Movielens Recommender Project

Ratna Ray

6/7/2019

Contents

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

Overview

This project has been implemented as part of the Data Science: Capstone project in edX. This report has all details related to the project and the thought process behind the final objective of a recommender model.

We start with a basic introduction and objective. This is followed by basic data analysis and cleaning for preparation. An exploratory data analysis is carried out in order to develop multiple machine learning models that can predict movie ratings and then finalize a model. The report ends with an explanation of the results and a conclusion.

Introduction

Recommendation systems use ratings that users have given to items to make specific recommendations. Companies that sell many products to many customers and permit these customers to rate their products, like Amazon, are able to collect massive datasets that can be used to predict what rating a particular user will give to a specific item. Items for which a high rating is predicted for a given user are then recommended to that user.

The same could be done for other items, as movies for instance in our case. Recommendation systems are one of the most used models in machine learning algorithms. In fact the success of Netflix is said to be based on its strong recommendation system. The Netflix prize (open competition for the best collaborative filtering algorithm to predict user ratings for films, based on previous ratings without any other information about the users or films), in fact, represent the high importance of algorithm for products recommendation system.

For this project we will focus on create a movie recommendation system using the 10M version of MovieLens dataset, collected by GroupLens Research.

Aim of the project

The aim in this project is to train a machine learning algorithm that predicts user ratings (from 0.5 to 5 stars) using the inputs of a provided subset.

The value used to evaluate algorithm performance is the Root Mean Square Error, or RMSE. RMSE is one of the most used measure of the differences between values predicted by a model and the values observed. RMSE is a measure of accuracy, to compare forecasting errors of different models for a particular dataset, a lower RMSE is better than a higher one. The effect of each error on RMSE is proportional to the size of the squared error; thus larger errors have a disproportionately large effect on RMSE. Consequently, RMSE is sensitive to outliers. The evaluation criteria for this algorithm is a RMSE expected to be lower than 0.8775.

The best resulting model will be used to predict the movie ratings.

Dataset

For this recommender system, we use a version of the movielens dataset. This is a small subset of a much larger dataset with millions of ratings. We will use the 10M version of the complete MovieLens dataset to make the computation a little easier.

The MovieLens dataset is automatically downloaded from:

- [MovieLens 10M dataset] https://grouplens.org/datasets/movielens/10m/
- [MovieLens 10M dataset - zip file] http://files.grouplens.org/datasets/movielens/ml-10m.zip

MovieLens 10M movie ratings is a stable benchmark dataset. It has 10 million ratings and 100,000 tag applications applied to 10,000 movies by 72,000 users.

Load Dataset

```r
# Note: this process could take a couple of minutes

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

```r
                                     repos = "http://cran.us.r-project.org")


####################################
# Create edx set and validation set
####################################

# MovieLens 10M dataset:
# https://grouplens.org/datasets/movielens/10m/
# http://files.grouplens.org/datasets/movielens/ml-10m.zip

dl <- tempfile()
dl
```

```
## [1] "/var/folders/yc/crffhmc5649blbjdw95z94l00000gn/T//RtmpEJV0rp/filea1421db1116"
```

```r
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(movieId))[movieId],
                                           title = as.character(title),
                                           genres = as.character(genres))

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

In order to predict in the most possible accurate way the movie rating of the users that haven't seen the movie yet, the MovieLens dataset will be split into 2 subsets: • edx: a training subset to train the algorithm • validation: a subset to test the movie ratings

```r
# Validation set will be 10% of MovieLens data

set.seed(1) # if using R 3.6.0: set.seed(1, sample.kind = "Rounding")
test_index <- createDataPartition(y = movielens$rating, times = 1, p = 0.1, list = FALSE)
edx <- movielens[-test_index,]
temp <- movielens[test_index,]

# Make sure userId and movieId in validation set are also in edx set

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

# Add rows removed from validation set back into edx set

removed <- anti_join(temp, validation)
edx <- rbind(edx, removed)

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

Data exploration and analysis

Data Analysis of movies

```r
str(movies)
```

```
## 'data.frame':    10681 obs. of  3 variables:
##  $ movieId: num  1 2 3 4 5 6 7 8 9 10 ...
##  $ title  : chr  "Toy Story (1995)" "Jumanji (1995)" "Grumpier Old Men (1995)" "Waiting to Exhale (19
##  $ genres : chr  "Adventure|Animation|Children|Comedy|Fantasy" "Adventure|Children|Fantasy" "Comedy|R
```

```r
# This has over 10K movies along with the title and associated genre for each movie
```

Summary of movies and several rows of this dataframe:

```r
summary(movies)
```

```
##     movieId          title              genres         
##  Min.   :    1   Length:10681       Length:10681      
##  1st Qu.: 2755   Class :character   Class :character  
##  Median : 5436   Mode  :character   Mode  :character  
##  Mean   :13121                                        
##  3rd Qu.: 8713                                        
##  Max.   :65133
```

```r
head(movies)
```

```
##   movieId                              title
## 1       1                   Toy Story (1995)
## 2       2                     Jumanji (1995)
## 3       3            Grumpier Old Men (1995)
## 4       4           Waiting to Exhale (1995)
## 5       5 Father of the Bride Part II (1995)
## 6       6                          Heat (1995)
##                                         genres
## 1 Adventure|Animation|Children|Comedy|Fantasy
## 2                  Adventure|Children|Fantasy
## 3                              Comedy|Romance
## 4                        Comedy|Drama|Romance
## 5                                      Comedy
## 6                       Action|Crime|Thriller
```

Data Analysis of ratings

```r
str(ratings)
```

```
## 'data.frame':    10000054 obs. of  4 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : int  122 185 231 292 316 329 355 356 362 364 ...
##  $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ timestamp: int  838985046 838983525 838983392 838983421 838983392 838983392 838984474 838983653 83
```

```r
# This has 10M ratings and has been joined with movies to build the movielens dataset
```

Summary of ratings:

```r
summary(ratings)
```

```
##      userId         movieId          rating        timestamp        
##  Min.   :    1   Min.   :    1   Min.   :0.500   Min.   :7.897e+08  
##  1st Qu.:18123   1st Qu.:  648   1st Qu.:3.000   1st Qu.:9.468e+08  
##  Median :35740   Median : 1834   Median :4.000   Median :1.035e+09  
##  Mean   :35870   Mean   : 4120   Mean   :3.512   Mean   :1.033e+09  
##  3rd Qu.:53608   3rd Qu.: 3624   3rd Qu.:4.000   3rd Qu.:1.127e+09  
##  Max.   :71567   Max.   :65133   Max.   :5.000   Max.   :1.231e+09
```

```r
head(ratings)
```

```
##   userId movieId rating timestamp
## 1      1     122      5 838985046
## 2      1     185      5 838983525
## 3      1     231      5 838983392
## 4      1     292      5 838983421
## 5      1     316      5 838983392
## 6      1     329      5 838983392
```

Data Analysis of movielens

```r
str(movielens)
```

```
## 'data.frame':    10000054 obs. of  6 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : num  122 185 231 292 316 329 355 356 362 364 ...
##  $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ timestamp: int  838985046 838983525 838983392 838983421 838983392 838983392 838984474 838983653 8
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Dumb & Dumber (1994)" "Outbreak (1995)" ...
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Comedy" "Action|Drama|Sci-Fi|Thriller"
```


```r
# This has 10M ratings and has been joined with movies to build the movielens dataset
```


The movielens dataset has more than 10 million ratings. Each record is associated with: 1. userId 2. movieId 3. rating 4. timestamp 5. title 6. genres 7. year

Summary of movielens

```r
summary(movielens)
```

```
##      userId         movieId          rating        timestamp        
##  Min.   :    1   Min.   :    1   Min.   :0.500   Min.   :7.897e+08  
##  1st Qu.:18123   1st Qu.:  648   1st Qu.:3.000   1st Qu.:9.468e+08  
##  Median :35740   Median : 1834   Median :4.000   Median :1.035e+09  
##  Mean   :35870   Mean   : 4120   Mean   :3.512   Mean   :1.033e+09
```

```
##  3rd Qu.:53608   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:10000054    Length:10000054
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

```r
head(movielens)
```

```
##   userId movieId rating timestamp                         title
## 1      1     122      5 838985046              Boomerang (1992)
## 2      1     185      5 838983525               Net, The (1995)
## 3      1     231      5 838983392          Dumb & Dumber (1994)
## 4      1     292      5 838983421               Outbreak (1995)
## 5      1     316      5 838983392               Stargate (1994)
## 6      1     329      5 838983392 Star Trek: Generations (1994)
##                          genres
## 1                Comedy|Romance
## 2         Action|Crime|Thriller
## 3                        Comedy
## 4  Action|Drama|Sci-Fi|Thriller
## 5       Action|Adventure|Sci-Fi
## 6 Action|Adventure|Drama|Sci-Fi
```

```r
# This has 10M ratings and has been joined with movies to build the movielens dataset
```

Summary of edx

```r
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
##
##
##
```

```r
# Summary of the edx dataset confirms no missing values

# First entries of the edx dataset
head(edx)
```

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

```r
#Data Analysis of edx
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 838984474 838983653 838984885 83
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "Stargate (1994)" ...
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thriller" "Action|Ad
```

```r
# This consists of 90% of the movielens dataset ~9M
# Each record is associated with:
#1. userId
#2. movieId
#3. rating
#4. timestamp
#5. title
#6. genres

# Checking the edx dataset properties.
n_distinct(edx$movieId)
```

```
## [1] 10677
```

```r
n_distinct(edx$genres)
```

```
## [1] 797
```

```r
n_distinct(edx$userId)
```

```
## [1] 69878
```

```r
nrow(edx)
```

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

```
#The total of unique movies and users in the edx subset is about 70,000 unique users
# and about 10,700 different movies approximately
```

Summary of the validation dataset

```
##      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
##
##
##
```

```
##   userId movieId rating timestamp
## 1      1     231      5 838983392
## 2      1     480      5 838983653
## 3      1     586      5 838984068
## 4      2     151      3 868246450
## 5      2     858      2 868245645
## 6      2    1544      3 868245920
##                                                  title
## 1                                  Dumb & Dumber (1994)
## 2                                  Jurassic Park (1993)
## 3                                     Home Alone (1990)
## 4                                        Rob Roy (1995)
## 5                                 Godfather, The (1972)
## 6 Lost World: Jurassic Park, The (Jurassic Park 2) (1997)
##                                  genres
## 1                                Comedy
## 2        Action|Adventure|Sci-Fi|Thriller
## 3                        Children|Comedy
## 4                Action|Drama|Romance|War
## 5                             Crime|Drama
## 6 Action|Adventure|Horror|Sci-Fi|Thriller
```

```
## [1] 9809
```

```
## [1] 773
```

```
## [1] 68534
```

```
## [1] 999999
```

Data Pre-processing or analysis and preparation

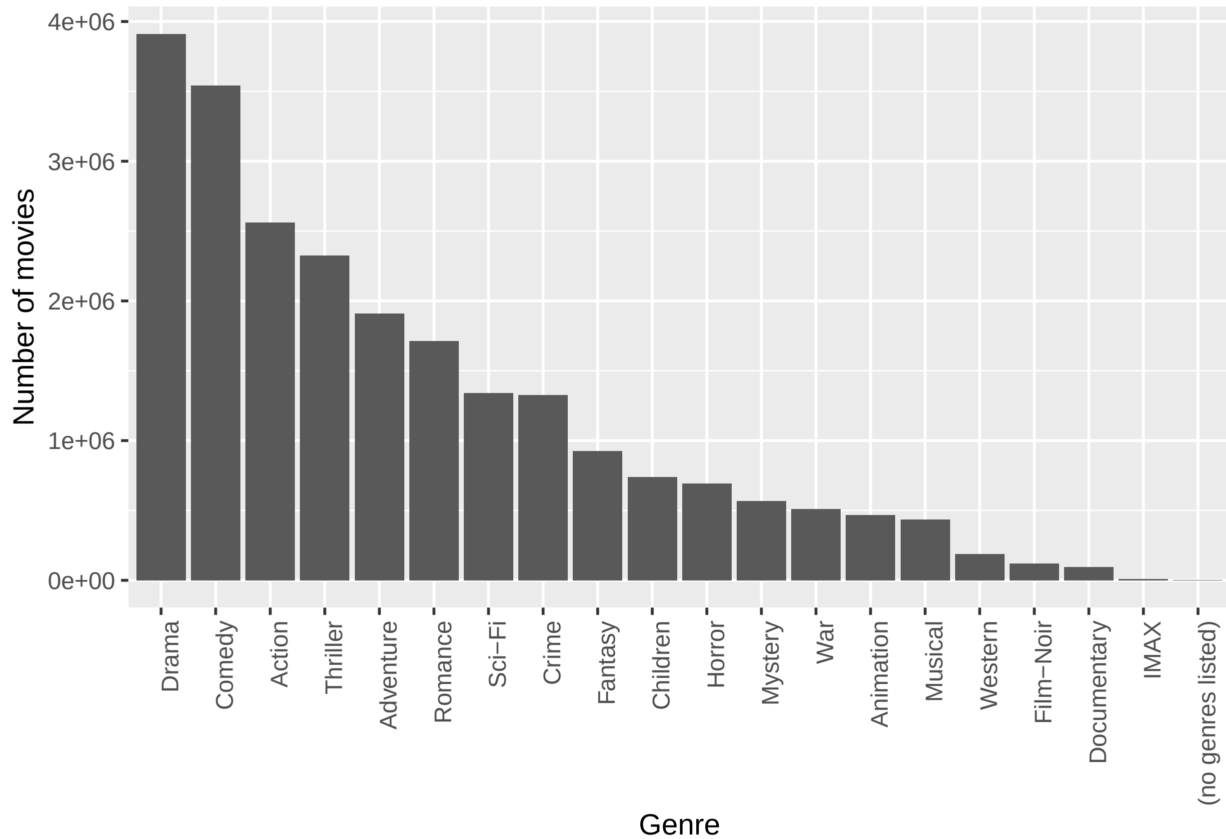
By Genres

```
##   userId movieId rating timestamp            title   genres
## 1      1     122      5 838985046 Boomerang (1992)   Comedy
## 2      1     122      5 838985046 Boomerang (1992)  Romance
## 3      1     185      5 838983525  Net, The (1995)   Action
## 4      1     185      5 838983525  Net, The (1995)    Crime
## 5      1     185      5 838983525  Net, The (1995) Thriller
## 6      1     292      5 838983421  Outbreak (1995)   Action
```

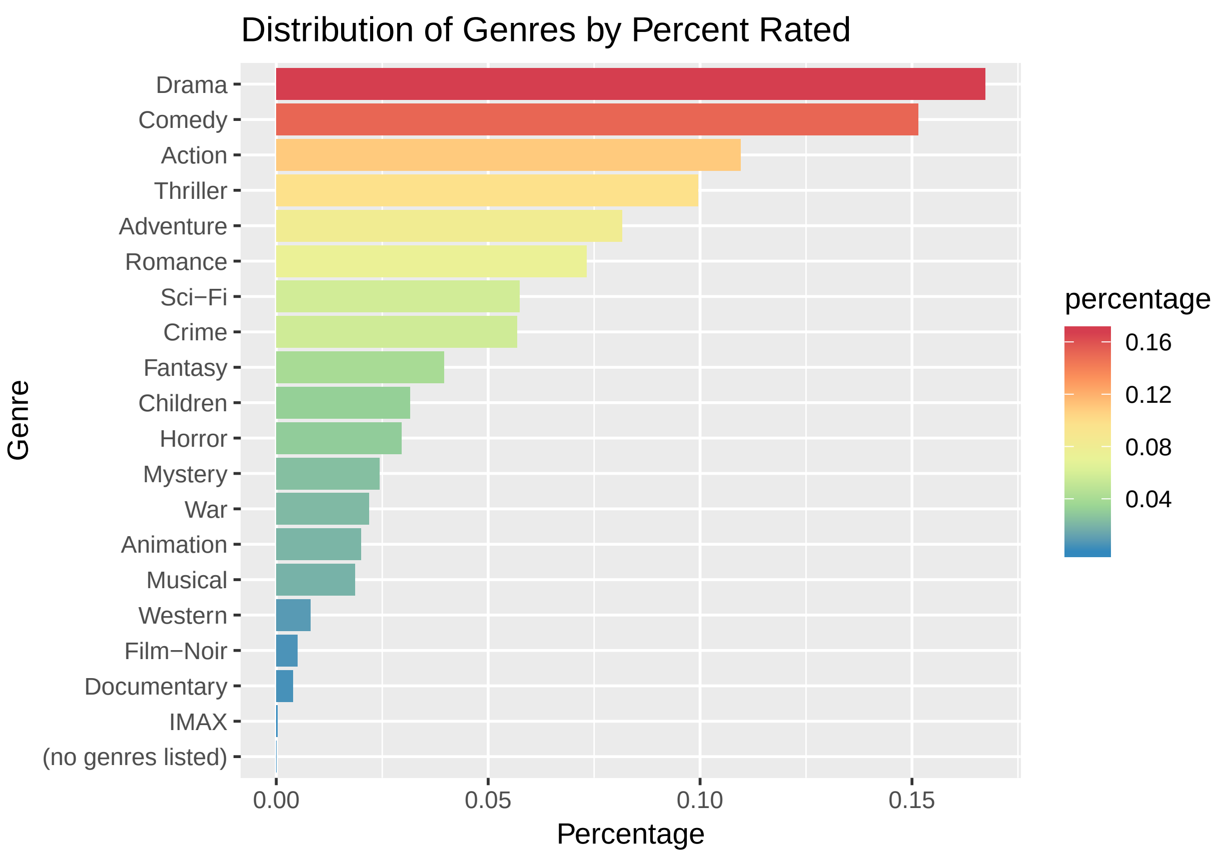
```
## # A tibble: 6 x 2
##   genres                   n
##   <chr>                <int>
## 1 (no genres listed)       7
## 2 Action             2560545
## 3 Adventure          1908892
## 4 Animation           467168
## 5 Children            737994
## 6 Comedy             3540930
```

```
##                       n
## genres               7 8181 93066 118541 189394 433080 467168 511147
##   (no genres listed) 1    0     0      0      0      0      0      0
##   Action             0    0     0      0      0      0      0      0
##   Adventure          0    0     0      0      0      0      0      0
##   Animation          0    0     0      0      0      0      1      0
##   Children           0    0     0      0      0      0      0      0
##   Comedy             0    0     0      0      0      0      0      0
##   Crime              0    0     0      0      0      0      0      0
##   Documentary        0    0     1      0      0      0      0      0
##   Drama              0    0     0      0      0      0      0      0
##   Fantasy            0    0     0      0      0      0      0      0
##   Film-Noir          0    0     0      1      0      0      0      0
##   Horror             0    0     0      0      0      0      0      0
##   IMAX               0    1     0      0      0      0      0      0
##   Musical            0    0     0      0      0      1      0      0
##   Mystery            0    0     0      0      0      0      0      0
##   Romance            0    0     0      0      0      0      0      0
##   Sci-Fi             0    0     0      0      0      0      0      0
##   Thriller           0    0     0      0      0      0      0      0
##   War                0    0     0      0      0      0      0      1
##   Western            0    0     0      0      1      0      0      0
##                       n
## genres               568332 691485 737994 925637 1327715 1341183 1712100
##   (no genres listed)      0      0      0      0       0       0       0
##   Action                  0      0      0      0       0       0       0
##   Adventure               0      0      0      0       0       0       0
##   Animation               0      0      0      0       0       0       0
##   Children                0      0      1      0       0       0       0
##   Comedy                  0      0      0      0       0       0       0
##   Crime                   0      0      0      0       1       0       0
```

```
##   Documentary                0       0       0       0        0        0        0
##   Drama                      0       0       0       0        0        0        0
##   Fantasy                    0       0       0       1        0        0        0
##   Film-Noir                  0       0       0       0        0        0        0
##   Horror                     0       1       0       0        0        0        0
##   IMAX                       0       0       0       0        0        0        0
##   Musical                    0       0       0       0        0        0        0
##   Mystery                    1       0       0       0        0        0        0
##   Romance                    0       0       0       0        0        0        1
##   Sci-Fi                     0       0       0       0        0        1        0
##   Thriller                   0       0       0       0        0        0        0
##   War                        0       0       0       0        0        0        0
##   Western                    0       0       0       0        0        0        0
##                     n
## genres               1908892 2325899 2560545 3540930 3910127
##   (no genres listed)       0       0       0       0       0
##   Action                   0       0       1       0       0
##   Adventure                1       0       0       0       0
##   Animation                0       0       0       0       0
##   Children                 0       0       0       0       0
##   Comedy                   0       0       0       1       0
##   Crime                    0       0       0       0       0
##   Documentary              0       0       0       0       0
##   Drama                    0       0       0       0       1
##   Fantasy                  0       0       0       0       0
##   Film-Noir                0       0       0       0       0
##   Horror                   0       0       0       0       0
##   IMAX                     0       0       0       0       0
##   Musical                  0       0       0       0       0
##   Mystery                  0       0       0       0       0
##   Romance                  0       0       0       0       0
##   Sci-Fi                   0       0       0       0       0
##   Thriller                 0       1       0       0       0
##   War                      0       0       0       0       0
##   Western                  0       0       0       0       0
```



4e+06
3e+06
2e+06
1e+06
0e+00
Number of movies
Drama
Comedy
Action
Thriller
Adventure
Romance
Sci−Fi
Crime
Fantasy
Children
Horror
Mystery
War
Animation
Musical
Western
Film−Noir
Documentary
IMAX
(no genres listed)
Genre



Distribution of Genres by Percent Rated
Drama
Comedy
Action
Thriller
Adventure
Romance
Sci−Fi
Crime
Fantasy
Children
Horror
Mystery
War
Animation
Musical
Western
Film−Noir
Documentary
IMAX
(no genres listed)
Genre
0.00
0.05
0.10
0.15
Percentage
percentage
0.16
0.12
0.08
0.04

Check ratings

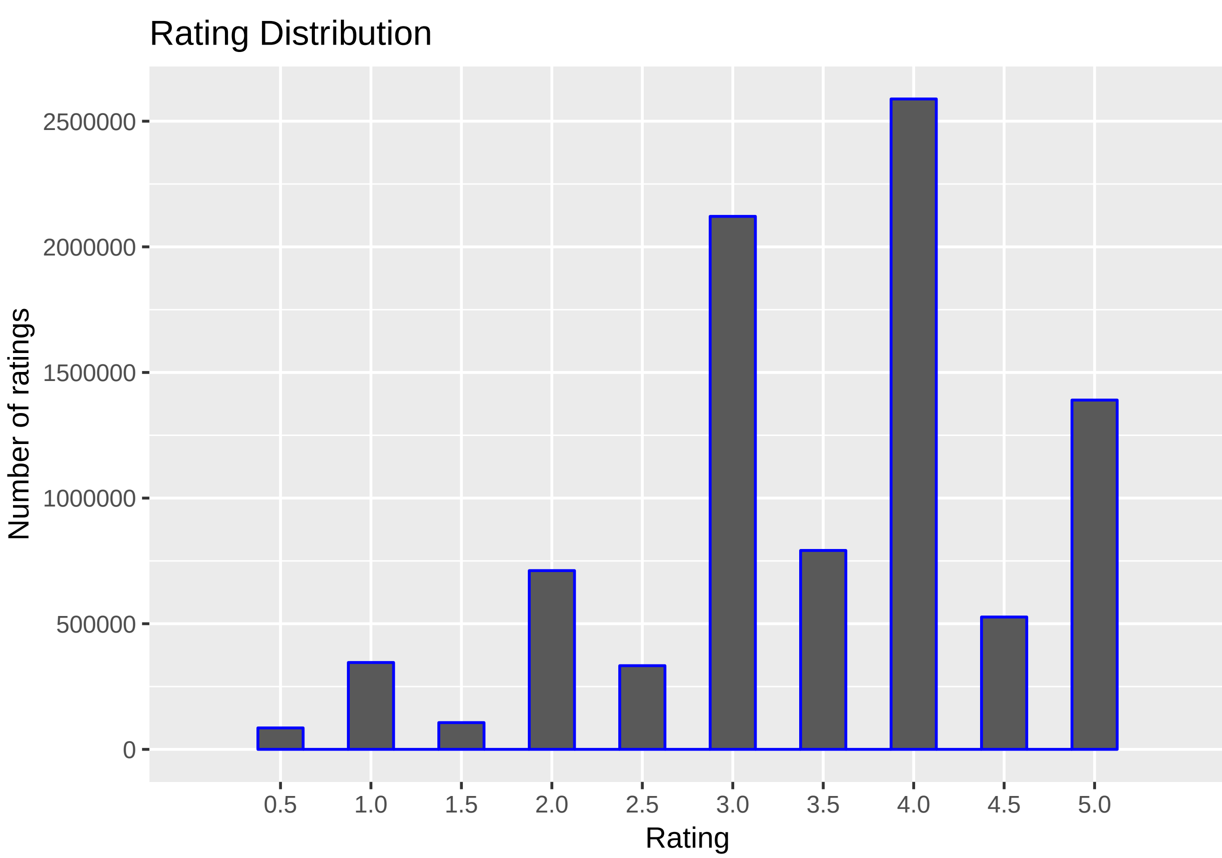
```r
# Check ratings
table(edx$rating)
```

```
##
##     0.5       1     1.5       2     2.5       3     3.5       4     4.5
##   85374  345679  106426  711422  333010 2121240  791624 2588430  526736
##       5
## 1390114
```

```r
summary(edx$rating)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.500   3.000   4.000   3.512   4.000   5.000
```

```r
# Ratings range from 0.5 to 5.0.
# The difference in median and mean shows that the distribution is skewed towards higher ratings.

edx %>%
  ggplot(aes(rating)) +
  geom_histogram(binwidth = 0.25, color = "blue") +
  scale_x_discrete(limits = c(seq(0.5,5,0.5))) +
  scale_y_continuous(breaks = c(seq(0, 3000000, 500000))) +
  xlab("Rating") +
  ylab("Number of ratings") +
  ggtitle("Rating Distribution")
```



The chart shows that whole-number ratings are more common that 0.5 ratings. This also shows that users have a preference to rate movies rather higher than lower as shown by the distribution of ratings 4 is the most common rating, followed by 3 and 5. 0.5 is the least common rating.

To make further analysis on the rating distribution, we will validate and update the edx dataset

```
##   userId movieId rating timestamp                         title
## 1      1     122      5 838985046              Boomerang (1992)
## 2      1     185      5 838983525               Net, The (1995)
## 3      1     292      5 838983421               Outbreak (1995)
## 4      1     316      5 838983392               Stargate (1994)
## 5      1     329      5 838983392 Star Trek: Generations (1994)
## 6      1     355      5 838984474       Flintstones, The (1994)
##                           genres year
## 1                 Comedy|Romance 1996
## 2          Action|Crime|Thriller 1996
## 3  Action|Drama|Sci-Fi|Thriller 1996
## 4        Action|Adventure|Sci-Fi 1996
## 5 Action|Adventure|Drama|Sci-Fi 1996
## 6       Children|Comedy|Fantasy 1996
```

```
## [1] "userId"    "movieId"   "rating"    "timestamp" "title"     "genres"
## [7] "year"
```

```
##   userId movieId rating timestamp                         title
## 1      1     122      5 838985046              Boomerang (1992)
## 2      1     185      5 838983525               Net, The (1995)
## 3      1     292      5 838983421               Outbreak (1995)
## 4      1     316      5 838983392               Stargate (1994)
## 5      1     329      5 838983392 Star Trek: Generations (1994)
## 6      1     355      5 838984474       Flintstones, The (1994)
##                           genres year premiered
## 1                 Comedy|Romance 1992      1992
## 2          Action|Crime|Thriller 1995      1995
## 3  Action|Drama|Sci-Fi|Thriller 1995      1995
## 4        Action|Adventure|Sci-Fi 1994      1994
## 5 Action|Adventure|Drama|Sci-Fi 1994      1994
## 6       Children|Comedy|Fantasy 1994      1994
```

```
## # A tibble: 6 x 4
## # Groups:   movieId, title [?]
##   movieId title                                         premiered     n
##     <dbl> <chr>                                             <dbl> <int>
## 1     671 Mystery Science Theater 3000: The Movie (1996)      3000  3280
## 2    2308 Detroit 9000 (1973)                                 9000    22
## 3    4159 3000 Miles to Graceland (2001)                      3000   714
## 4    5310 Transylvania 6-5000 (1985)                          5000   195
## 5    8864 Mr. 3000 (2004)                                     3000   146
## 6   27266 2046 (2004)                                         2046   426
```

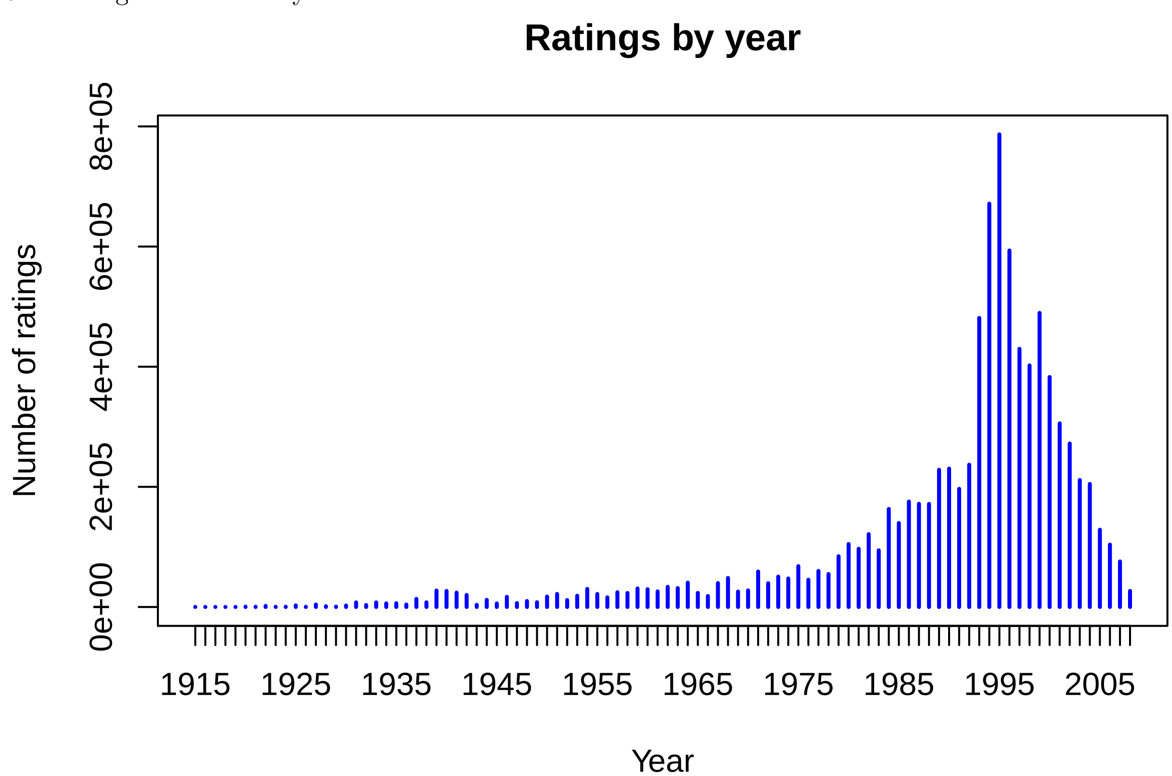
```
## # A tibble: 8 x 4
## # Groups:   movieId, title [?]
##   movieId title                                           premiered     n
```

```
##     <dbl> <chr>                                                 <dbl> <int>
## 1    1422 Murder at 1600 (1997)                                  1600  1566
## 2    4311 Bloody Angels (1732 Høtten: Marerittet Har et P~      1732     9
## 3    5472 1776 (1972)                                            1776   185
## 4    6290 House of 1000 Corpses (2003)                           1000   367
## 5    6645 THX 1138 (1971)                                        1138   464
## 6    8198 1000 Eyes of Dr. Mabuse, The (Tausend Augen des~      1000    24
## 7    8905 1492: Conquest of Paradise (1992)                      1492   134
## 8   53953 1408 (2007)                                            1408   466
```

```
##   userId movieId rating timestamp                          title
## 1      1     122      5 838985046               Boomerang (1992)
## 2      1     185      5 838983525                Net, The (1995)
## 3      1     292      5 838983421                Outbreak (1995)
## 4      1     316      5 838983392                Stargate (1994)
## 5      1     329      5 838983392 Star Trek: Generations (1994)
## 6      1     355      5 838984474        Flintstones, The (1994)
##                          genres year premiered age_of_movie
## 1                Comedy|Romance 1992      1992           27
## 2         Action|Crime|Thriller 1995      1995           24
## 3  Action|Drama|Sci-Fi|Thriller 1995      1995           24
## 4       Action|Adventure|Sci-Fi 1994      1994           25
## 5 Action|Adventure|Drama|Sci-Fi 1994      1994           25
## 6       Children|Comedy|Fantasy 1994      1994           25
```

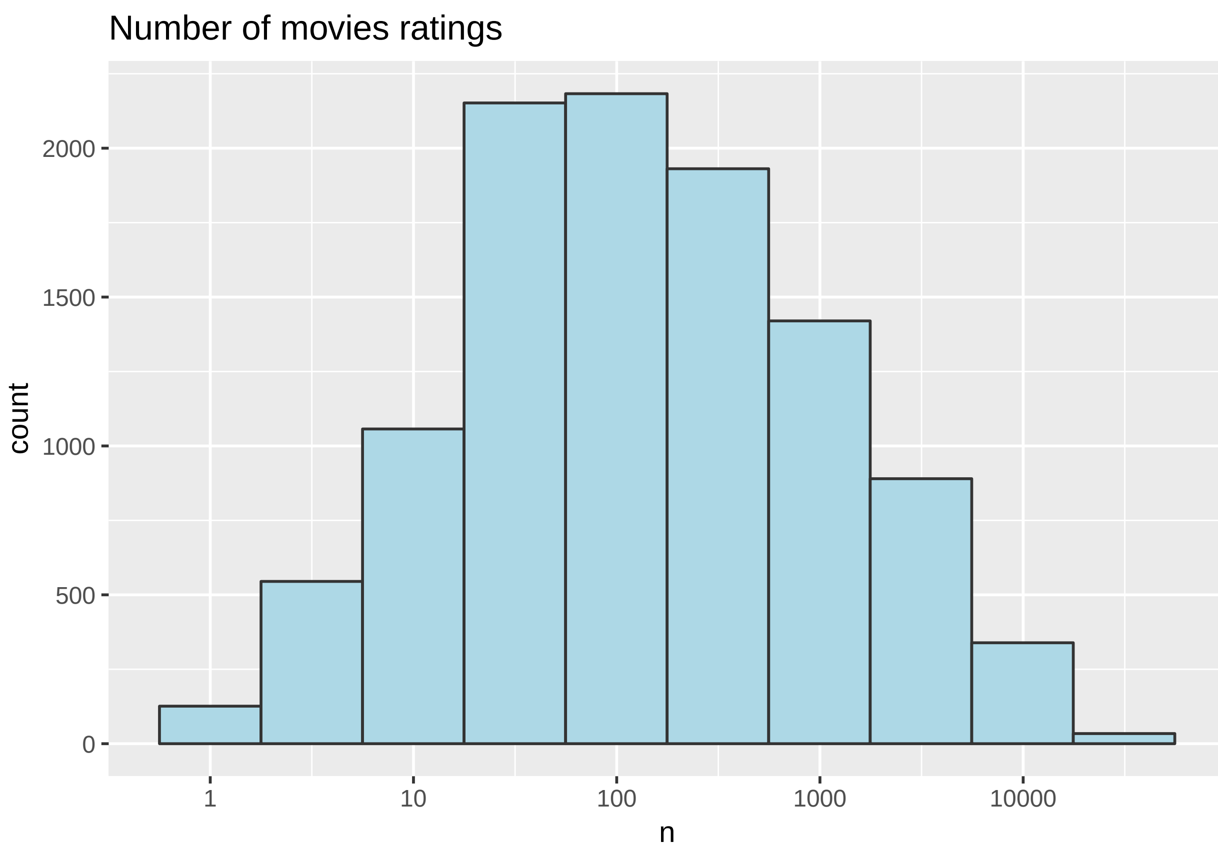
If required, the same steps can be applied to the validation dataset

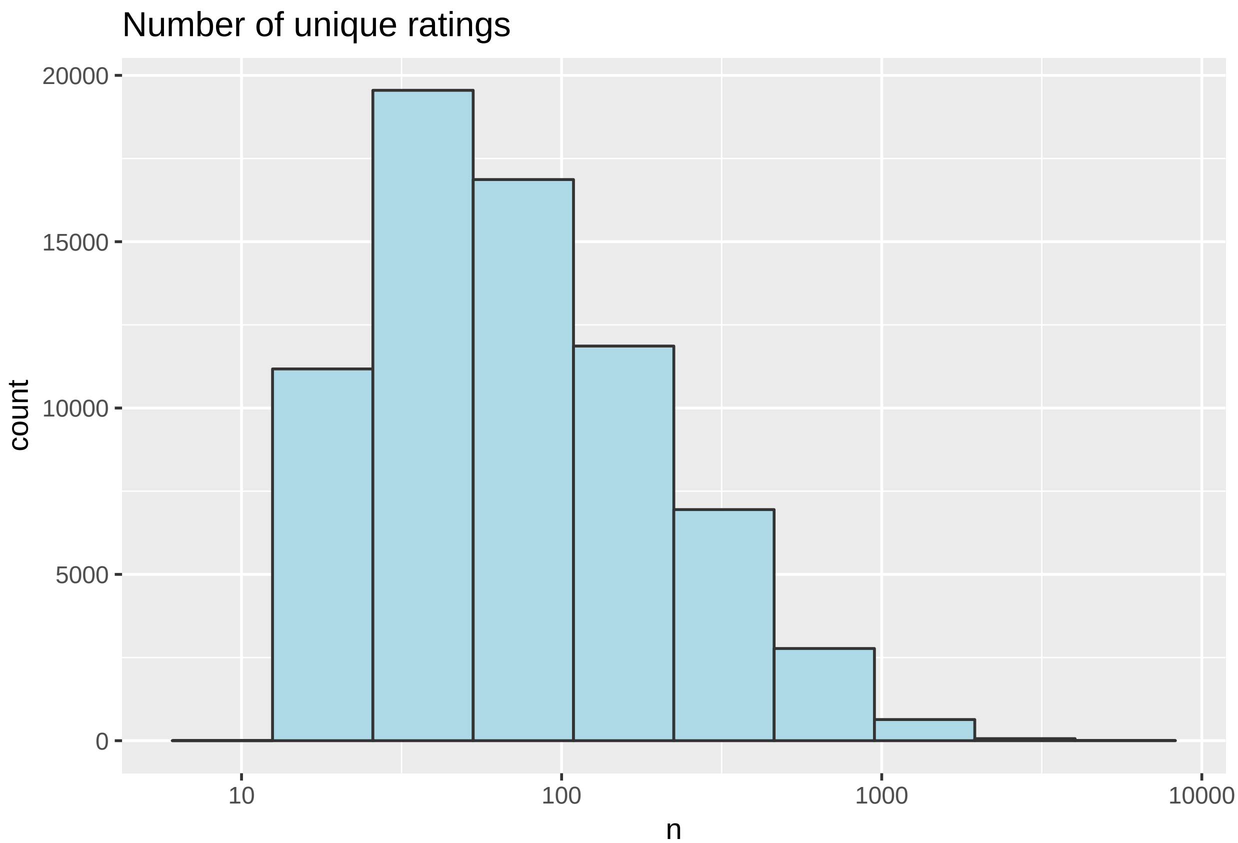
Continuing with the analysis

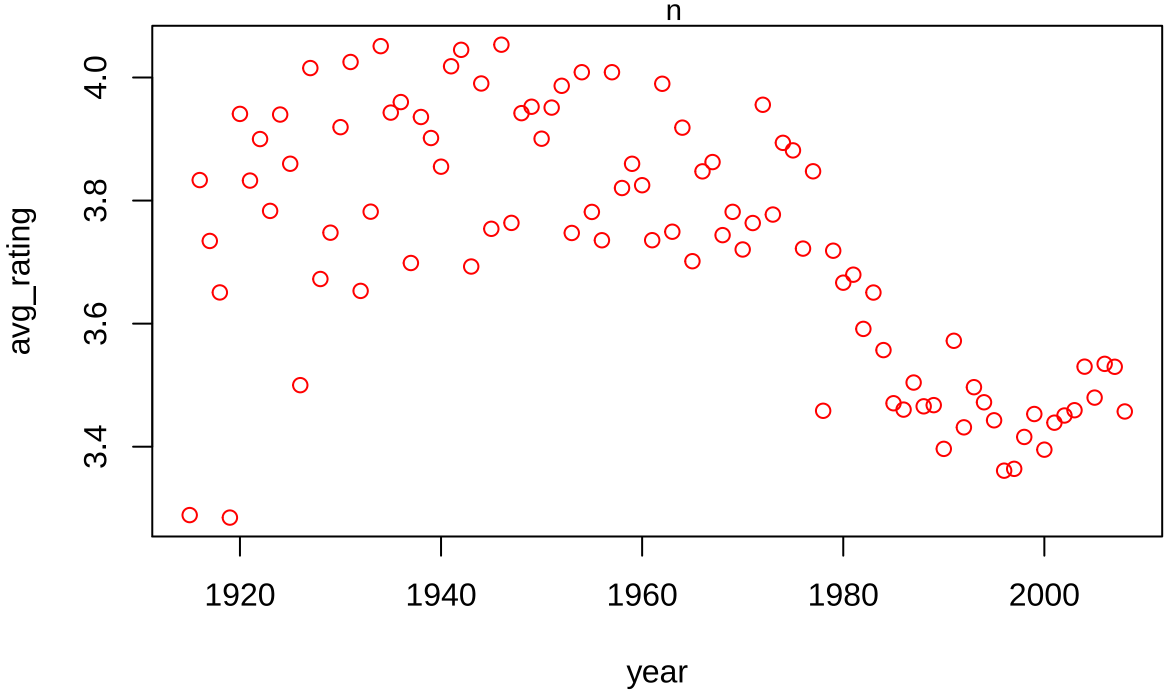


We observe that more recent movies get more user ratings. Movies earlier than 1950 get few ratings, whereas newer movies, especially in the 90s get far more ratings.

```
## # A tibble: 6 x 9
## # Groups:   movieId [6]
##   userId movieId rating timestamp title genres  year movies_by_user
##    <int>   <dbl>  <dbl>     <int> <chr> <chr>  <dbl>          <int>
## 1      1     122      5 838985046 Boom~ Comed~  1992             19
## 2      1     185      5 838983525 Net,~ Actio~  1995             19
## 3      1     292      5 838983421 Outb~ Actio~  1995             19
## 4      1     316      5 838983392 Star~ Actio~  1994             19
## 5      1     329      5 838983392 Star~ Actio~  1994             19
## 6      1     355      5 838984474 Flin~ Child~  1994             19
## # ... with 1 more variable: users_by_movie <int>
```







We see that the older the movies, the more widely distributed are their ratings, which can be explained by the lower frequency of movie ratings

```
# To check what affects the movie ratings are affected by

# Movie rating averages
movie_avg <- edx_upd %>% group_by(movieId) %>% summarize(avg_movie_rating = mean(rating))
```

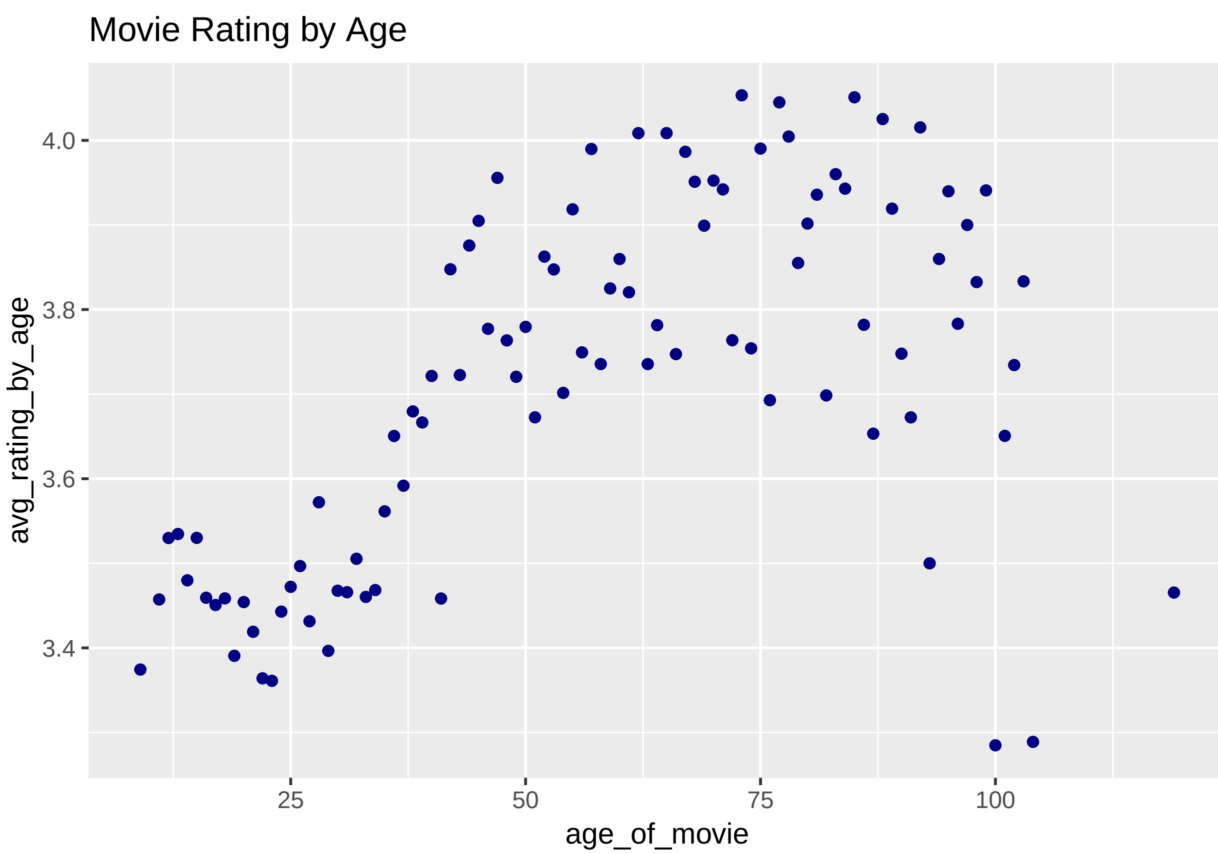
```r
user_avg <- edx_upd %>% group_by(userId) %>% summarize(avg_user_rating = mean(rating))
year_avg <- edx_upd%>% group_by(year) %>% summarize(avg_rating_by_year = mean(rating)) #year the movie 
age_avg <- edx_upd %>% group_by(age_of_movie) %>% summarize(avg_rating_by_age = mean(rating)) #age of m

# View sample data and plot
head(age_avg)
```

```
## # A tibble: 6 x 2
##   age_of_movie avg_rating_by_age
##          <dbl>             <dbl>
## 1            9              3.37
## 2           11              3.46
## 3           12              3.53
## 4           13              3.53
## 5           14              3.48
## 6           15              3.53
```

```r
head(user_avg)
```

```
## # A tibble: 6 x 2
##   userId avg_user_rating
##    <int>           <dbl>
## 1      1            5
## 2      2            3.29
## 3      3            3.94
## 4      4            4.06
## 5      5            3.92
## 6      6            3.95
```

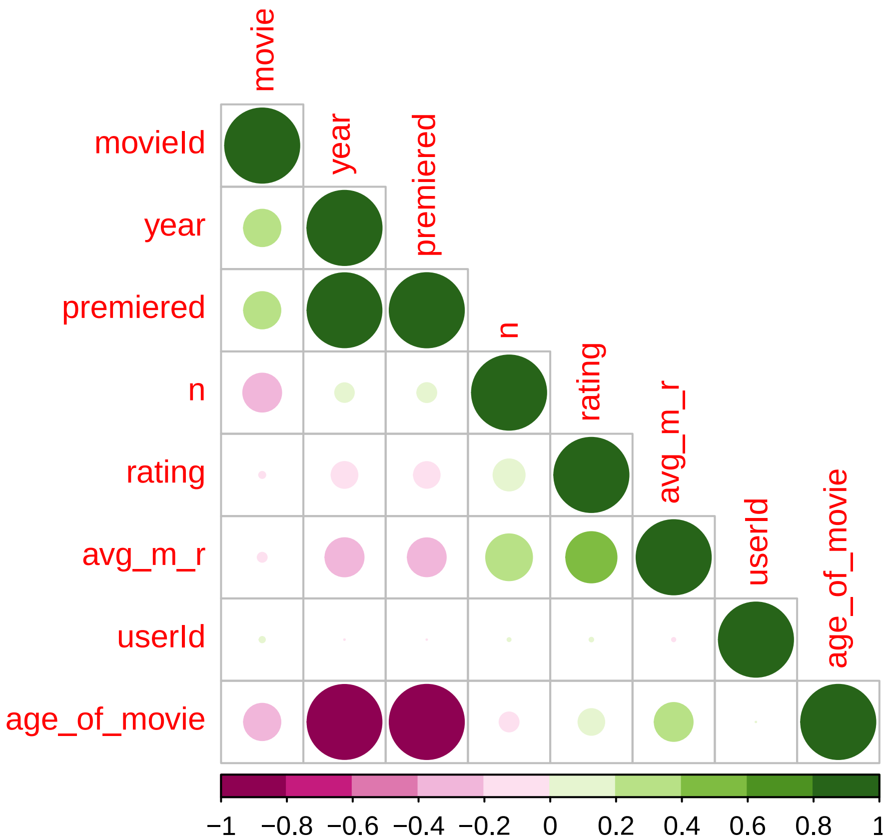
```r
age_avg %>% ggplot(aes(age_of_movie, avg_rating_by_age)) +
  geom_point(color = "navy") +  ggtitle("Movie Rating by Age")
```



This follows our earlier observation and also shows higher ratings for older movies up to about ~80 years after which the ratings decline.

To further understand what affects the ratings, we will try to correlate all the available parameters

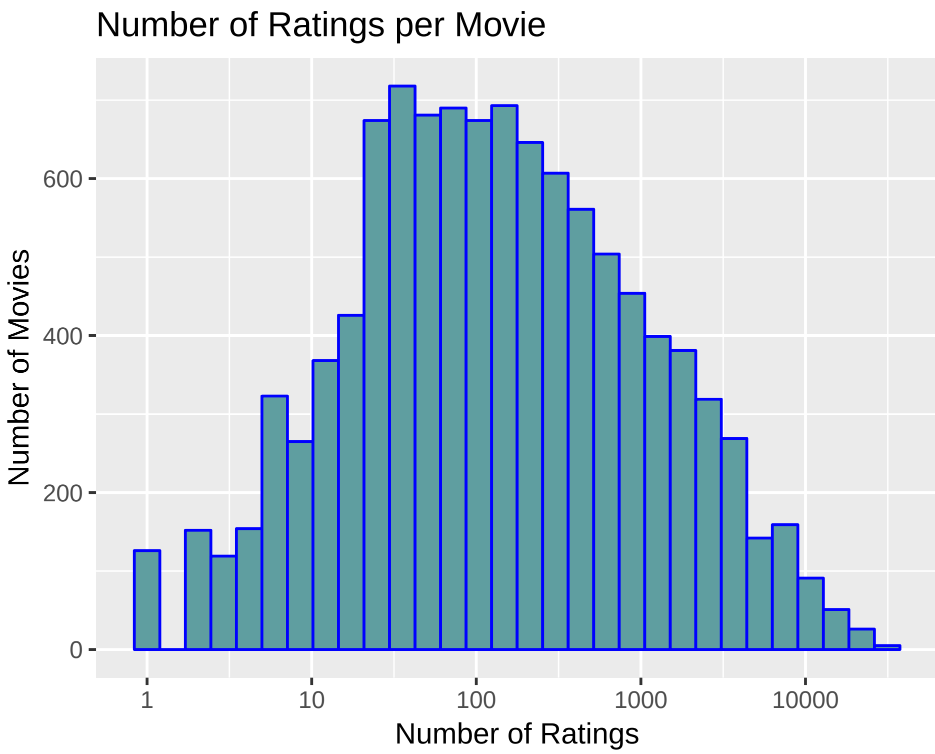
```
##   rating movieId userId year age_of_movie premiered     n  avg_m_r
## 1      5     122      1 1992           27      1992  2178 2.858586
## 2      5     185      1 1995           24      1995 13469 3.129334
## 3      5     292      1 1995           24      1995 14447 3.418011
## 4      5     316      1 1994           25      1994 17030 3.349677
## 5      5     329      1 1994           25      1994 14550 3.337457
## 6      5     355      1 1994           25      1994  4831 2.487787
```



We notice that some movies have been rated much often than others, while some have very few ratings and sometimes even only one rating. This will be important for our model as very low rating numbers might results in untrustworthy estimate for our predictions. Almost 125 movies have been rated only once.

We also notice that some movies have been rated a lot more often than others, while some have very few ratings and sometimes even one rating

```
edx %>%
count(movieId) %>%
ggplot(aes(n)) +
geom_histogram(bins = 30, color = "blue", fill = "cadetblue") +
scale_x_log10() +
xlab("Number of Ratings") +
  ylab("Number of Movies") +
ggtitle("Number of Ratings per Movie")
```



We notice that these movies have only one rating, making future rating prediction difficult.

```
edx %>%
  group_by(movieId) %>%
  summarize(count = n()) %>%
  filter(count == 1) %>%
  left_join(edx, by = "movieId") %>%
  group_by(title) %>%
  summarize(rating = rating, n_rating = count) %>%
  slice(1:20) %>%
  knitr::kable()
```

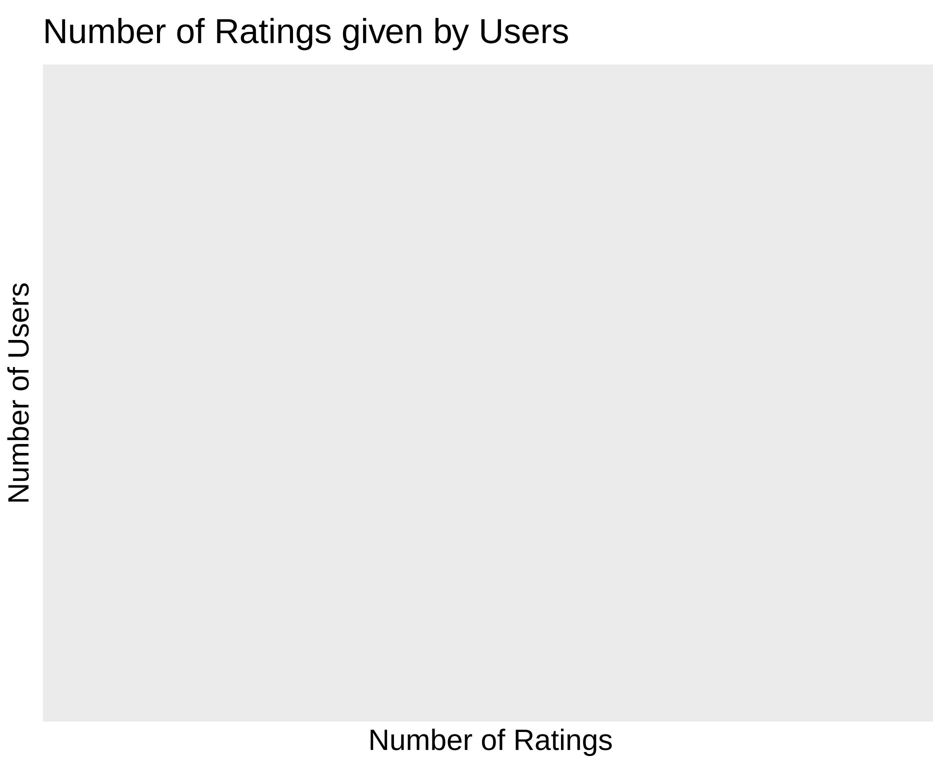
title	rating	n_rating
1, 2, 3, Sun (Un, deuz, trois, soleil) (1993)	2.0	1
100 Feet (2008)	2.0	1
4 (2005)	2.5	1
Accused (Anklaget) (2005)	0.5	1
Ace of Hearts (2008)	2.0	1
Ace of Hearts, The (1921)	3.5	1
Adios, Sabata (Indio Black, sai che ti dico: Sei un gran figlio di…) (1971)	1.5	1
Africa addio (1966)	3.0	1
Aleksandra (2007)	3.0	1
Bad Blood (Mauvais sang) (1986)	4.5	1
Battle of Russia, The (Why We Fight, 5) (1943)	3.5	1
Bellissima (1951)	4.0	1
Big Fella (1937)	3.0	1
Black Tights (1-2-3-4 ou Les Collants noirs) (1960)	3.0	1
Blind Shaft (Mang jing) (2003)	2.5	1
Blue Light, The (Das Blaue Licht) (1932)	5.0	1

title	rating	n_rating
Borderline (1950)	3.0	1
Brothers of the Head (2005)	2.5	1
Chapayev (1934)	1.5	1
Cold Sweat (De la part des copains) (1970)	2.5	1

One can observe that the majority of users have rated between 30 and 100 movies. So, a user penalty term needs to be included later in our models.

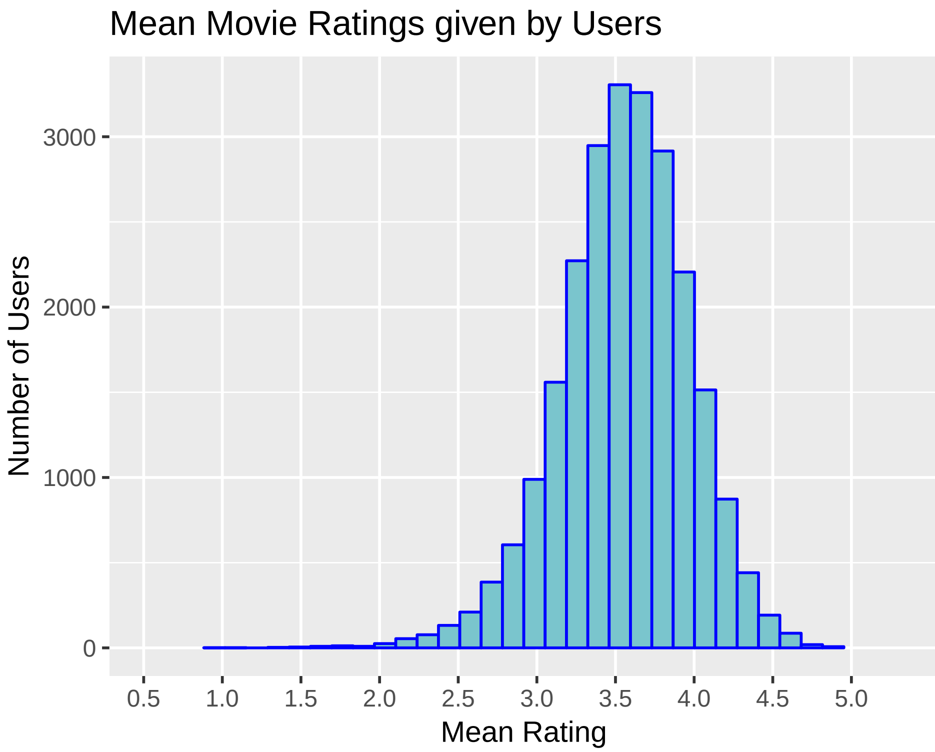
```
# Plot number of ratings given by users
edx %>%
count(userId) %>%
ggplot(aes(n)) +
geom_histogram(bins = 30, color = "blue", fill = "cadetblue") +
scale_x_log10() +
xlab("Number of Ratings") +
ylab("Number of Users") +
ggtitle("Number of Ratings given by Users")
```

```
## Warning: Computation failed in `stat_bin()`:
## `binwidth` must be positive
```



Furthermore, users differ vastly in how critical they are with their ratings. Some users tend to give much lower star ratings and some users tend to give higher star ratings than average. The visualization below includes only users that have rated at least 100 movies.

```r
edx %>%
  group_by(userId) %>%
  filter(n() >= 100) %>%
  summarize(b_u = mean(rating)) %>%
  ggplot(aes(b_u)) +
  geom_histogram(bins = 30, color = "blue",fill="cadetblue3") +
  xlab("Mean Rating") +
  ylab("Number of Users") +
  ggtitle("Mean Movie Ratings given by Users") +
  scale_x_discrete(limits = c(seq(0.5,5,0.5)))
```



Modelling Approach

The loss-function, that computes the RMSE is defined as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i}(\hat{y}_{u,i} - y_{u,i})^2}$$

with N being the number of user/movie combinations and the sum occurring over all these combinations. The RMSE is our measure of model accuracy.

The written function to compute the RMSE for vectors of ratings and their corresponding predictions is:

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

1. Average Movie Rating Model

The first model predicts the same rating for all movies, so we compute the dataset's mean rating. The expected rating of the underlying data set is between 3 and 4. We start by building the simplest possible recommender system by predicting the same rating for all movies regardless of user who give it. A model based approach assumes the same rating for all movie with all differences explained by random variation :

$$Y_{u,i} = \mu + \epsilon_{u,i}$$

with $\epsilon_{u,i}$ independent error sample from the same distribution centered at 0 and μ the "true" rating for all movies. This very simple model makes the assumption that all differences in movie ratings are explained by random variation alone. The estimate that minimizes the RMSE is the least square estimate of $Y_{u,i}$, in this case, is the average of all ratings: The expected rating of the underlying data set is between 3 and 4.

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

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

If we predict all unknown ratings with μ or mu, we obtain the first RMSE:

```
model_1_rmse <- RMSE(validation$rating, mu)
model_1_rmse
```

```
## [1] 1.061202
```

Here, we represent results table with the first RMSE:

```
rmse_results <- data_frame(method = "Average Movie Rating Model",
                           RMSE = model_1_rmse)
rmse_results %>% knitr::kable()
```

method	RMSE
Average Movie Rating Model	1.061202

This give us our baseline RMSE to compare with next modelling approaches.

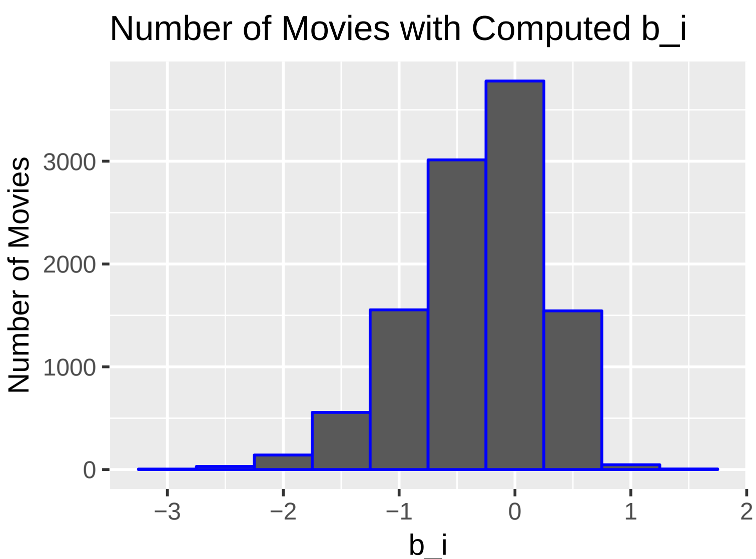
In order to do better than simply predicting the average rating, one can incorporate some of insights gained during the exploratory data analysis.

2. Movie Effect Model

To improve above model one can focus on the fact that some movies are just generally rated higher than others. Higher ratings are mostly linked to popular movies among users and the opposite is true for unpopular movies. We compute the estimated deviation of each movies' mean rating from the total mean of all movies μ. The resulting variable is called "b" (as bias) for each movie "i" b_i, that represents average ranking for movie i:

$$Y_{u,i} = \mu + b_i + \epsilon_{u,i}$$

```r
movie_avgs <- edx %>%
  group_by(movieId) %>%
  summarize(b_i = mean(rating - mu))
movie_avgs %>% qplot(b_i, geom ="histogram", bins = 10,
                     data = ., color = I("blue"),
                     ylab = "Number of Movies",
                     main = "Number of Movies with Computed b_i")
```



The histogram is left skewed, implying that more movies have negative effects This is called the penalty term movie effect.

Our prediction can improve once prediction is done using this model.

```r
predicted_ratings <- mu +  validation %>%
  left_join(movie_avgs, by='movieId') %>%
  pull(b_i)
model_2_rmse <- RMSE(predicted_ratings, validation$rating)
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Movie Effect Model",
                                     RMSE = model_2_rmse ))
rmse_results %>% knitr::kable()
```

method	RMSE
Average Movie Rating Model	1.0612018
Movie Effect Model	0.9439087

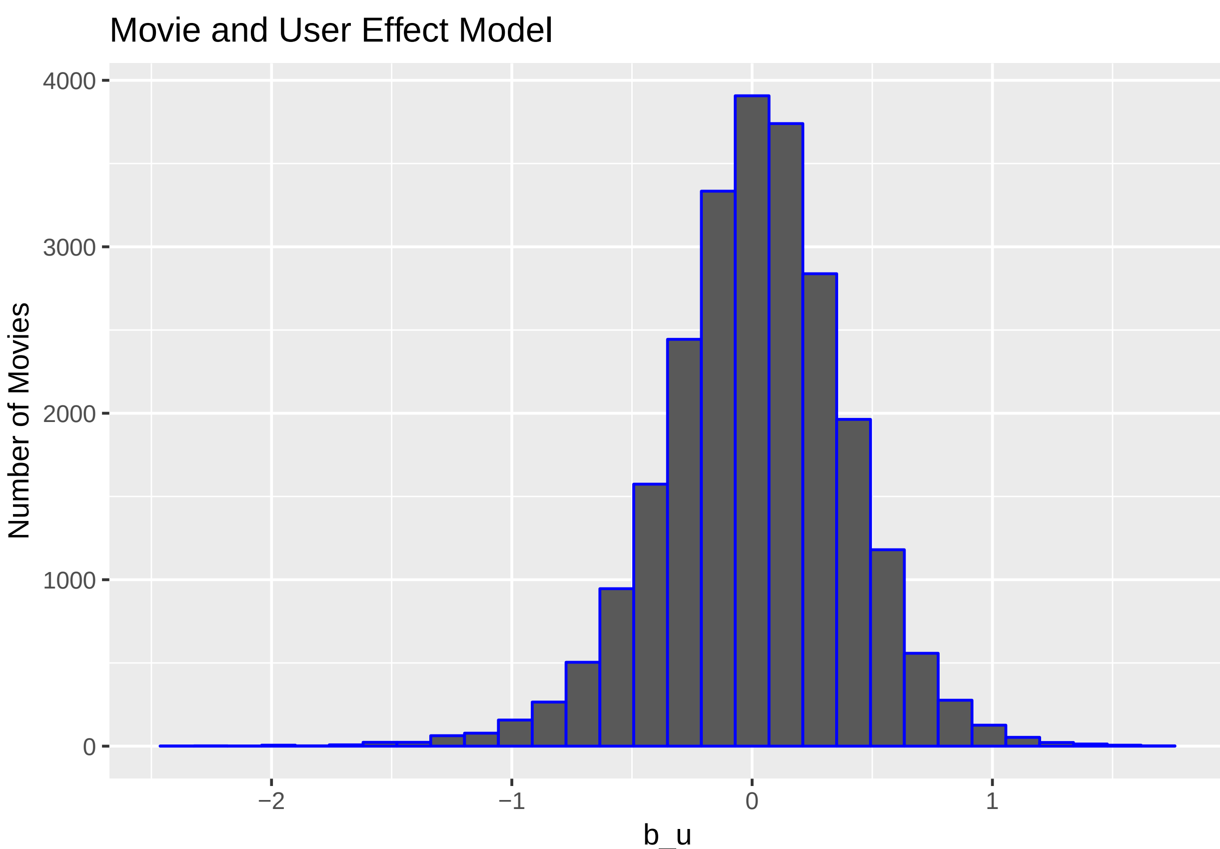
So we have predicted movie rating based on the fact that movies are rated differently by adding the computed b_i to μ. If an individual movie is on average rated worse that the average rating of all movies μ , we predict that it will rated lower that μ by b_i, the difference of the individual movie average from the total average.

We can see an improvement but this model does not consider the individual user rating effect.

3. Movie and User Effect Model

The average rating for user μ, for those that have rated over 100 movies, said penalty term user effect. In fact users affect the ratings positively or negatively.

```
user_avgs<- edx %>%
  left_join(movie_avgs, by='movieId') %>%
  group_by(userId) %>%
  filter(n() >= 100) %>%
  summarize(b_u = mean(rating - mu - b_i))
user_avgs%>% qplot(b_u, geom ="histogram",
                   bins = 30, data = ., color = I("blue"),
                   ylab = "Number of Movies",
                   main = "Movie and User Effect Model")
```



There is substantial variability across users as well: some users are very cranky and others love every movie. This implies that further improvement to the model may be:

$$Y_{u,i} = \mu + b_i + b_u + \epsilon_{u,i}$$

where b_u is a user-specific effect. If a cranky user (negative b_u rates a great movie (positive b_i), the effects counter each other and we may be able to correctly predict that this user gave this great movie a 3 rather than a 5.

An approximation can be computed by μ and b_i, and estimating b_u, as the average of

$$Y_{u,i} - \mu - b_i$$

```
user_avgs <- edx %>%
  left_join(movie_avgs, by='movieId') %>%
  group_by(userId) %>%
  summarize(b_u = mean(rating - mu - b_i))
```

Construct predictors can improve RMSE.

```
predicted_ratings <- validation%>%
  left_join(movie_avgs, by='movieId') %>%
  left_join(user_avgs, by='userId') %>%
  mutate(pred = mu + b_i + b_u) %>%
  pull(pred)

model_3_rmse <- RMSE(predicted_ratings, validation$rating)
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Movie and User Effect Model",
                                     RMSE = model_3_rmse))
rmse_results %>% knitr::kable()
```

method	RMSE
Average Movie Rating Model	1.0612018
Movie Effect Model	0.9439087
Movie and User Effect Model	0.8653488

Our rating predictions further reduced the RMSE, But still, mistakes were made on our first model (using only movies). The supposed "best" and "worst" movies were rated by few users, in most cases just one user. These movies were mostly obscure ones. This is because with a few users, more uncertainty is created. Therefore larger estimates of b_i, negative or positive, are more likely.

Until now, the computed standard error and constructed confidence intervals account for different levels of uncertainty. The concept of regularization permits to penalize large estimates that come from small sample sizes. The general idea is to add a penalty for large values of b_i to the sum of squares equation that we minimize. So having many large b_i, make it harder to minimize. Regularization is a method used to reduce the effect of overfitting.

4. Regularized Movie and User Effect Model

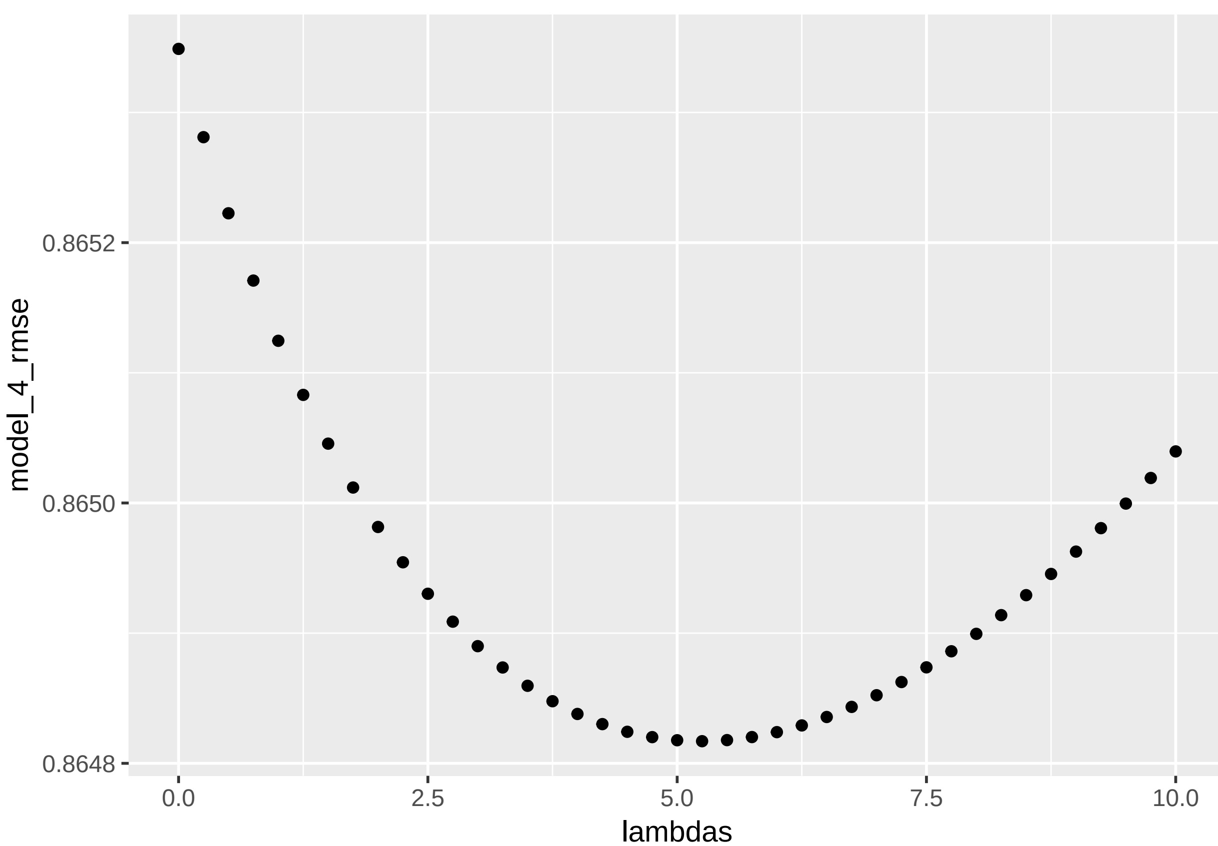
So estimates of b_i and b_u are caused by movies with very few ratings and in some users that only rated a very small number of movies. Hence this can strongly influence the prediction. The use of the regularization permits to penalize these aspects. We should find the value of lambda (that is a tuning parameter) that will minimize the RMSE. This shrinks the b_i and b_u in case of small number of ratings.

```
lambdas <- seq(0, 10, 0.25)

model_4_rmse <- 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 = sum(rating - b_i - mu)/(n()+l))

  predicted_ratings <-
    validation %>%
    left_join(b_i, by = "movieId") %>%
    left_join(b_u, by = "userId") %>%
    mutate(pred = mu + b_i + b_u) %>%
    pull(pred)

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

We plot RMSE vs Lambdas to select the optimal lambda

```
qplot(lambdas, model_4_rmse)
```



For the full model, the optimal lambda is:

```
  lambda <- lambdas[which.min(model_4_rmse)]
lambda
```

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

For the full model, the optimal lambda is: 5.25

The new results will be:

```r
rmse_results <- bind_rows(rmse_results,
                          data_frame(
                            method="Regularized Movie and User Effect Model",
                                     RMSE = min(model_4_rmse)))
rmse_results %>% knitr::kable()
```

method	RMSE
Average Movie Rating Model	1.0612018
Movie Effect Model	0.9439087
Movie and User Effect Model	0.8653488
Regularized Movie and User Effect Model	0.8648170

Results

The RMSE values of all the represented models are the following:

```r
rmse_results %>% knitr::kable()
```

method	RMSE
Average Movie Rating Model	1.0612018
Movie Effect Model	0.9439087
Movie and User Effect Model	0.8653488
Regularized Movie and User Effect Model	0.8648170

The lowest identified value of RMSE is 0.8648170.

Discussion

It can be confirmed that the final model for the project is the following:

$$Y_{u,i} = \mu + b_i + b_u + \epsilon_{u,i}$$

This model will work well if the average user doesn't rate a particularly good/popular movie with a large positive b_i, by disliking a particular movie.

Conclusion

A machine learning model has been successfully built to predict movie ratings with MovieLens dataset. The optimal model (Regularized Model) characterised by the lowest RMSE value (0.8648170) is thus the optimal selection. This is lower than the initial evaluation criterion (0.8775) given by the goal of the present project.