# MovieLens Project

*Daniel Cash (dwcash)*

*April 26, 2019*

## MovieLens: Introduction

On October 2006, Netflix offered a challenge to improve their recommendation algorithm by 10% and win a million dollars. The Netflix data is not publicly available, but thanks to the GroupLens research lab we have a similar kind of dataset to analyze. Only a small subset of the data is used, which is available in the dslabs package. The data has already been split into training and testing sets per the course instructions. The goal of this project is to predict the rating a user would give to a specific movie. This algorithm would allow us to provide relevant movie suggestions to users, in other words, a recommendation system. The loss function used is the residual mean squared error (RMSE) and the cut-off for full credit is to achieve a RMSE below or equal to 0.87750.

## Methods and Analysis

Since the dataset provided was already in tidy format, no data cleaning was necessary. First, the composition of the dataset is analyzed to see if it is more sparse or dense. Of all the Netflix videos or shows I've watched I may have only rated once, if at all. If the average user is anything like me, there will be a lot of missing ratings.

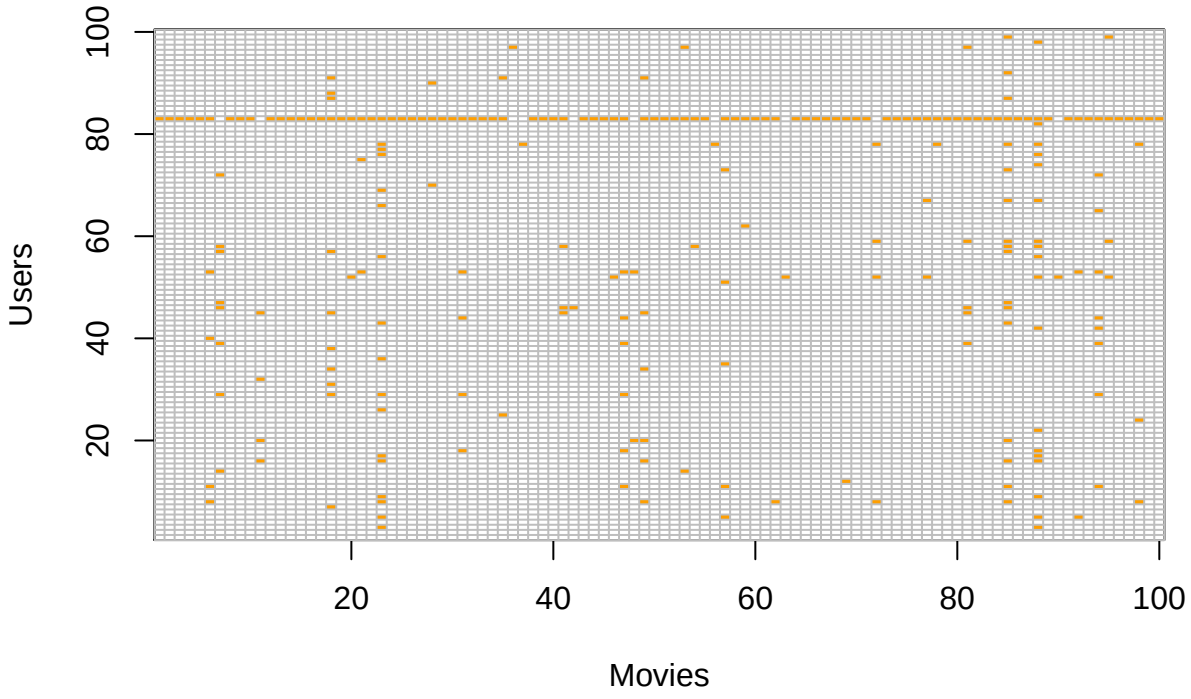
### Unique users that provided ratings and how many unique movies were rated

```
##   n_users n_movies
## 1   69878    10677
```

### Number of ratings in dataset

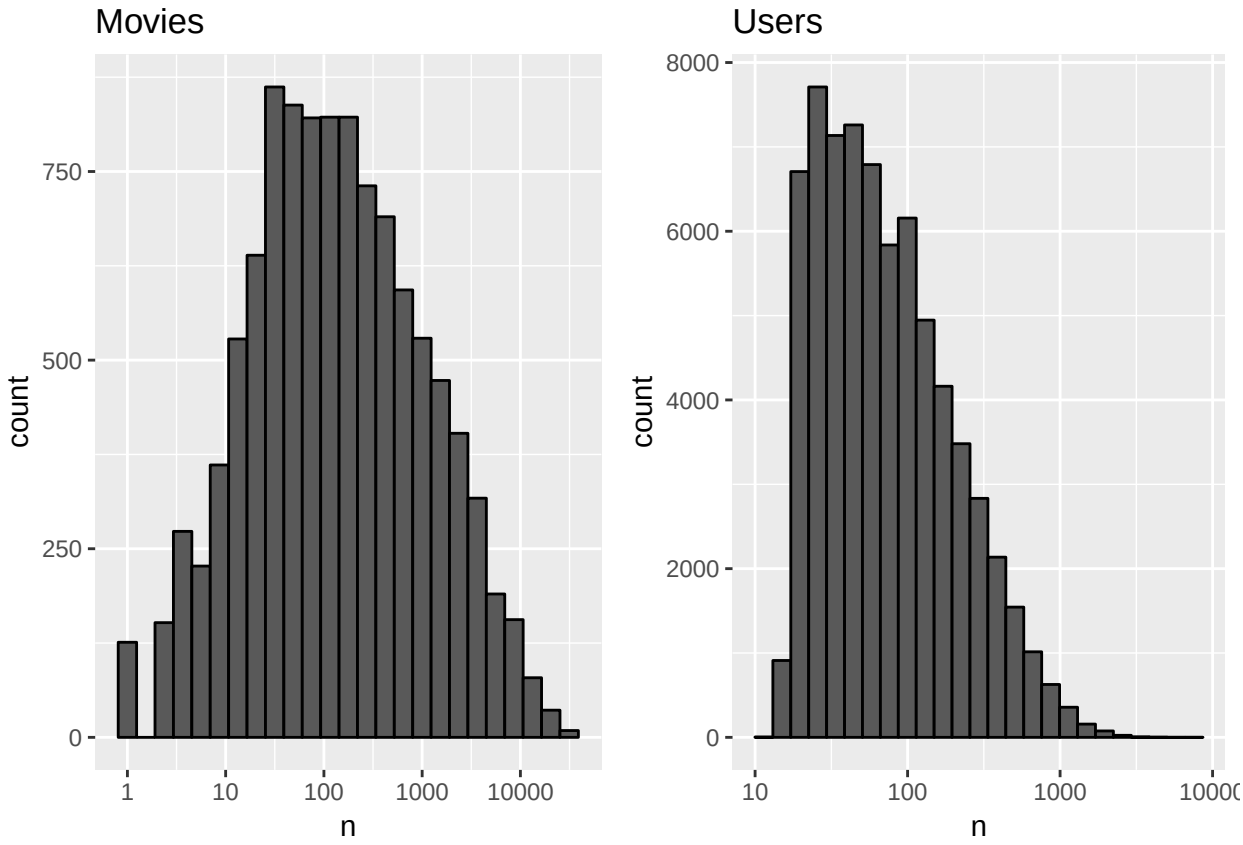
```
## [1] 9000055
```

The product of the two numbers is far greater than the rows in our dataset which shows that often users do not rate movies. We can visualize this by taking a random sample.



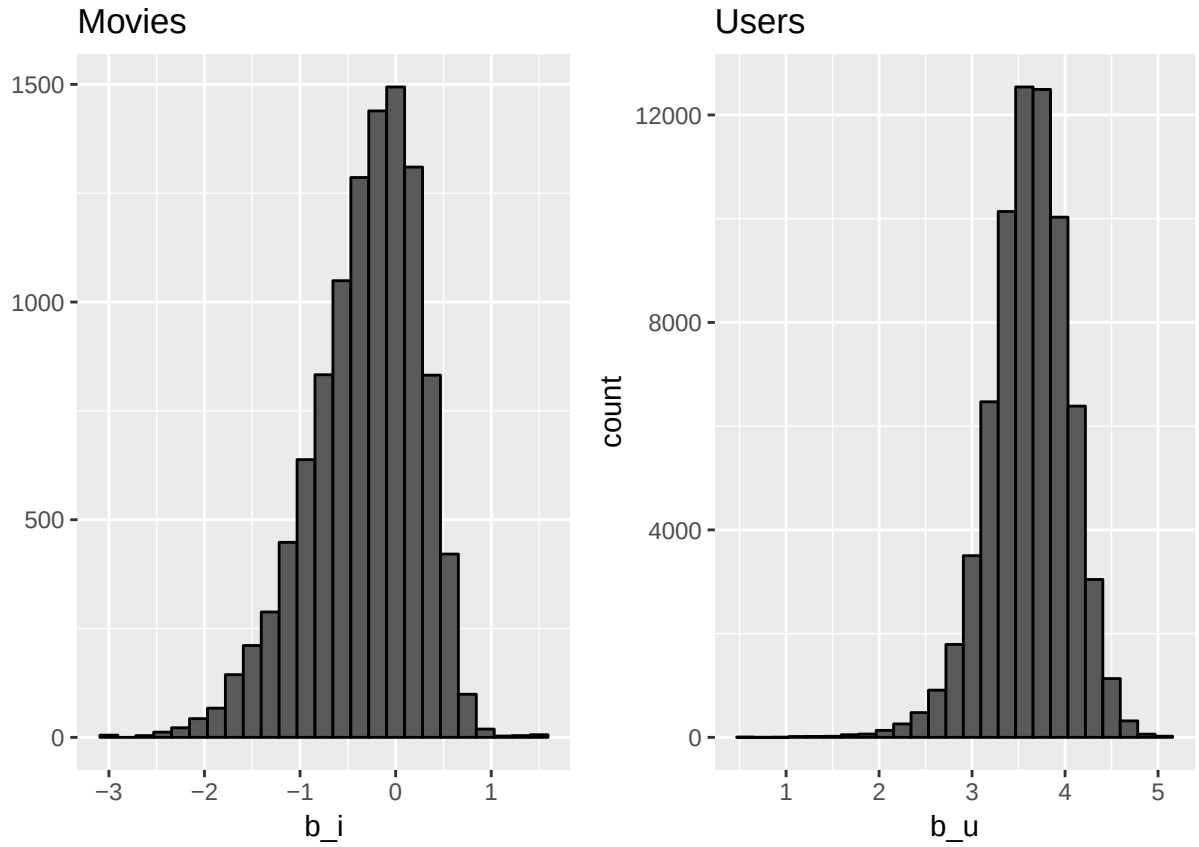

This matrix is quite sparse and shows that are a lot of missing ratings.

Next, the distribution of ratings for movies and users is analyzed and displayed in logscale.




These graphs show that some movies are rated more than others and that some users rate more often than others, with the opposite being true as well.

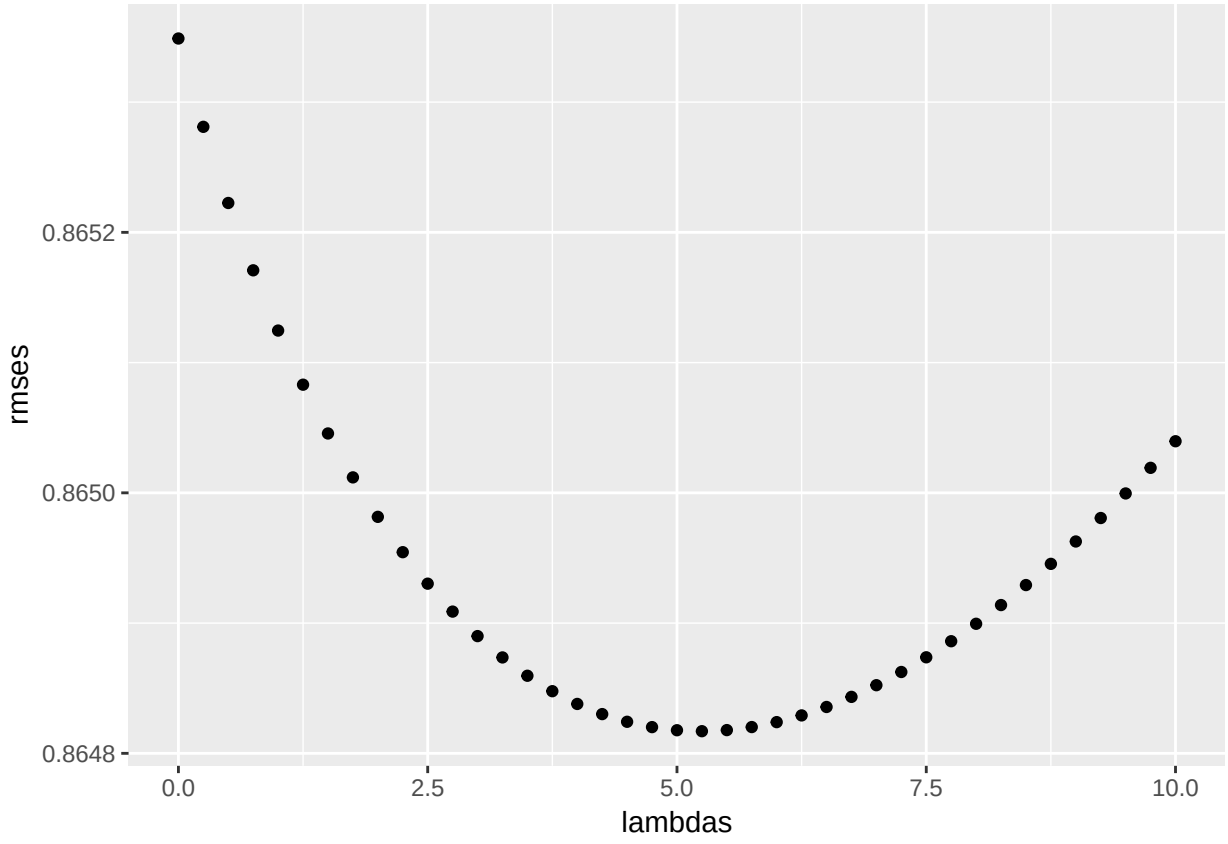
Let's compare how individual movies and users stack up vs. the average rating.




These graphs show that some movies receive higher ratings above/below the average than others and that some users rate higher/lower than the average for any given movie.

It is apparent that there are user and movie effects, or biases, that can help build a good model. However, these biases must account for smaller sample sizes. A movie could have the highest rating, but have only been rated once. Taking it at face value would not help our model. Therefore, regularization will be used to penalize these smaller sample sizes. The formulas for movie effects and user effects are the following respectively, b_i = sum(rating - mu)/(n()+l) and b_u = sum(rating - b_i - mu)/(n()+l). Since the loss function was RMSE, the goal is to minimize it. The specific cut-off to achieve full credit is RMSE <= 0.87750.

The model uses cross validation to find the optimal penalty parameter, lambda, in the range of 0-10 by .25 that will minimize the RMSE.




```
## [1] 5.25
```

# Results

| method | RMSE |
|---|---|
| Just the Average | 1.060331 |
| Regularized Movie + User Effect Model | 0.864817 |

The model performs significantly better than the baseline and is below the cut-off for full-credit.

# Conclusion

The training and validation datasets were provided. Exploratory data analysis revealed useful biases in movie and user ratings and by reason regularization was used since the sample sizes can be very small. Using these, a model was produced that meets the RMSE requirement.