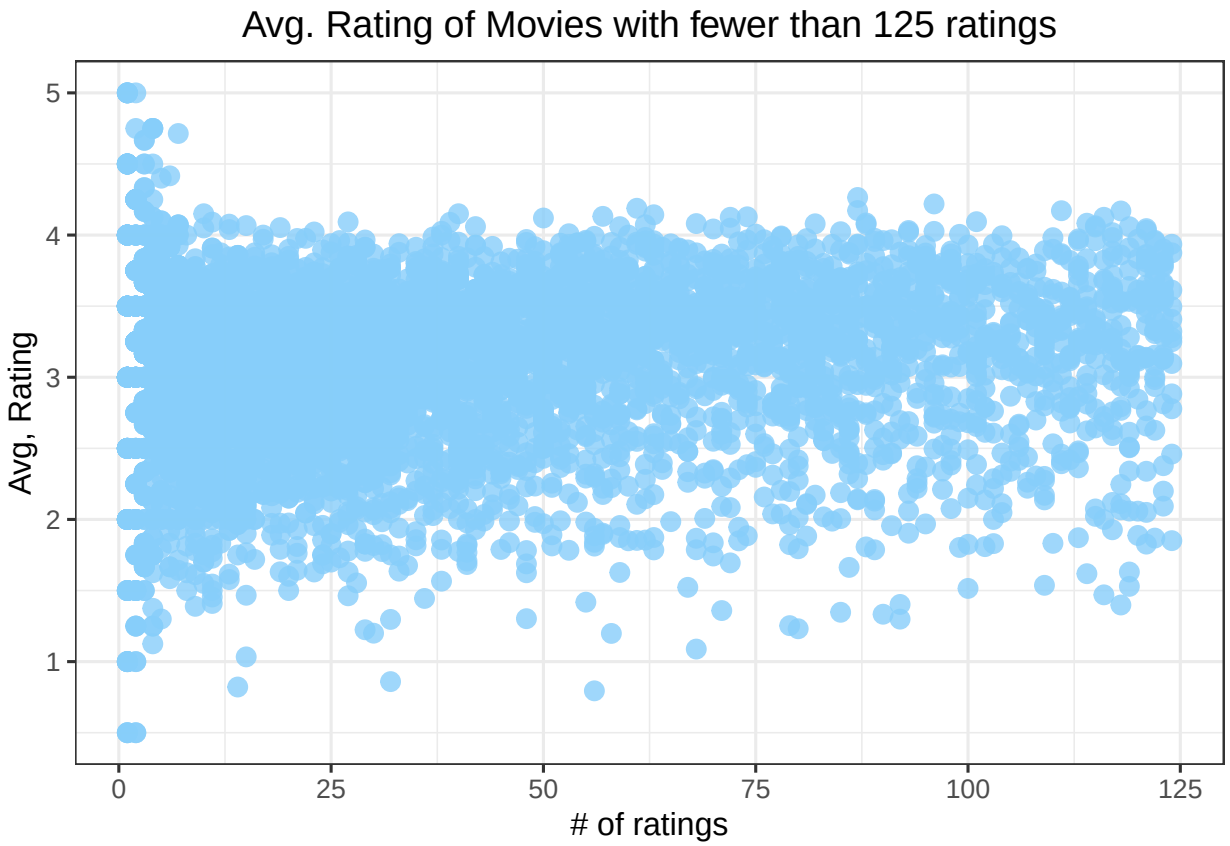



```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0    30.0   122.0   842.9   565.0 31362.0
```

This shows that a vast number of movies received rating fewer than 100 times. 75% of the movies in the dataset received fewer than 565 ratings. Pulp Fiction was the movie which received a rating most number of times. Let us evaluate if the number of times rating received has any correlation with the average rating received.



Let see how the averages fare if we look at movies that only received less than 125 ratings



Similar to the user data, the fewer the ratings a movies received, the distribution of average ratings spans the entire scale.

Let us look the movies that received the most number of ratings.

Table 5: Movies with highest number of ratings

Id	Title	Ratings	Average
296	Pulp Fiction (1994)	31362	4.15
356	Forrest Gump (1994)	31079	4.01
593	Silence of the Lambs, The (1991)	30382	4.20
480	Jurassic Park (1993)	29360	3.66
318	Shawshank Redemption, The (1994)	28015	4.46
110	Braveheart (1995)	26212	4.08
457	Fugitive, The (1993)	25998	4.01
589	Terminator 2: Judgment Day (1991)	25984	3.93
260	Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977)	25672	4.22
150	Apollo 13 (1995)	24284	3.89

The average rating of the movies with the highest number of ratings tend to concentrate around 4 (a quiz question), with Pulp Fiction receiving the most number of ratings. Now that we are done with the number of ratings, let us look at the same data from average ratings perspective. To start with, let us see which movies received the highest average rating.Make sure we don't get confused between average rating of a movie and number of ratings a movie received. This analysis will help us during model building to account for any biases so we can regularize the data to correct them.

Table 6: Movies with least number of ratings

Id	Title	Ratings	Average
64903	Nazis Strike, The (Why We Fight, 2) (1943)	1	3.5
64918	Small Cuts (Petites coupures) (2003)	1	3.0
64926	Battle of Russia, The (Why We Fight, 5) (1943)	1	3.5
64944	Face of a Fugitive (1959)	1	3.0
64953	Dirty Dozen, The: The Fatal Mission (1988)	1	3.5
64976	Hexed (1993)	1	1.5
65006	Impulse (2008)	1	4.0
65011	Zona Zamfirova (2002)	1	4.0
65025	Double Dynamite (1951)	1	2.0
65027	Death Kiss, The (1933)	1	2.5

Table 7: Movies with highest average ratings

Id	Title	Ratings	Average
3226	Hellhounds on My Trail (1999)	1	5.00
33264	Satan's Tango (Sátántangó) (1994)	2	5.00
42783	Shadows of Forgotten Ancestors (1964)	1	5.00
51209	Fighting Elegy (Kenka erejii) (1966)	1	5.00
53355	Sun Alley (Sonnenallee) (1999)	1	5.00
64275	Blue Light, The (Das Blaue Licht) (1932)	1	5.00
5194	Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to tamo peva) (1980)	4	4.75
26048	Human Condition II, The (Ningen no joken II) (1959)	4	4.75
26073	Human Condition III, The (Ningen no joken III) (1961)	4	4.75
65001	Constantine's Sword (2007)	2	4.75

As you can see the movies with the highest average ratings did not have that many ratings. These are not the most popular movies either. Finishing up the analysis, let us look at the movies that receive the least average rating.

Table 8: Movies with least average ratings

Id	Title	Ratings	Average
55324	Relative Strangers (2006)	1	1.00
6483	From Justin to Kelly (2003)	199	0.90
61348	Disaster Movie (2008)	32	0.86
7282	Hip Hop Witch, Da (2000)	14	0.82
8859	SuperBabies: Baby Geniuses 2 (2004)	56	0.79
5805	Besotted (2001)	2	0.50
8394	Hi-Line, The (1999)	1	0.50
61768	Accused (Anklaget) (2005)	1	0.50
63828	Confessions of a Superhero (2007)	1	0.50
64999	War of the Worlds 2: The Next Wave (2008)	2	0.50

As we guessed, these are some unknown movies and also these movies did not receive many ratings.

3.4 Genre Analysis

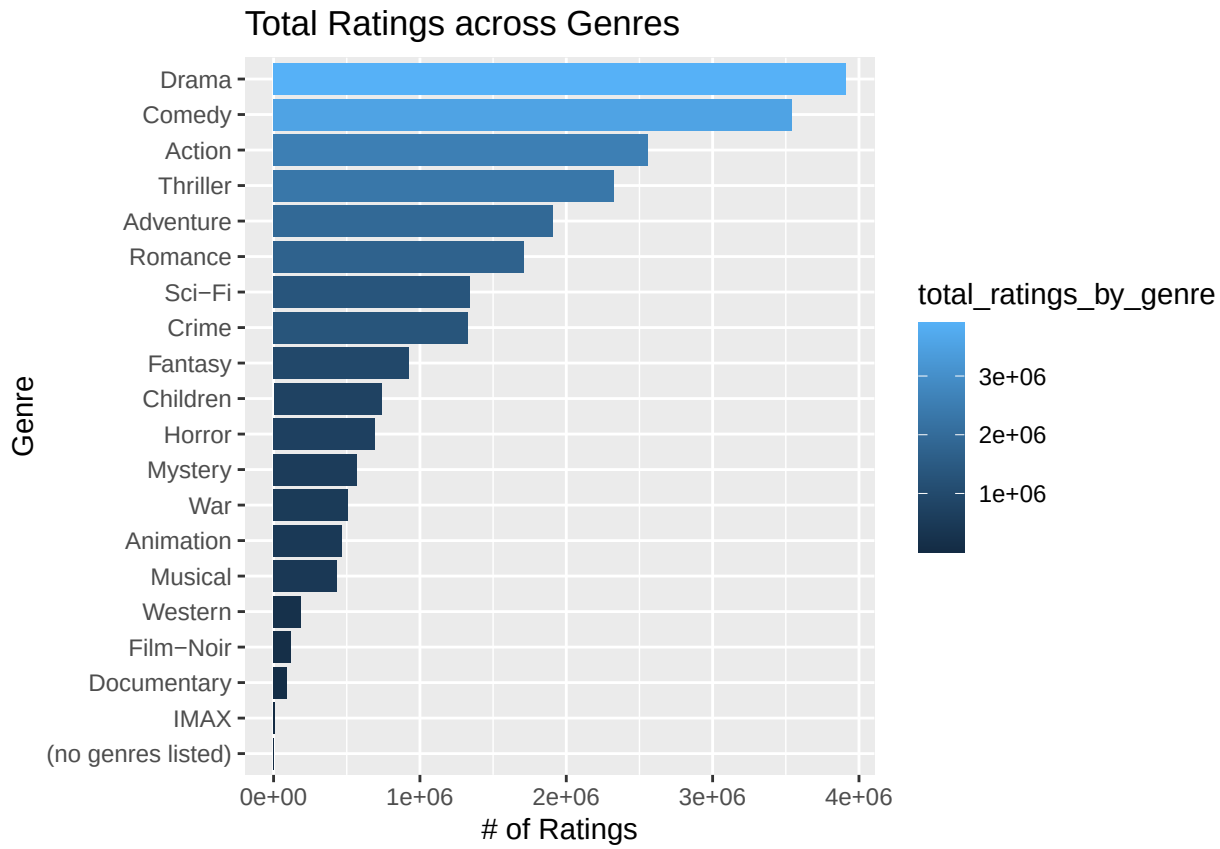
Let us see if Genre has any relation with the rating received.Since any given movie can belong to multiple genres, we need to isolate them. This will create a matrix of total movies times each genre a movie belonged to.Let us start by looking at the distribution of number of ratings by genre.

Table 9: Movie counts by genre

Genre	Ratings
Drama	3910127
Comedy	3540930
Action	2560545
Thriller	2325899
Adventure	1908892
Romance	1712100
Sci-Fi	1341183
Crime	1327715
Fantasy	925637
Children	737994
Horror	691485
Mystery	568332
War	511147
Animation	467168
Musical	433080
Western	189394
Film-Noir	118541
Documentary	93066
IMAX	8181
(no genres listed)	7

Drama category gets the highest number of ratings with Film-Noir getting the least number. This is predictable as the number of drama movies generally made every year are much more than specialized categories such as Film-Noir and documentaries.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       7  372158  714740 1168571 1761298 3910127
```



However, let's see if we have a similar variation with the average ratings by Genre.

This is definitely interesting. The variation of average rating by genre is much less than what we have seen with users and movies. Film-Noir gets the highest average of 4.01(which had the least number of ratings, by the way) with horror genre getting the least of 3.27. The variation here is much better than what we have observed with movie and user data earlier.

3.5 Age of the movie Analysis

Let us evaluate if the age of the movie has any relation on its average rating. Intuitively it should. As older the movie gets, the more time users have to provide ratings. Retrieving the age of the movie is a little tricky with the data set. Given that age is not provided. We need to calculate the date/year the movie was released and calculate its age based on off it. The year is provided in the title, so we need to extract it out.

Table 10: Average Rating by Genre

Genre	Total Ratings	Avg. Rating
Film-Noir	118541	4.01
Documentary	93066	3.78
War	511147	3.78
IMAX	8181	3.77
Mystery	568332	3.68
Drama	3910127	3.67
Crime	1327715	3.67
(no genres listed)	7	3.64
Animation	467168	3.60
Musical	433080	3.56
Western	189394	3.56
Romance	1712100	3.55
Thriller	2325899	3.51
Fantasy	925637	3.50
Adventure	1908892	3.49
Comedy	3540930	3.44
Action	2560545	3.42
Children	737994	3.42
Sci-Fi	1341183	3.40
Horror	691485	3.27

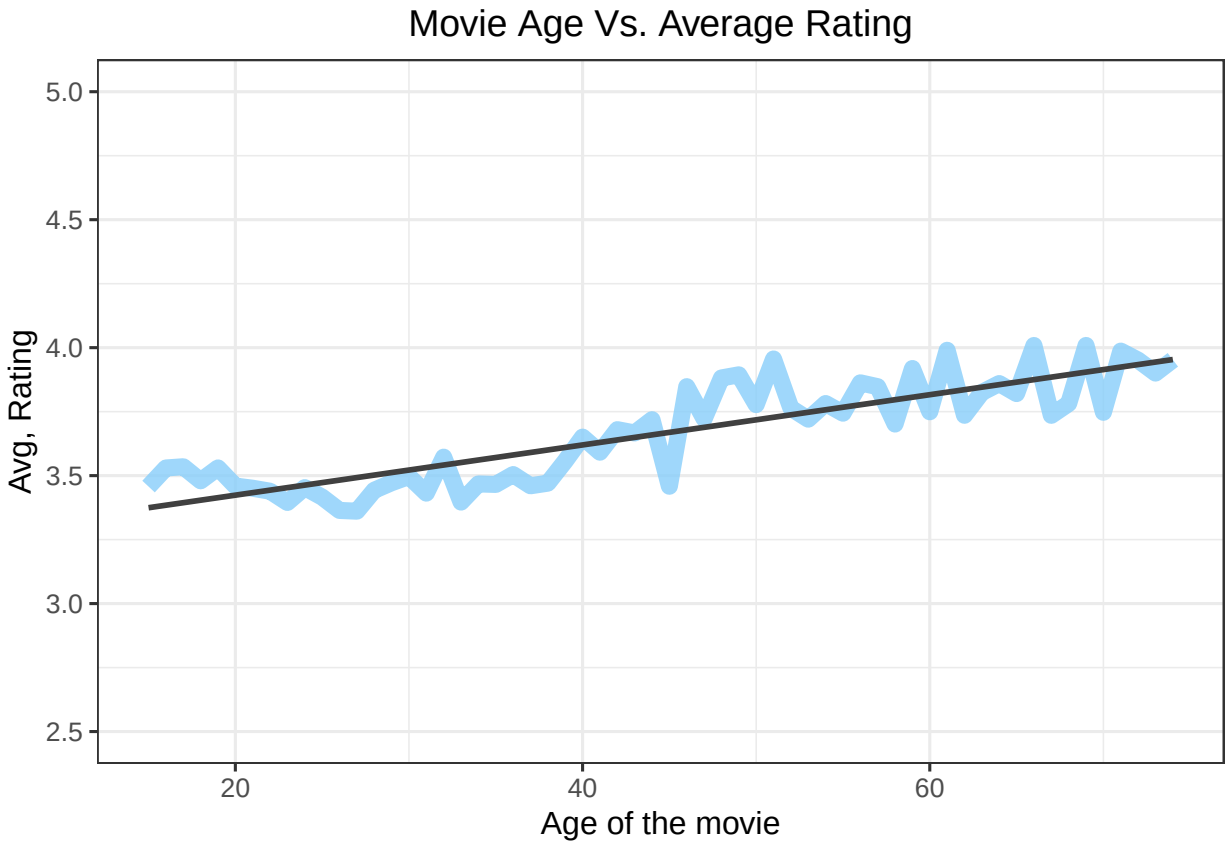
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   15.00   25.00   29.00   32.78   36.00  108.00
```

The median age of the movie if 33, with 75% of the movies in our data set being 36 years old. There are a few outliers in the data, but given that median is closer to the mean, we should be okay here. Let us see if the age of the movie has any effect on number of ratings available and also its average rating.

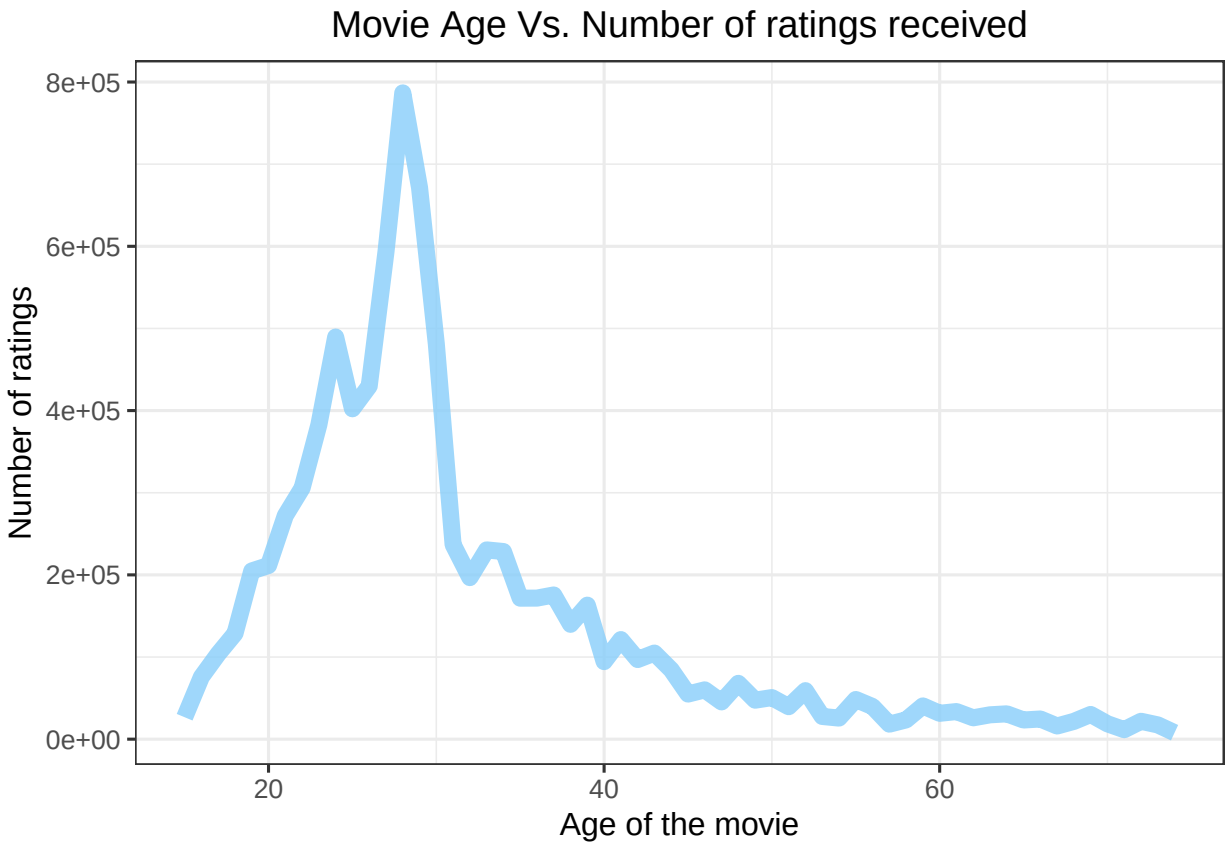
Number of movies with older than 75 years is

```
## [1] 214626
```

Since only a very small portion of movies in our data set are older than 75 years (213626 to be specific), We will exclude them from the following data visualization.



It is an interesting trend. Looks like the rating is a little higher for movies between ages 15 and 20 and tends to dip within ages 20 and 30, and picks back up. Let us look at the number of ratings received by age.



Looks interesting. In contrast to the data above. The movies between the ages of 20 and 30 had the most number of ratings but they are also the one's with lesser average ratings relatively.

Looks like there definitely is a relation between age of the movie and its average rating received by users. Now that we have analyzed the data, let start building a few models using some of the relational insights we have gained thus far. The strategy is to evaluate the effect of movie, user, age and genre on the overall rating of the movie. We will compute the model efficiency as we go and try to make it better. We can also deploy regularization techniques to make it better.

Since the combination of user and movie gets really large and with a lot of missing values, using a regression algorithm is not recommended. We will instead use the residual mean squared error (RMSE) strategy.

To start with, let us build a basic guessing model and see how it fares.

4 Model Building

4.1 Random Guessing

Let us start by randomly guessing a rating between 0 and 5 and see how we do.

```
set.seed(1)
model_0 <- RMSE(sample(seq(1, 5, by = 0.5), size = 1),edx$rating)
training_results <- data.frame(Method="Random Guessing", RMSE=model_0)
training_results %>% knitr::kable(caption = "Training RMSE results",digits = 6) %>%
  kable_styling(font_size = 12, position = "l", latex_options = "hold_position")
```

Table 11: Training RMSE results

Method	RMSE
Random Guessing	1.84712

As you can see the value of RMSE is pretty high.So, just a random guess will not do. Let's see how we do if we use the predicted rating for any movie as the overall mean of the entire data set.

4.2 Using the Mean from Data Set

Now, let us try using the mean value of ratings of the entire data set as our prediction and calculate the RMSE.

```
mu <- mean(edx$rating)
model_1 <- RMSE(mu, edx$rating)
training_results <- bind_rows(training_results,
                              data_frame(Method=" Average of all ratings in Edx", RMSE=model_1))
training_results %>% knitr::kable(caption = "Training RMSE results",digits = 6) %>%
                            kable_styling(font_size=10, latex_options = "hold_position")
```

Table 12: Training RMSE results

Method	RMSE
Random Guessing	1.847120
Average of all ratings in Edx	1.060331

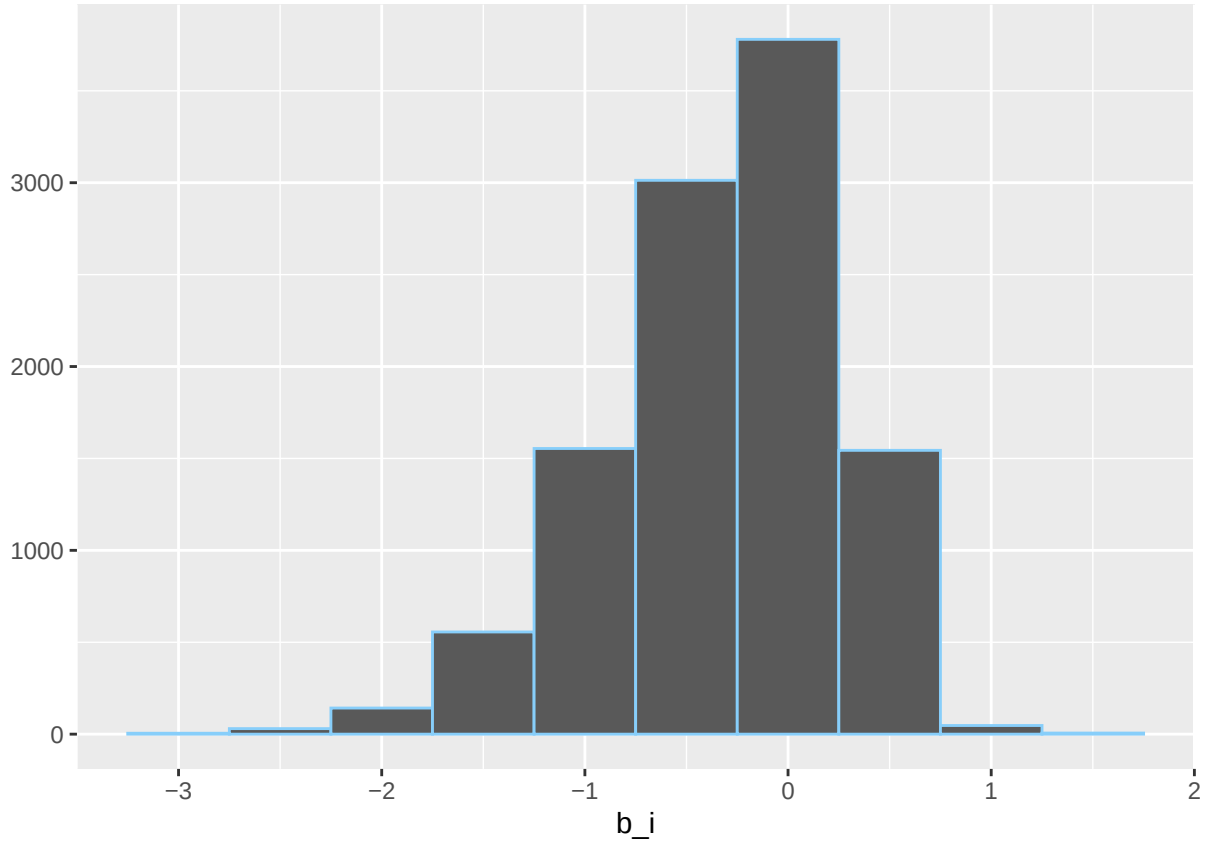
It's gotten better, but is still too high. This is when we start to deploy our earlier mentioned strategy. Let us start evaluating movie effect on our model.

4.3 Modelling movie effect

Let us account for the movie bias, and calculate RMSE again.

```
movie_avgs <- edx %>%
  group_by(movieId) %>%
  dplyr::summarize(b_i = mean(rating - mu))

movie_avgs %>% qplot(b_i, geom ="histogram", bins = 10, data = ., color = I("#87CEFA"))
```




```
predicted_ratings_by_movie <- mu + edx %>%
  left_join(movie_avgs, by='movieId') %>%
  .$b_i

model_2 <- RMSE(predicted_ratings_by_movie, edx$rating)

training_results <- bind_rows(training_results, data_frame(Method="Movie Effect Model",RMSE = model_2 ))
training_results %>% knitr::kable(caption = "Training RMSE results",digits = 6) %>%
                            kable_styling(font_size=10, latex_options = "hold_position")
```

Table 13: Training RMSE results

Method	RMSE
Random Guessing	1.847120
Average of all ratings in Edx	1.060331
Movie Effect Model	0.942348

It is getting better. Let us proceed to evaluate the user effect on predictions.

4.4 Modelling user effect

Similarly, let us account for user bias as well.

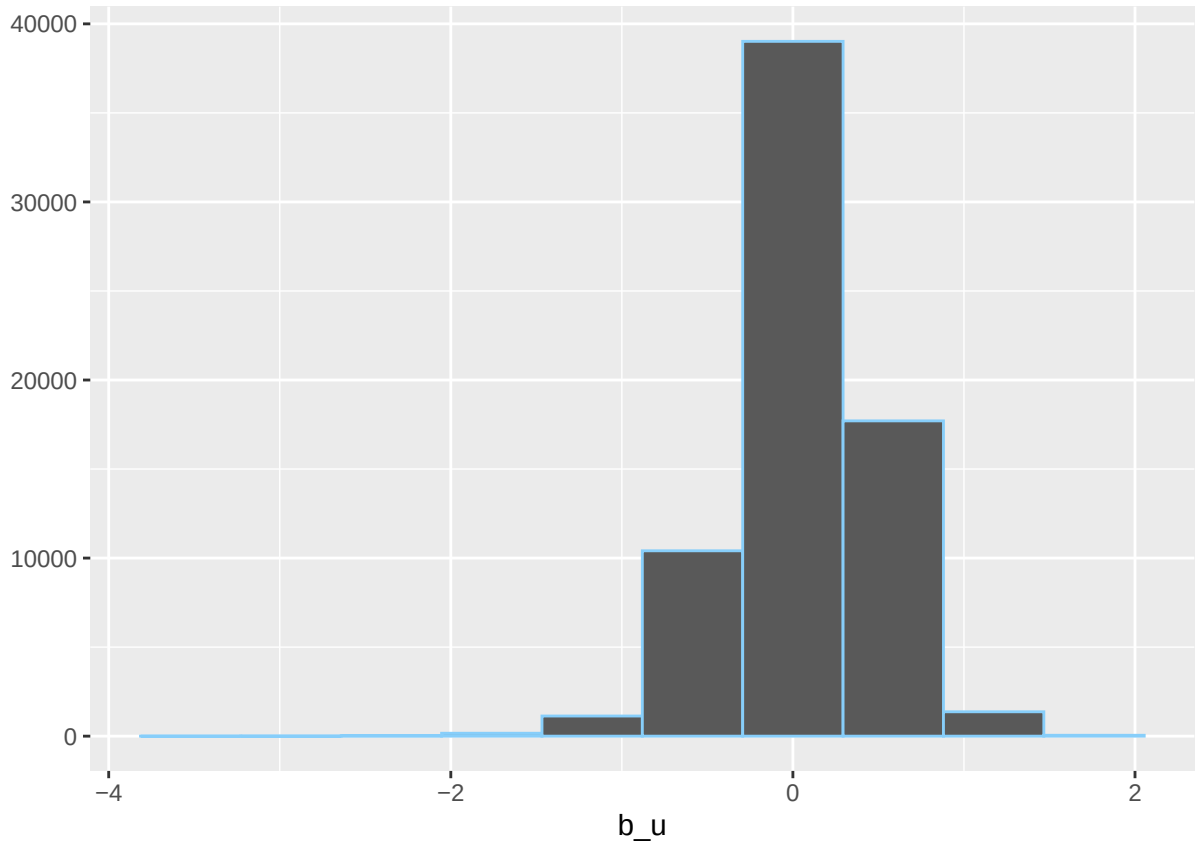


Table 14: Training RMSE results

Method	RMSE
Random Guessing	1.847120
Average of all ratings in Edx	1.060331
Movie Effect Model	0.942348
Movie & Users Model	0.856704

It has definitely gotten better. From our preliminary data exploration, we observed that age did influence the rating of the movie. Let us evaluate age effect into our model.

4.5 Modelling age effect

Accounting for age bias.

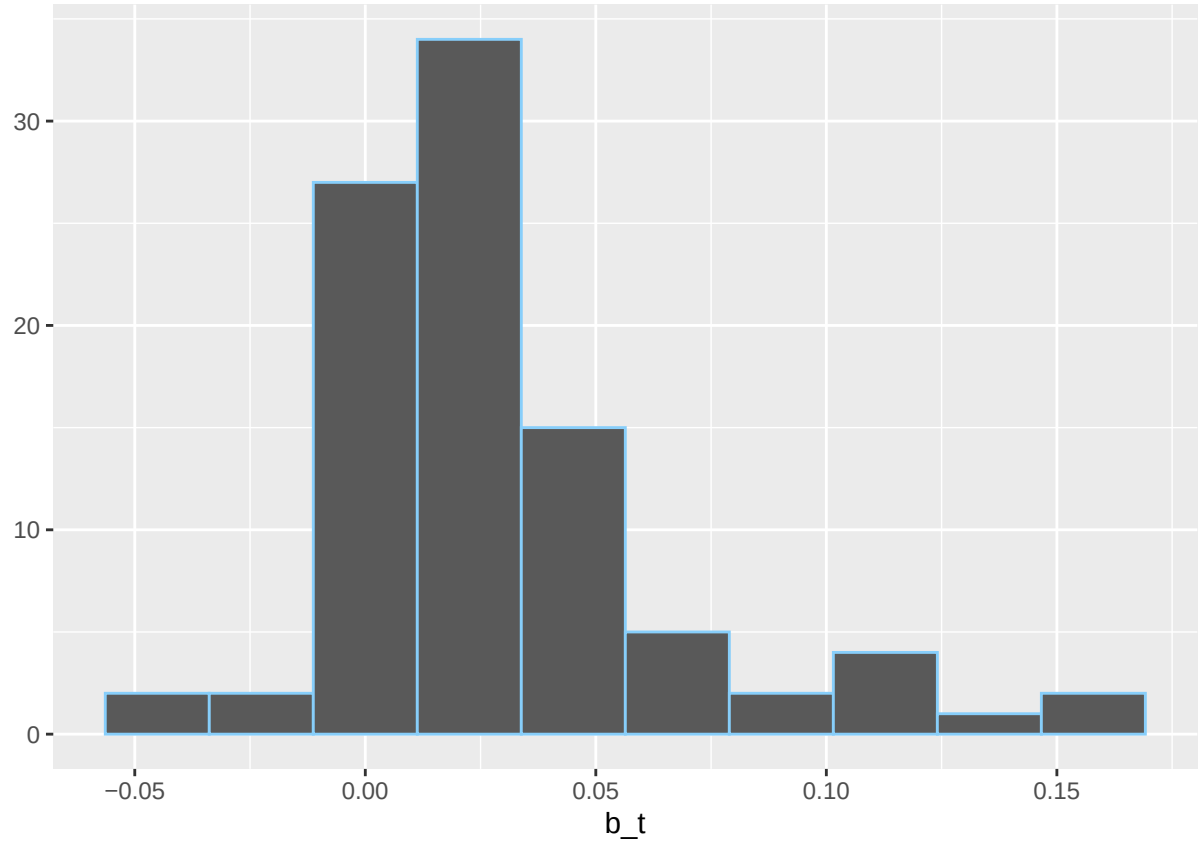


Table 15: Training RMSE results

Method	RMSE
Random Guessing	1.847120
Average of all ratings in Edx	1.060331
Movie Effect Model	0.942348
Movie & Users Model	0.856704
Movie, User and Release Year Model	0.856378

We were able to achieve the least value of RMSE thus far with a training model value of 0.856378.

5 Applying our Model on the validation data set

Now that we have made our model very effective, let us deploy this model on our testing/validation set and evaluate the results. Since we will be using the age as a factor as well, lets add the age column to the testing dataset as well.

```
final_holdout_test <- final_holdout_test %>%
  mutate(year_released = as.integer(substr(title, str_length(title) - 4,
                                           str_length(title) - 1)))

current_year <- as.integer(substr(Sys.Date(),1,4))
final_holdout_test <- final_holdout_test %>%
  mutate(age = as.integer(current_year-year_released))
```

```
predicted_ratings_in_validation_set <- final_holdout_test %>%
  left_join(movie_avgs, by='movieId') %>%
  left_join(user_avgs, by='userId') %>%
  left_join(time_avgs, by='age') %>%
  mutate(pred = mu + b_i + b_u + b_t) %>%
  .$pred

validation_model <- RMSE(predicted_ratings_in_validation_set, final_holdout_test$rating)

validation_results <- data_frame(Method=" Final Validation", RMSE=validation_model)
validation_results %>% knitr::kable(caption = "Validation RMSE results",digits = 6) %>%
                              kable_styling(font_size=10, latex_options = "hold_position")
```

Table 16: Validation RMSE results

Method	RMSE
Final Validation	0.865004

This is a decent RMSE and is very close to the value we computed on the training set, but still not there yet.As per guidelines, the target score we have to achieve in this project should be less than 0.086490. We are a little off from our target.

As we have seen from the data exploration on "user vs ratings" data, there are definitely outliers in the data that are skewing the data. The users that rated the movie very high or very low only rated a few times, and in some instances only once. Hence, regularization of the data might help make the model better.

5.1 Regularize Movie Data

Let us begin by regularizing movie data

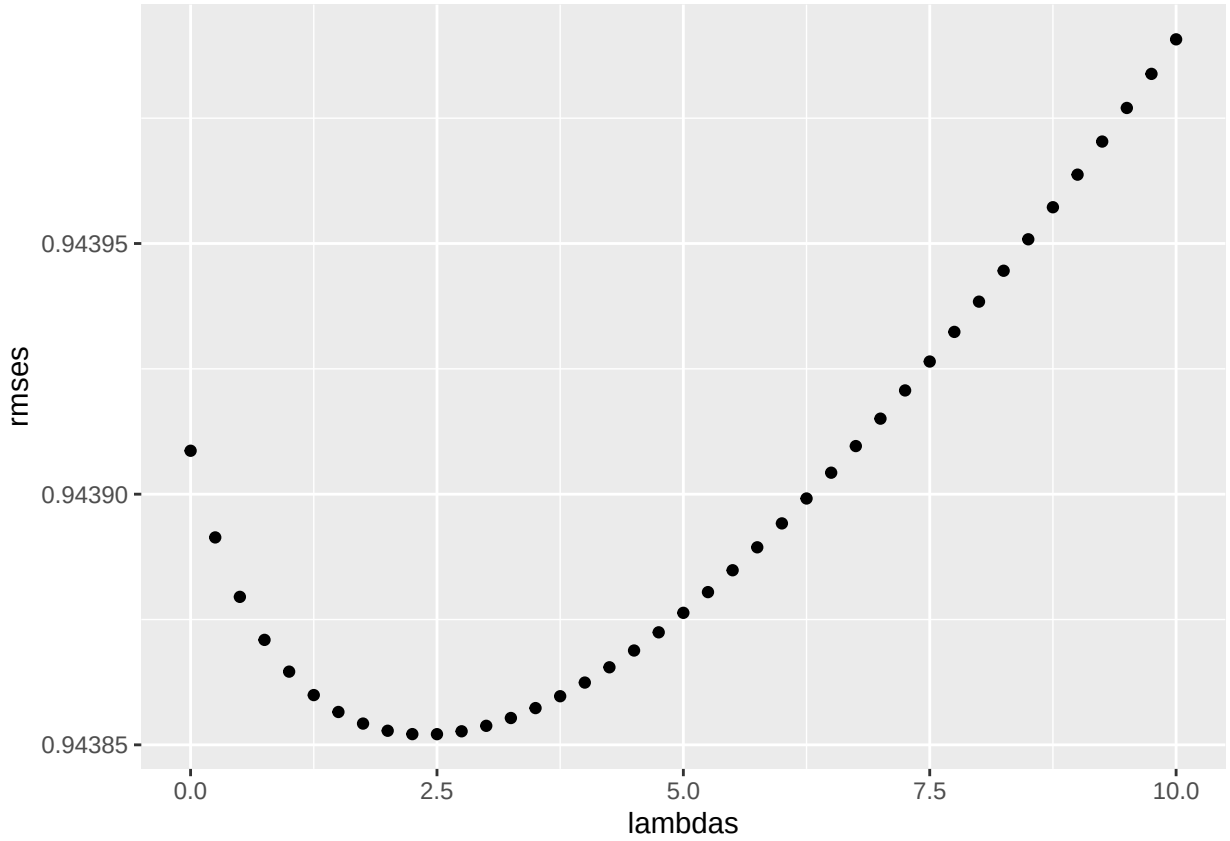


Table 17: Validation RMSE results

Method	RMSE
Final Validation	0.865004
Regularized Movie Model	0.943852

Regularization of movie data by itself actually made it worse.

5.1.1 Regularize Movie and User Data

Let us move on to user data and see its effects on our score.We will also use cross validation technique to derive the lambda value that gives us the least RMSE for this model. We will follow a similar technique for regularizing other factors as well, as necessary.

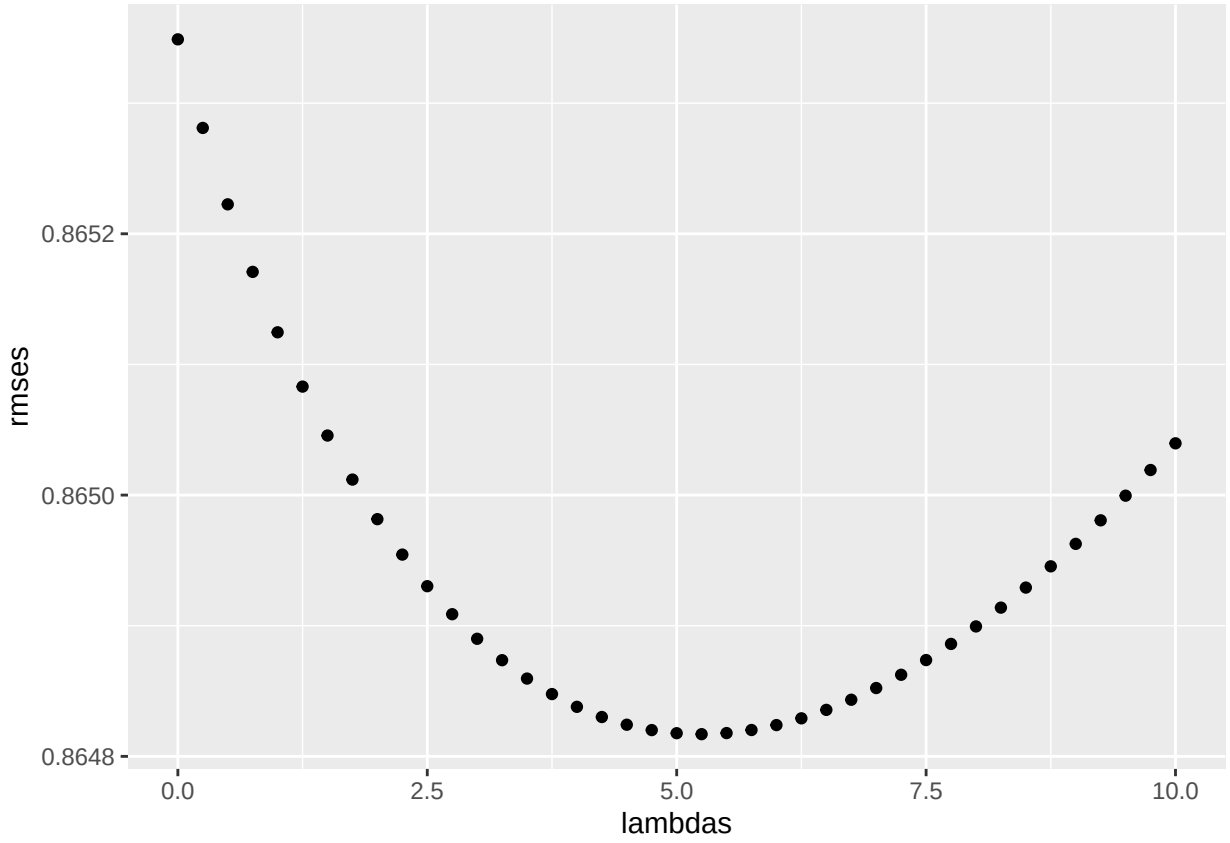


Table 18: Validation RMSE results

Method	RMSE
Final Validation	0.865004
Regularized Movie Model	0.943852
Regularized Movie + Regulaized User Effect Model	0.864817

This helped the most. With a RMSE value of 0.86481, we have hit our goal of getting a score less than 0.086490.

5.1.2 Movie, User and Age regularization

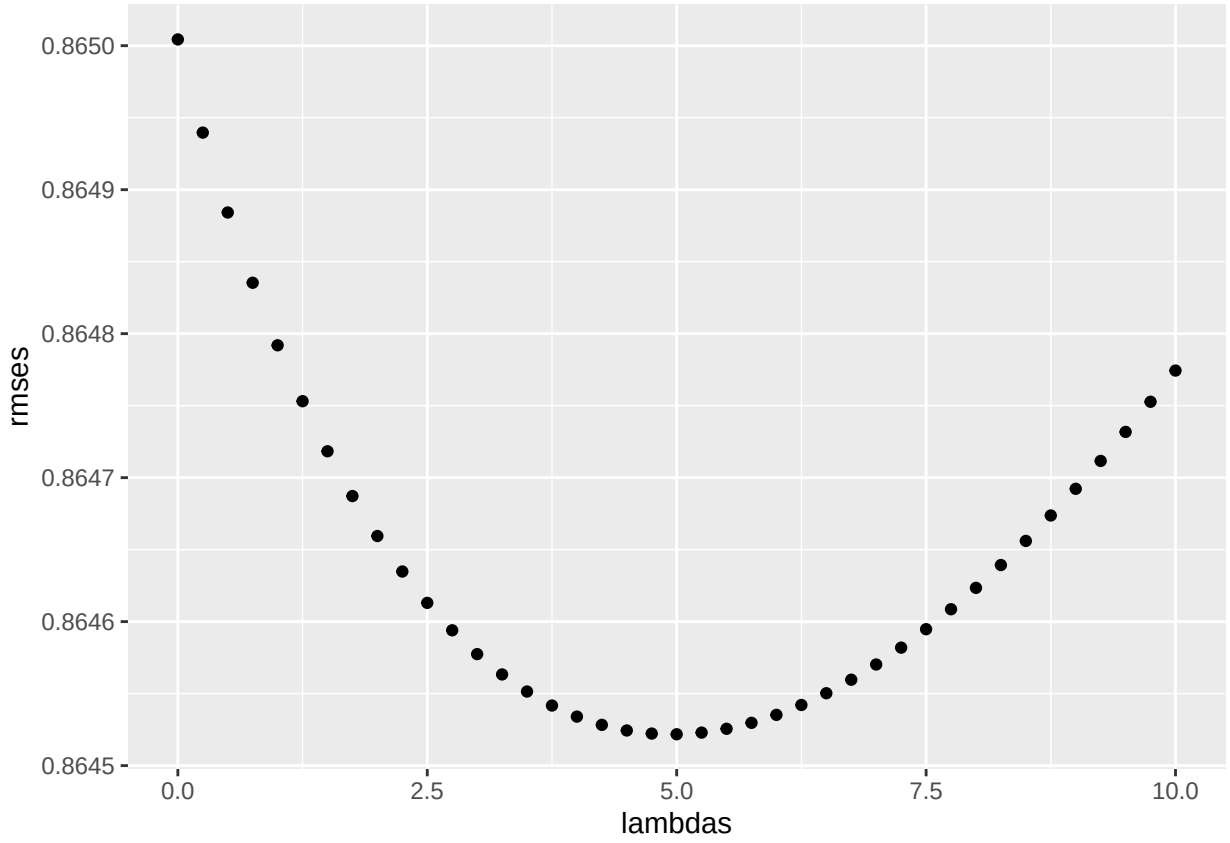
Let us also see if age regularization will have any effect on our final outcome.

```
lambdas <- seq(0, 10, 0.25)
rmses <- sapply(lambdas, function(l){

  mu <- mean(edx$rating)
  b_i <- edx %>%
    group_by(movieId) %>%
    dplyr::summarize(b_i = sum(rating - mu)/(n()+l))
  b_u <- edx %>%
    left_join(b_i, by="movieId") %>%
    group_by(userId) %>%
```

```
    dplyr::summarize(b_u = sum(rating - b_i - mu)/(n()+l))
  b_t <- edx %>%
    left_join(b_i, by="movieId") %>%
    left_join(b_u, by= "userId") %>%
    group_by(age) %>%
    dplyr::summarize(b_t = sum(rating - b_i - b_u - mu)/(n()+l))
  predicted_ratings <- final_holdout_test %>%
    left_join(b_i, by = "movieId") %>%
    left_join(b_u, by = "userId") %>%
    left_join(b_t, by = "age") %>%
    mutate(pred = mu + b_i + b_u + b_t) %>%
    .$pred
  return(RMSE(predicted_ratings, final_holdout_test$rating))
})

qplot(lambdas, rmses)
```




```
lambdas[which.min(rmses)]
```

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

By far, regularization of User data had the most effect on our final value, and age did make it a little better as well.

It did help make it a little better.This is the best score we have been achieve thus far. Our final score is **0.086452.**

Table 19: Validation RMSE results

Method	RMSE
Final Validation	0.865004
Regularized Movie Model	0.943852
Regularized Movie + Regulaized User Effect Model	0.864817
Regularized Movie + Regulaized User + Regularized Age Effect Model	0.864522

6 Conclusion

In the end, the model that accounts for regularization of predictors produced the best results for us. We could have deployed various other strategies to get the rmse lower such as matrix factorization with stochastic gradient (best suited for use cases such as these), single value decomposition method (the actual strategy that won the Netflix contest) and Bayesian SVD+ etc.,. While working on this project, I had to refer back to the course text book multiple times which helped me understand the problem and various available techniques even better.

Next steps are to actually build a matrix factorization model, analyze and evaluate its performance.

Citations:

1. Simsekli, Umut & Koptagel, Hazal & Güldaş, Hakan & Cemgil, Ali & Öztoprak, Figen & Birbil, Ilker. (2015). Parallel Stochastic Gradient Markov Chain Monte Carlo for Matrix Factorisation Models.
2. Yu-Chin Juan, Wei-Sheng Chin, Yong Zhuang, Bo-Wen Yuan, Meng-Yuan Yang, and Chih-Jen Lin.LIBMF: A Matrix-factorization Library for Recommender Systems
3. Rafael A. Irizarry, Introduction to Data Science.