Analysis of the Movie lens dataset

Introduction

The object of this analysis is to construct a model for a recommendation system using data given from the Movielens dataset. The data consists of 6 variables from 10 million observations and each observation has the following characteristics:

- **userId**: An integer value assigned to an individual user with a range of **1-71657**
- **movieId**: A numeric value assigned to each movie with a range of **1-65133**
- **rating**: A numeric value given by the user for a particular user ranging from **1-5** in **.5** increments
- **title**: A character variable encoding the movie title
- **generes**: A character variable assigned to a movie that designates the genre to which the movie belongs, there are **797** distinct genres
- **timestamp**: The date and time at which the movie was reviewed

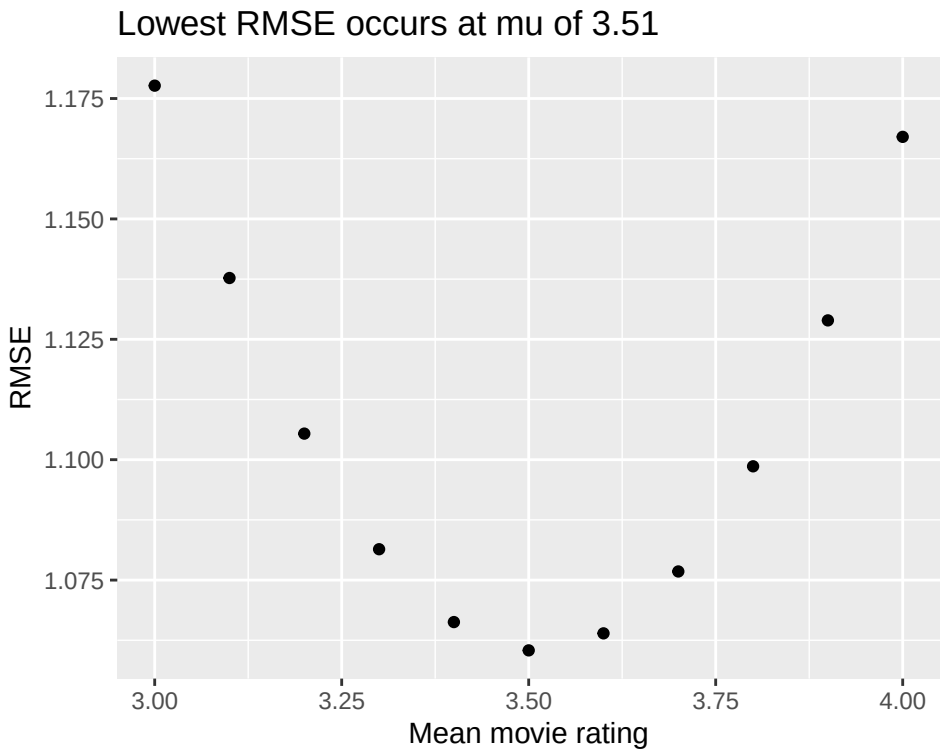
Further, all movies in the dataset have at least one recorded review and each user in the dataset has reviewed at least one movie, all movies have a genre category assigned to it.

Ninety percent of the data was used as a training set and used to train a simple linear model using penalized least squares estimates

Analysis

We will attempt to construct a linear model using root mean squared error as a metric. We chose an x-intercept for the linear equation that is equal to the average rating of all movies, and then verify it is the optimal choice by plotting it against the root mean square error on the training set.

```
mu<-mean(edx$rating)

x<-data.frame(mean=seq(3.0,4.0,by=.1))
y<-data.frame(RMSE=x%>%group_by(mean)%>%summarize(RMSE=RMSE(edx$rating,mean))%>%pull(RMSE))

qplot(x$mean,y$RMSE,xlab="Mean movie rating",ylab="RMSE",main="Lowest RMSE occurs at mu of 3.51")
```

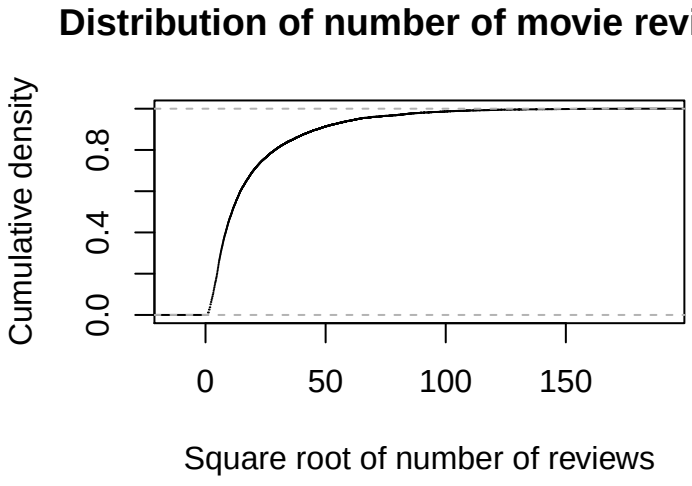


For the variable movieId, we can see that the majority of reviews are relatively small in number and stray from the mean in inverse proportion to the number of reviews

```
edx%>%group_by(movieId)%>%summarize(n=n())%>%.$n%>%quantile(probs=seq(0,1,.25))
```

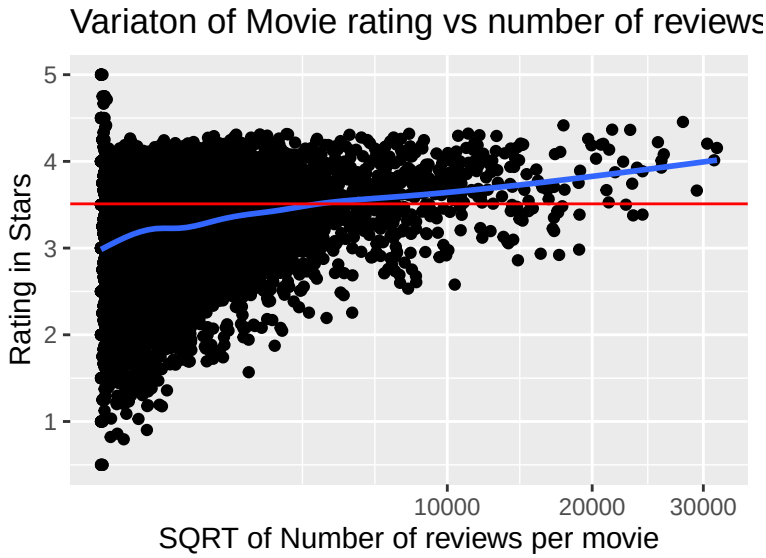
```
##    0%   25%   50%   75%  100%
##     1    30   122   565 31362
```

```
n<-edx%>%group_by(movieId)%>%summarize(n=n())
plot(ecdf(sqrt(n$n)),xlab="Square root of number of reviews",ylab="Cumulative density",main="Distributi
```



Note that in general, as the number of movie reviews per movie increase, we see an increase in the movie rating. Thus more popular movies tend to have a higher rating.

```
edx%>%group_by(movieId)%>%summarize(stars=mean(rating),n=n())%>%ggplot(aes(y=stars,x=n,))+geom_point()+
```

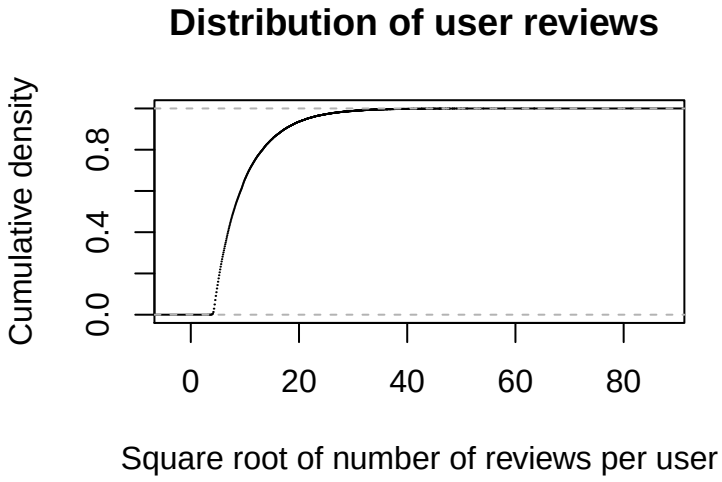


For the userId variable, we see that the majority also consist of relatively small number of user reviews per user

```
edx%>%group_by(userId)%>%summarize(n=n())%>%.$n%>%quantile(prob=seq(0,1,0.25))
```

```
##   0%  25%  50%  75% 100%
##   10   32   62  141 6616
```

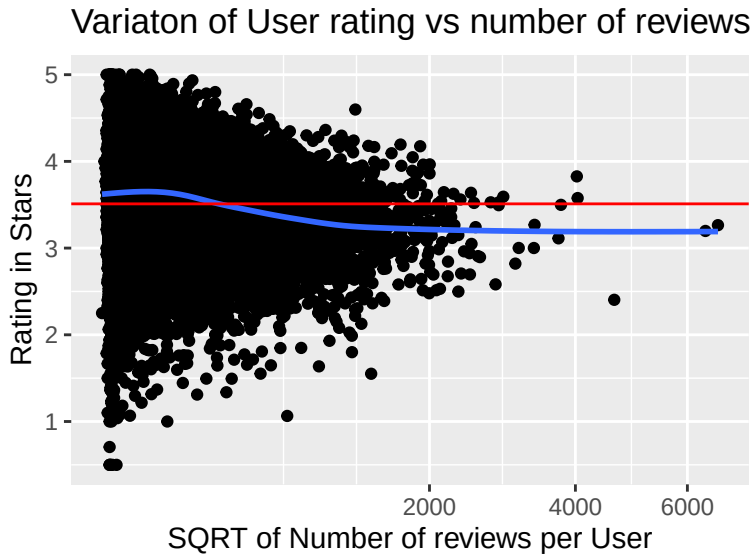
```
n<-edx%>%group_by(userId)%>%summarize(n=n())
plot(ecdf(sqrt(n$n)),xlab="Square root of number of reviews per user",ylab="Cumulative density",main="D
```



As the number of movie reviews per user increased we see the average movie rating drop below the mean. Thus users who rate more movies tend to give below average ratings.

=

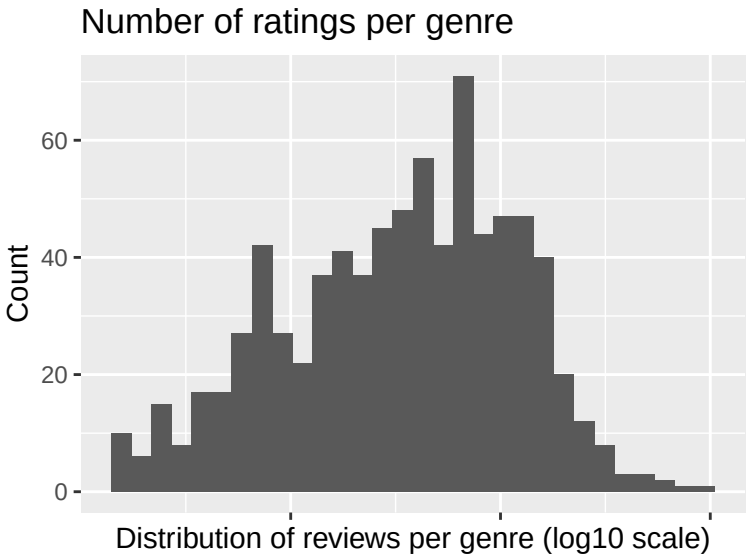
```
edx%>%group_by(userId)%>%summarize(stars=mean(rating),n=n())%>%ggplot(aes(y=stars,x=n,))+geom_point()+g
```

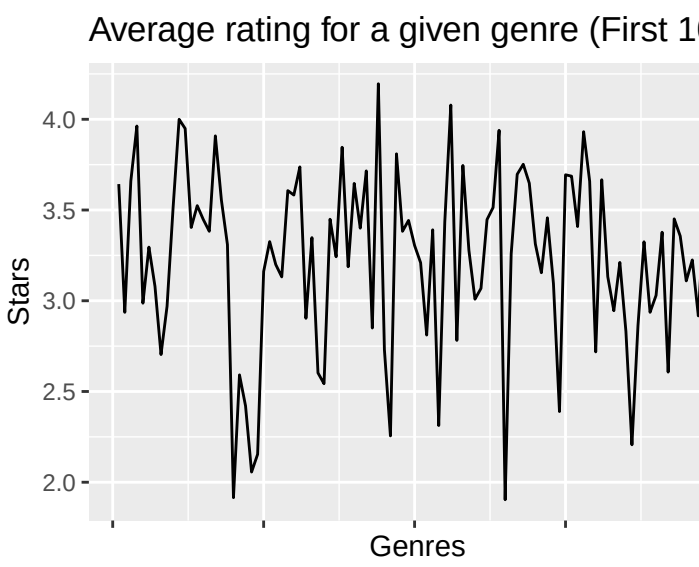


For the genres variable we see variation in genre membership as well as considerable variation in rating per category

```
edx%>%group_by(genres)%>%summarize(n=n())%>%ggplot(aes(n))+geom_histogram()+scale_x_log10()+ylab("Count
```

```
edx%>%group_by(genres)%>%summarize(stars=mean(rating),n=n())%>%head(100)%>%ggplot(aes(x=seq(1,100),y=sta
```



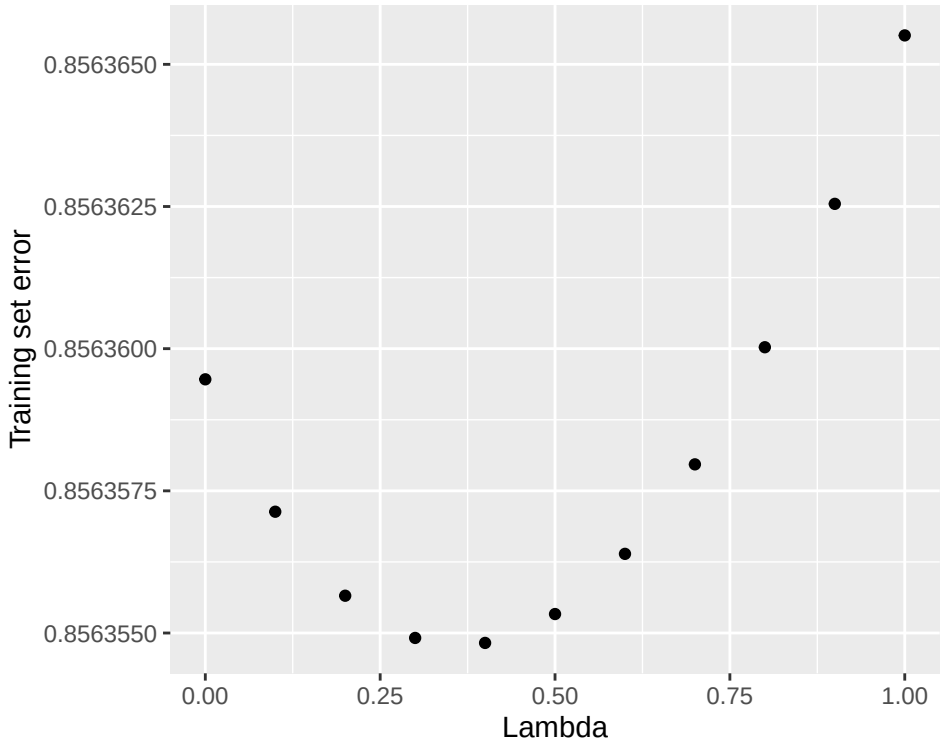


We see considerable variation in average rating per genre and the number of reviews per genre

Results:

A simple linear model was constructed using the movieID,userID, and genre variables. The model was crosstrained and regularized with a lambda of *.4* selected, which give us a RMSE of *.8563* on the training data and a RMSE of *.8648* on the test data

```
lamb<-seq(0,1,.1)
results<-sapply(lamb,function(lmb){
  #for each iteration we take the movieset average rating
  mu<-mean(edx$rating)
  #now take the average of differences and divide by lambda
  reg_movie_bias<-edx%>%group_by(movieId)%>%summarize(movie_bias=sum(rating-mu)/(n()+lmb))

  reg_user_bias<-edx%>%left_join(reg_movie_bias,by="movieId")%>%group_by(userId)%>%summarize(user_bias=

  genre_bias<-edx%>%left_join(reg_movie_bias,by="movieId")%>%left_join(reg_user_bias,by="userId")%>%grou

  predicted_ratings_training<-edx%>%left_join(reg_movie_bias,by="movieId")%>%left_join(reg_user_bias,by=

  RMSE(edx$rating,predicted_ratings_training)

})

data.frame(lamb,results)%>%ggplot(aes(x=lamb,y=results))+geom_point()+ylab("Training set error")+xlab("L
```




```
reg_movie_bias<-edx%>%group_by(movieId)%>%summarize(movie_bias=sum(rating-mu)/(n()+4))
reg_user_bias<-edx%>%left_join(reg_movie_bias,by="movieId")%>%group_by(userId)%>%summarize(user_bias=su
reg_genre_bias<-edx%>%left_join(reg_movie_bias,by="movieId")%>%left_join(reg_user_bias,by="userId")%>%g

predicted_ratings_training<-edx%>%left_join(reg_movie_bias,by="movieId")%>%left_join(reg_user_bias,by="

edx<-edx%>%mutate(predicted=predicted_ratings_training,error=rating-predicted_ratings_training)

predicted_ratings<-validation%>%left_join(reg_movie_bias,by="movieId")%>%left_join(reg_user_bias,by="us
```

Validation set error:

```
RMSE(validation$rating,predicted_ratings)
```

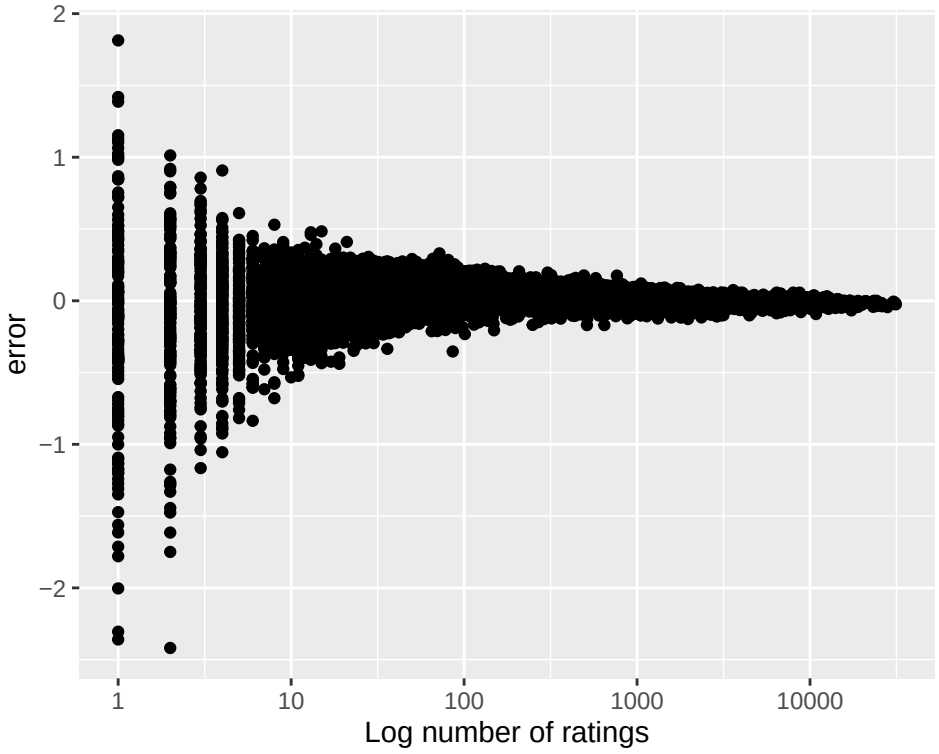
```
## [1] 0.8644656
```

Conclusions:

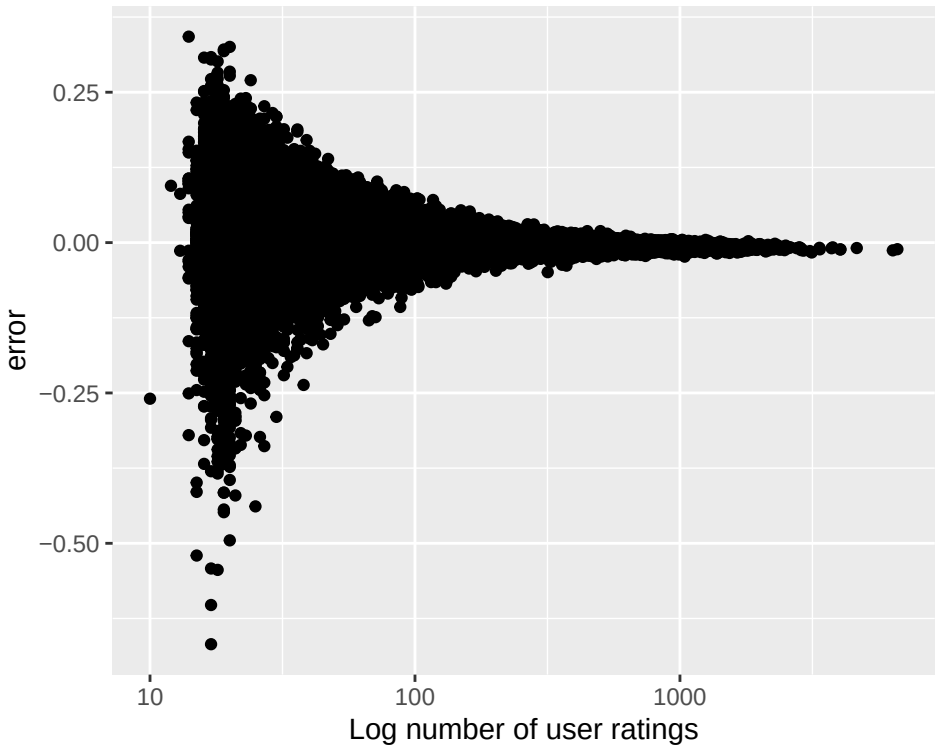
As expected, the model performed with less accuracy on movies with a lower number of reviews despite the normalization. There was generally less error across the userid variable and across the genre variable, indicating that there may be some movie to movie interaction not accounted for in this model. Further study and modeling to account for this effect will be necessary.

The day, week variable extracted from the timestamp variable did not correlate well with the rating variable and did not contribute significantly to our RMSE and were not included in the model.

```
edx%>%group_by(movieId)%>%summarize(number=n(),error=mean(error))%>%ggplot(aes(x=number,y=error))+geom_
```




```
edx%>%group_by(userId)%>%summarize(number=n(),error=mean(error))%>%ggplot(aes(x=number,y=error))+geom_p
```



Top ten error by genre:

```
edx%>%group_by(genres)%>%summarise(n=n(),error=mean(error))%>%arrange(desc(abs(error)))%>%head(10)
```

```
## # A tibble: 10 x 3
##    genres                                          n  error
##    <chr>                                       <int>  <dbl>
##  1 Action|Animation|Comedy|Horror                  2 -0.613
##  2 Adventure|Fantasy|Film-Noir|Mystery|Sci-Fi      2  0.496
##  3 Action|War|Western                              2  0.354
##  4 Action|Drama|Horror|Sci-Fi                      4 -0.342
##  5 Action|Adventure|Animation|Comedy|Sci-Fi        3  0.227
##  6 Animation|Documentary|War                       4  0.215
##  7 Fantasy|Horror|Sci-Fi                           6 -0.197
##  8 Animation|IMAX|Sci-Fi                           7  0.170
##  9 Drama|Musical|Thriller                          4  0.152
## 10 Adventure|Comedy|Fantasy|Romance                6 -0.137
```

Correlations of day and week and month vs rating

```
cor(edx$day,edx$rating)
```

```
## [1] 0.0003036543
```

```
cor(edx$week,edx$rating)
```

```
## [1] 0.01212409
```

```r
cor(edx$month,edx$rating)
```

```
## [1] 0.01197937
```