Movie Lens

Linda Jones
6/12/2019

#INTRODUCTION

This report is being submitted to satisfy the criteria of the 'HarvardX: PH125.9x Data Science Capstone project. Using the prescribed data from the 10M version of the MovieLens dataset, an algorithm was developed to predict movie ratings. The edx set and validation set were employed in the algorithm. The capstone project instructions directed the use of Root Mean Square Error (RMSE) to forecast and assess predictions against the validation set.

#OVERVIEW

This capstone project provides students with an opportunity to demonstrate their mastery of the course material by applying thier skills using R and RStudio.The goal of this project is to develop a machine learning algorithm to predict movie ratings. This report has five parts: (1) Dataset Description, (2) Data Exploration, (3) Methods & Analysis, (4) Results; and (5) Conclusion.

The data's dimensions and attributes are assessed in the dataset exploration section to ensure the dataset is suitable for the intended purpose. In the data exploration section, data attributes, such as the summary, class, number of columns or rows, are explored. The methods employed to predict the ratings and the corresponding analysis are presented in the methods and analysis section. The final sections of the report present the results and conclusion.

#DATASET DESCRIPTION

A prescribed dataset was used for this project and the code was provided in the course instructions. The edx and validation sets were also already created.

#Load the dataset

```
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us
.r-project.org")

## Loading required package: tidyverse

## -- Attaching packages ---------------------------------------------------
--------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.1
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

## -- Conflicts ------------------------------------------------------------
--------------------------------- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-proje
ct.org")
```

```
## Loading required package: caret
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
dl <- tempfile()
download.file("http://files.grouplens.org/datasets/movielens/ml-10m.zip", dl)

ratings <- read.table(text = gsub("::", "\t", readLines(unzip(dl, "ml-10M100K
/ratings.dat"))),
                      col.names = c("userId", "movieId", "rating", "timestamp
"))

movies <- str_split_fixed(readLines(unzip(dl, "ml-10M100K/movies.dat")), "\\:
:", 3)
colnames(movies) <- c("movieId", "title", "genres")
movies <- as.data.frame(movies) %>% mutate(movieId = as.numeric(levels(movieI
d))[movieId],
                                           title = as.character(title),
                                           genres = as.character(genres))

movielens <- left_join(ratings, movies, by = "movieId")

set.seed(1)
test_index <- createDataPartition(y = movielens$rating, times = 1, p = 0.1, l
ist = FALSE)
edx <- movielens[-test_index,]
temp <- movielens[test_index,]

validation <- temp %>%
     semi_join(edx, by = "movieId") %>%
     semi_join(edx, by = "userId")

removed <- anti_join(temp, validation)
```

```
## Joining, by = c("userId", "movieId", "rating", "timestamp", "title", "genr
es")
```

```
edx <- rbind(edx, removed)

rm(dl, ratings, movies, test_index, temp, movielens, removed)
```

The following libraries were loaded: library(recommenderlab), library(ggplot2), library(data.table), library(reshape2), and library(devtools)

What follows are the stpes taken to explore the data. Specifically the variables were examined for accuracy, completeness, emerging patterns and to ensure the data could support a statistical analysis. Results of the examination appear below.

#EXPLORING THE DATA

#Identify the number of columns and rows in the edx data

```
ncol(edx)

## [1] 6

nrow(edx)

## [1] 9000055
```

#Look at a summary of the edx data

```
summary(edx)

##      userId         movieId          rating        timestamp
##  Min.   :    1   Min.   :    1   Min.   :0.500   Min.   :7.897e+08
##  1st Qu.:18124   1st Qu.:  648   1st Qu.:3.000   1st Qu.:9.468e+08
##  Median :35738   Median : 1834   Median :4.000   Median :1.035e+09
##  Mean   :35870   Mean   : 4122   Mean   :3.512   Mean   :1.033e+09
##  3rd Qu.:53607   3rd Qu.: 3626   3rd Qu.:4.000   3rd Qu.:1.127e+09
##  Max.   :71567   Max.   :65133   Max.   :5.000   Max.   :1.231e+09
##     title              genres
##  Length:9000055     Length:9000055
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

#Class identification for the edx data

```
class(edx)

## [1] "data.frame"
```

#Review edx structure

```
str(edx)
```

```
## 'data.frame':    9000055 obs. of  6 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : num  122 185 292 316 329 355 356 362 364 370 ...
##  $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ timestamp: int  838985046 838983525 838983421 838983392 838983392 83898
4474 838983653 838984885 838983707 838984596 ...
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)"
"Stargate (1994)" ...
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|
Sci-Fi|Thriller" "Action|Adventure|Sci-Fi" ...
```

#Identify the number of columns and rows in the validation data

```
ncol(validation)

## [1] 6

nrow(validation)

## [1] 999999
```

#Look at a summary of the validation data

```
summary(validation)

##      userId         movieId          rating        timestamp
##  Min.   :    1   Min.   :    1   Min.   :0.500   Min.   :7.897e+08
##  1st Qu.:18096   1st Qu.:  648   1st Qu.:3.000   1st Qu.:9.467e+08
##  Median :35768   Median : 1827   Median :4.000   Median :1.035e+09
##  Mean   :35870   Mean   : 4108   Mean   :3.512   Mean   :1.033e+09
##  3rd Qu.:53621   3rd Qu.: 3624   3rd Qu.:4.000   3rd Qu.:1.127e+09
##  Max.   :71567   Max.   :65133   Max.   :5.000   Max.   :1.231e+09
##     title              genres
##  Length:999999      Length:999999
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

#Class identification for the validation data

```
class(validation)

## [1] "data.frame"
```

#Review validation structure

```
str(validation)

## 'data.frame':    999999 obs. of  6 variables:
##  $ userId   : int  1 1 1 2 2 2 3 3 4 4 ...
##  $ movieId  : num  231 480 586 151 858 ...
```

```
## $ rating   : num  5 5 5 3 2 3 3.5 4.5 5 3 ...
## $ timestamp: int  838983392 838983653 838984068 868246450 868245645 86824
5920 1136075494 1133571200 844416936 844417070 ...
## $ title    : chr  "Dumb & Dumber (1994)" "Jurassic Park (1993)" "Home Alo
ne (1990)" "Rob Roy (1995)" ...
## $ genres   : chr  "Comedy" "Action|Adventure|Sci-Fi|Thriller" "Children|C
omedy" "Action|Drama|Romance|War" ...
```

VISUALIZING DATA

In this section the data is visualized. Visualization, when done well, facilitates communciation of information in a simple and easy to understand format. For the purpposes of this project the visualization is being used to learn about the relationsips, if any, between the users, movies and the associated ratings. There are 9000055 therefore, the top 15 movies are explored further. Upon review they do not appear to be in the same genre.

DATA EXPLORATION

```
library(dplyr)
length(edx$movieId)

## [1] 9000055
```

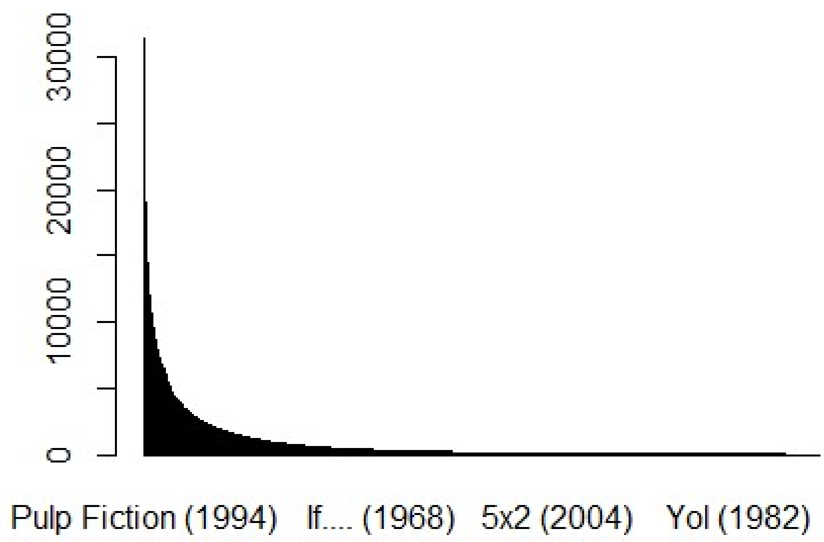
#Identify the top 15 movies

```
top_15movies<-edx%>%group_by(title)%>%summarize(count=n())%>%top_n(15,count)%
>%arrange(desc(count))
top_15movies

## # A tibble: 15 x 2
##    title                                                        count
##    <chr>                                                        <int>
##  1 Pulp Fiction (1994)                                          31362
##  2 Forrest Gump (1994)                                          31079
##  3 Silence of the Lambs, The (1991)                             30382
##  4 Jurassic Park (1993)                                         29360
##  5 Shawshank Redemption, The (1994)                             28015
##  6 Braveheart (1995)                                            26212
##  7 Fugitive, The (1993)                                         25998
##  8 Terminator 2: Judgment Day (1991)                            25984
##  9 Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977) 25672
## 10 Apollo 13 (1995)                                             24284
## 11 Batman (1989)                                                24277
## 12 Toy Story (1995)                                             23790
## 13 Independence Day (a.k.a. ID4) (1996)                         23449
## 14 Dances with Wolves (1990)                                    23367
## 15 Schindler's List (1993)                                      23193
```

#plot Top 15 Movies

```
plottop15movies<-table(edx$title)
barplot(plottop15movies[order(plottop15movies,decreasing = TRUE)])
```



Additionally, the movies by title are concentrated in one section and are skewed to the left. An evaluation of the top 15 movies by title and rank follow in order to assess the number of ratings for the top 15 movies.

#Plot the Top 15 Movies by Ratings

```
top_15movies%>%ggplot(aes(x=reorder(title, count),y=count))+geom_bar(stat='id
entity',fill="orange")+coord_flip(y=c(0, 35000))+
labs(x="", y="# of Ratings \n Figure 2")+geom_text(aes(label= count),hjust=-0
.1, size=3)+labs(title="Top 15 Movies Titles")
```

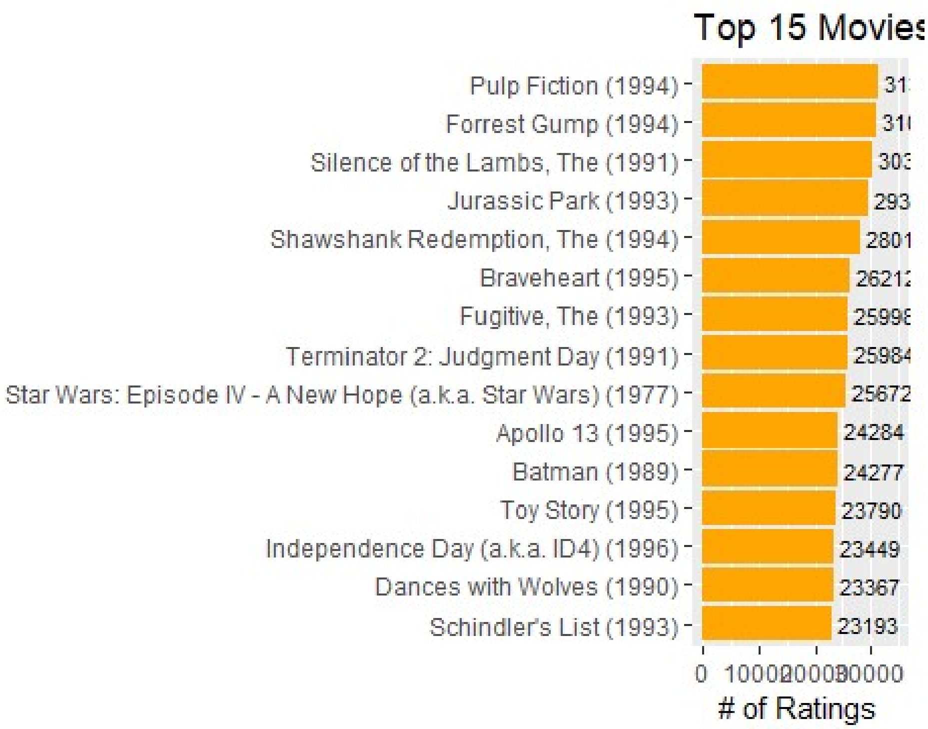


Figure 2

A review of the genres follows to see determine if any trends or patters emerge.

```
edx %>% group_by(genres) %>%summarize(n = n(), avg = mean(rating), se = sd(ra
ting)/sqrt(n())) %>%filter(n >= 100000)

## # A tibble: 14 x 4
##    genres                                  n   avg      se
##    <chr>                               <int> <dbl>   <dbl>
##  1 Action|Adventure|Sci-Fi            219938  3.51 0.00233
##  2 Action|Adventure|Sci-Fi|Thriller 105144  3.54 0.00299
##  3 Action|Adventure|Thriller          149091  3.43 0.00246
##  4 Action|Crime|Thriller              102259  3.46 0.00319
##  5 Comedy                             700889  3.24 0.00133
##  6 Comedy|Drama                       323637  3.60 0.00175
##  7 Comedy|Drama|Romance               261425  3.65 0.00192
##  8 Comedy|Romance                     365468  3.41 0.00171
##  9 Crime|Drama                        137387  3.95 0.00244
## 10 Crime|Drama|Thriller               106101  3.78 0.00277
## 11 Drama                              733296  3.71 0.00114
## 12 Drama|Romance                      259355  3.61 0.00203
## 13 Drama|Thriller                     145373  3.45 0.00252
## 14 Drama|War                          111029  3.98 0.00271

top_15movies_by_genres<-edx%>%group_by(title,genres)%>%summarize(count=n())%>
%top_n(15,count)%>%arrange(desc(count))
top_15movies_by_genres
```

```
## # A tibble: 10,677 x 3
## # Groups:   title [10,676]
##    title                                          genres                   coun
t
##    <chr>                                          <chr>                    <int
>
##  1 Pulp Fiction (1994)                            Comedy|Crime|Drama       3136
2
##  2 Forrest Gump (1994)                            Comedy|Drama|Romance|W~ 3107
9
##  3 Silence of the Lambs, The (1991)               Crime|Horror|Thriller    3038
2
##  4 Jurassic Park (1993)                           Action|Adventure|Sci-F~ 2936
0
##  5 Shawshank Redemption, The (1994)               Drama                    2801
5
##  6 Braveheart (1995)                              Action|Drama|War         2621
2
##  7 Fugitive, The (1993)                           Thriller                 2599
8
##  8 Terminator 2: Judgment Day (1991)              Action|Sci-Fi            2598
4
##  9 Star Wars: Episode IV - A New Hope (a.k.a~ Action|Adventure|Sci-Fi 2567
2
## 10 Apollo 13 (1995)                               Adventure|Drama          2428
4
## # ... with 10,667 more rows
```

#plot the movies by rating

```
hist(edx$rating,main="Movies by Ratings",xlab="Ratings\n Figure 3",col="yello
w")
```

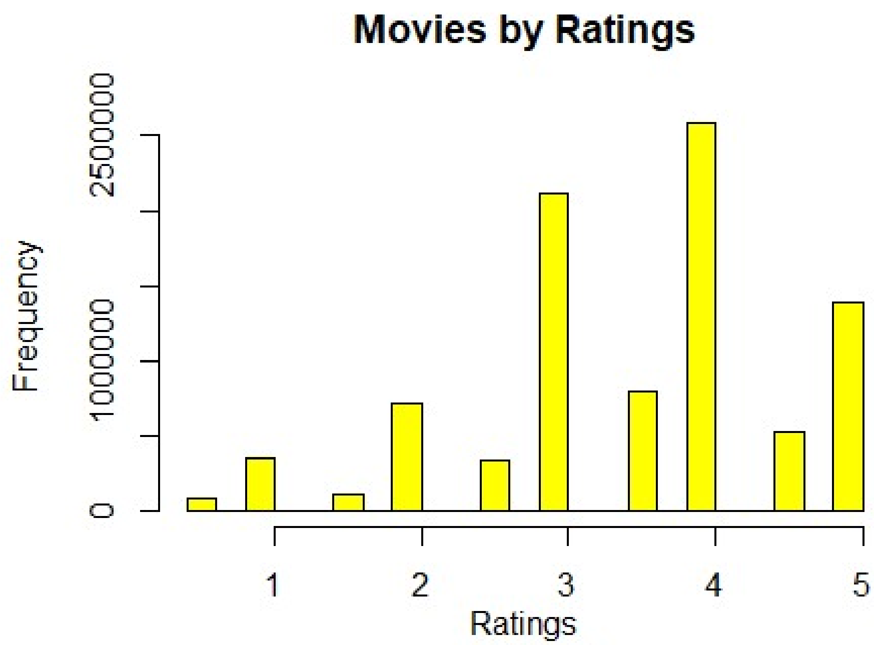


Figure 3

The movie ratings are not evenly distributed. Majority of the movies receive a rating between 3 and 4.

#plot users by rating

```
edx %>%
  count(userId) %>%
  ggplot(aes(n)) +
  geom_histogram(bins = 10, color = "blue") +
  scale_x_log10() +
  ylab("# of Users") +
  xlab("# of Ratings \n Figure 4") +
ggtitle("# of Ratings by Users")
```

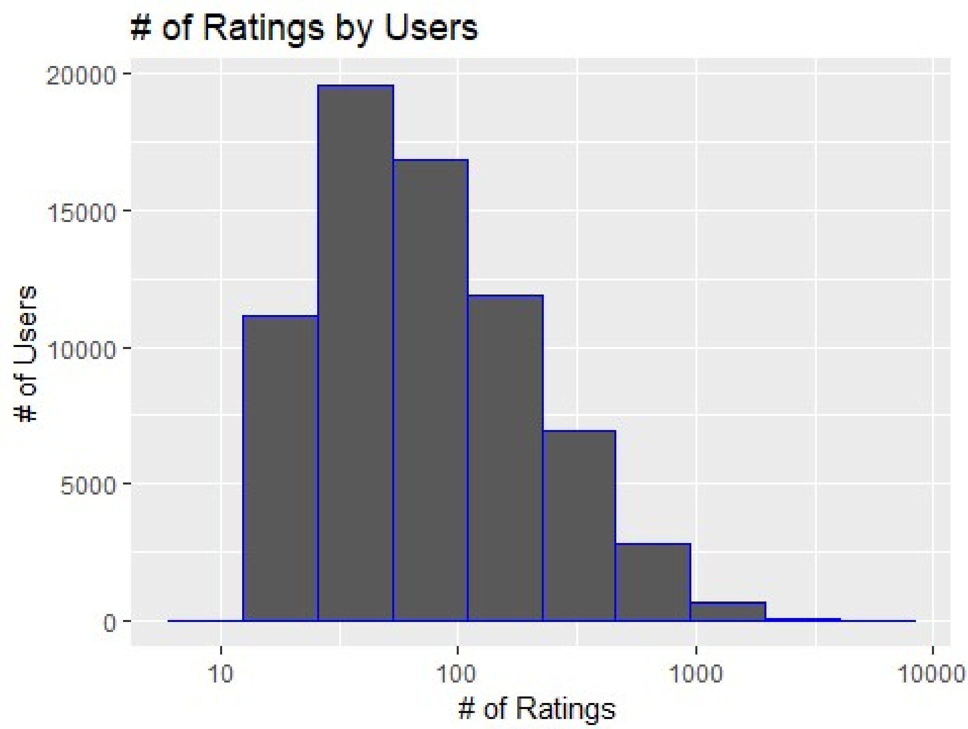


Figure 4

```
hist(edx$movieId,main="Movies by MovieID",xlab="Ratings\n Figure 5",col="yellow")
```

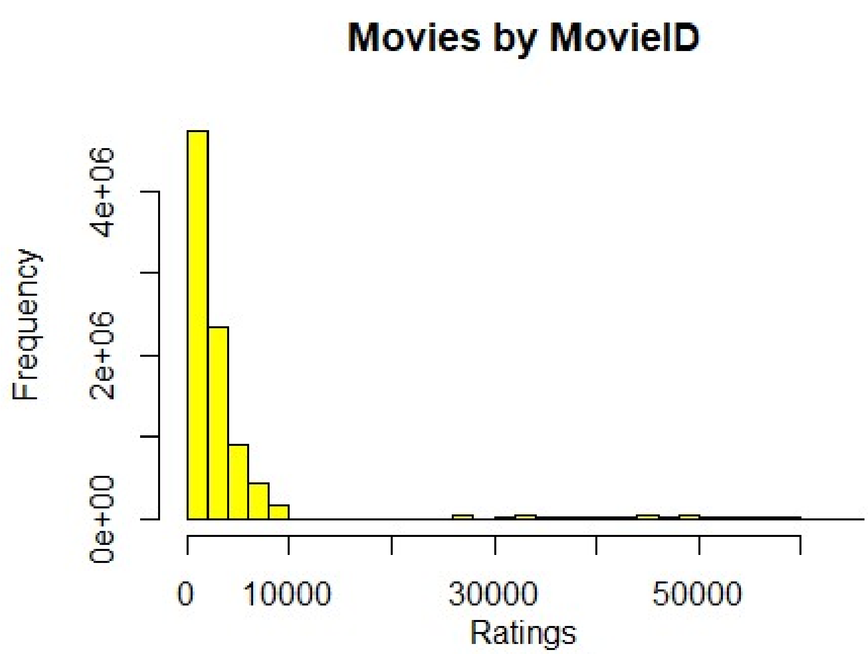


Figure 5

#METHOD & ANALYSIS

The capstone project criteria required use of the Root Mean Square Error (RMSE) to assess predictions. RMSE is a common statistical measure used to assess the standard deviation or errors. Two methods were used to predict movie ratings: Movie Effect Model (MEM) and Movie and User Effect Models. RMSE was calculated by finding the average of all movies

#calculate baseline RMSE

```
mu<-mean(edx$rating)
baseline_rmse <-RMSE(validation$rating, mu)
rmse_results <- data_frame(method = "Model - Average Movie Ratings", RMSE = b
aseline_rmse)

## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.

rmse_results

## # A tibble: 1 x 2
##   method                         RMSE
##   <chr>                         <dbl>
## 1 Model - Average Movie Ratings  1.06
```

The RMSE that will be used for comparision purposes is Model - Average Movie Ratings, 1.0612018.

#Gererate the Movie Effect Method (MEM) by calculating the estimated deviation by calculating the averge (mean) of all movies (edx$rating) on the training set and then calculating the predicted rating. Note: “mu” already calculated above

```
avg_of_movie_ratings<-edx%>%group_by(movieId)%>%summarize(b_i=mean(rating-mu)
)
pred_movie_ratings_model1<-mu+validation%>%left_join(avg_of_movie_ratings,by=
"movieId")%>%.$b_i
mem_rmse<-RMSE(validation$rating,pred_movie_ratings_model1)
mem_rmse

## [1] 0.9439087
```

The RMSE for the Movie Effect Method is 0.9439087.

#Gererate the Movie and User Effect Method (MUEM) by calculating the estimated deviation by calculating the averge (mean) of all moves(edx$rating) on the training set and then calculating the predicted rating. Note: “mu” already calculated above

```
avg_of_users<-edx%>%left_join(avg_of_movie_ratings,by="movieId")%>%group_by(u
serId)%>%summarize(b_u=mean(rating-mu-b_i))
pred_movie_ratings_model2<-validation%>%left_join(avg_of_movie_ratings,by="mo
vieId")%>%left_join(avg_of_users,by="userId")%>%mutate(pred=mu+b_i+b_u)%>%.$p
red
```

```
muem_rmse<-RMSE(validation$rating,pred_movie_ratings_model2)
muem_rmse

## [1] 0.8653488
```

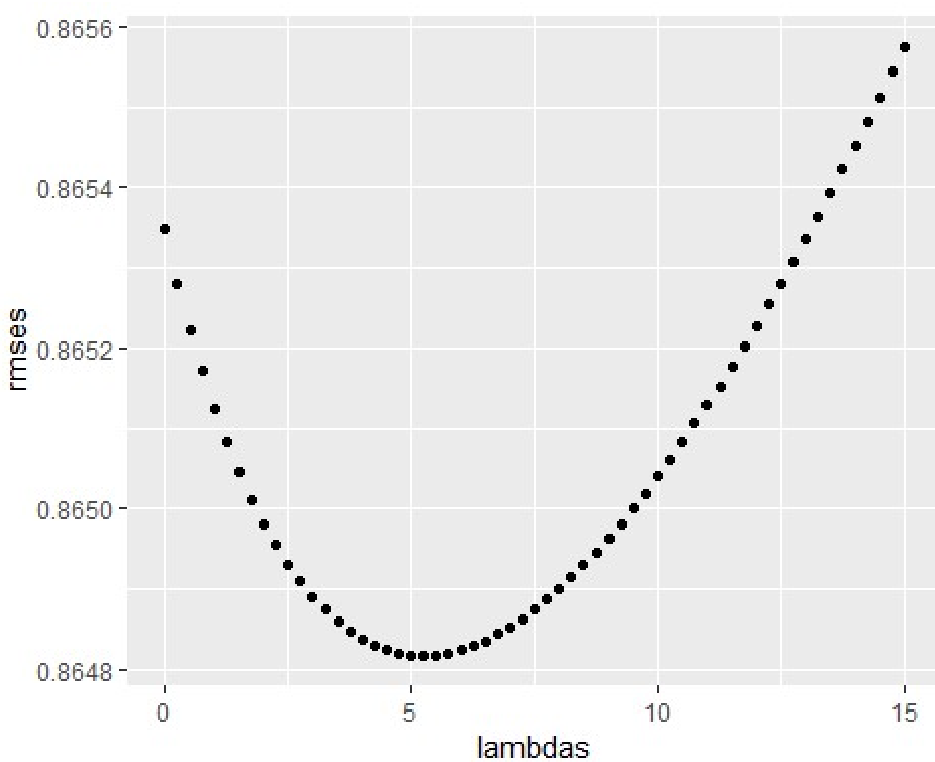
The RMSE for the Movie and User Effect Method is 0.8653488

#Assess model complexity and minimize overfitting with regularization

```
rmse<-function(actual_rating,rating_prediction){sqrt(mean((actual_rating-rati
ng_prediction)^2))}
lambdas<-seq(0,15,.25)

rmses<-sapply(lambdas,function(l){
mu<-mean(edx$rating)
b_i<-edx %>%group_by(movieId)%>%summarize(b_i=sum(rating-mu)/(n()+l))
b_u<-edx %>%left_join(b_i,by="movieId")%>%group_by(userId) %>%summarize(b_u=s
um(rating-b_i-mu)/(n()+l))
prediction<-validation%>%left_join(b_i,by="movieId")%>%left_join(b_u,by="user
Id")%>%mutate(pred=mu+b_i+b_u)%>%pull(pred)
return(RMSE(prediction,validation$rating))
})

qplot(lambdas,rmses)
```




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

```
## [1] 5.25

rmse_finalresult<-min(rmses)
rmse_finalresult

## [1] 0.864817
```

The rmses is `r rmse_finalresult'.

#RESULTS

When comparing the baseline RMSE to each method we find that: Movie Effect Method: is 0.9439087 Movie and User Effect: 0.8653488 The lambdas is 5.25 and the rmses is 0.864817.

#CONCLUSION

RMSE does not have perscribed values that wee seek to met. Instead we use RMSE as a comparative value that should be low and close to the regression line. Based on the findings the models demonstrated that the algorithm can predit movie ratings with a good level of accuracy.