Movielens Project

George Moisescu

Introduction

This is my final project within the Data Science program and I try to make a movies recommendation algorithm which will predict the movie rating based on the userId, movieId, year of rating and movie genre. I use the movilens dataset provided by GroupLens research lab.

```
head(edx)
```

```
##    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
## 1:                Comedy|Romance
## 2:          Action|Crime|Thriller
## 3:  Action|Drama|Sci-Fi|Thriller
## 4:       Action|Adventure|Sci-Fi
## 5: Action|Adventure|Drama|Sci-Fi
## 6:       Children|Comedy|Fantasy
```

Exploratory Data Analysis

As we can see the data is not in tidy format, there are timestamp variable which contain the number of seconds since 1970-01-01 and the genres variable which contain several genre in the same observation.

```
max(str_count(edx$genres, "\\|"))
```

```
## [1] 7
```

```
edx_tidy <- edx %>% separate(genres,c("1_genre","2_genre","3_genre","4_genre",
                                      "5_genre","6_genre","7_genre","8_genre"),"\\|")

individual_genre <- unique(c(unique(edx_tidy$`1_genre`),unique(edx_tidy$`2_genre`),
                             unique(edx_tidy$`3_genre`),unique(edx_tidy$`4_genre`),
                             unique(edx_tidy$`5_genre`),unique(edx_tidy$`6_genre`),
                             unique(edx_tidy$`7_genre`),unique(edx_tidy$`8_genre`)))
individual_genre <-individual_genre[order(individual_genre)]
```

```
individual_genre
```

```
##  [1] "(no genres listed)" "Action"             "Adventure"
##  [4] "Animation"          "Children"           "Comedy"
##  [7] "Crime"              "Documentary"        "Drama"
## [10] "Fantasy"            "Film-Noir"          "Horror"
## [13] "IMAX"               "Musical"            "Mystery"
## [16] "Romance"            "Sci-Fi"             "Thriller"
## [19] "War"                "Western"            NA
```

We have 20 individual genres. I create 20 new columns and I populate them width 1 if the movie belong to the that genre or 0 if it doesn't.I extract also the year in which the rate was done from the timestamp column.

```
edx_tidy <- edx_tidy %>% mutate(Action = 0,Adventure = 0,Animation = 0,Children = 0,
                                Comedy = 0,Crime = 0,Documentary = 0,Drama = 0,
                                Fantasy = 0,Film_Noir = 0,Horror = 0,IMAX = 0,
                                Musical = 0,Mystery = 0,Romance = 0,Sci_Fi = 0,
                                Thriller = 0,War = 0, Western = 0,No_Genre = 0)

temp <- edx_tidy$`1_genre` == "Action"
edx_tidy$Action[temp] <- 1

temp <- edx_tidy$`1_genre` == "Adventure"
edx_tidy$Adventure[temp] <- 1
```

————————- etc.

```
temp <- edx_tidy$`8_genre` == "War"
edx_tidy$War[temp] <- 1

temp <- edx_tidy$`8_genre` == "Western"
edx_tidy$Western[temp] <- 1

temp <- edx_tidy$`8_genre` == "(no genres listed)"
edx_tidy$No_Genre[temp] <- 1
```

```
edx_tidy <- edx_tidy[,-c(6:13)]
edx_tidy <- edx_tidy%>% mutate(rated_year = year(as_datetime(edx_tidy$timestamp)))
edx_tidy <- edx_tidy[,-4]
edx_tidy <- edx_tidy[,c(1:3,25,4:24)]
head(edx_tidy)
```

```
##    userId movieId rating rated_year                         title Action
## 1:      1     122      5       1996              Boomerang (1992)      0
## 2:      1     185      5       1996               Net, The (1995)      1
## 3:      1     292      5       1996               Outbreak (1995)      1
## 4:      1     316      5       1996               Stargate (1994)      1
## 5:      1     329      5       1996 Star Trek: Generations (1994)      1
## 6:      1     355      5       1996       Flintstones, The (1994)      0
```

```
##    Adventure Animation Children Comedy Crime Documentary Drama Fantasy
## 1:         0         0        0      1     0           0     0       0
## 2:         0         0        0      0     1           0     0       0
## 3:         0         0        0      0     0           0     1       0
## 4:         1         0        0      0     0           0     0       0
## 5:         1         0        0      0     0           0     1       0
## 6:         0         0        1      1     0           0     0       1
##    Film_Noir Horror IMAX Musical Mystery Romance Sci_Fi Thriller War Western
## 1:         0      0    0       0       0       1      0        0   0       0
## 2:         0      0    0       0       0       0      0        1   0       0
## 3:         0      0    0       0       0       0      1        1   0       0
## 4:         0      0    0       0       0       0      1        0   0       0
## 5:         0      0    0       0       0       0      1        0   0       0
## 6:         0      0    0       0       0       0      0        0   0       0
##    No_Genre
## 1:        0
## 2:        0
## 3:        0
## 4:        0
## 5:        0
## 6:        0
```

a. Genres Analysis

```
genre_sum <- rep(0,20)
genre_rating <- rep(0,20)

genre_sum[1] <- sum(edx_tidy$Action)
  filtered_data <- edx_tidy %>% filter(Action == 1)
  genre_rating[1] <- round(mean(filtered_data$rating),digits = 2)

  genre_sum[2] <- sum(edx_tidy$Adventure)
  filtered_data <- edx_tidy %>% filter(Adventure == 1)
  genre_rating[2] <- round(mean(filtered_data$rating),digits = 2)
```

————etc.

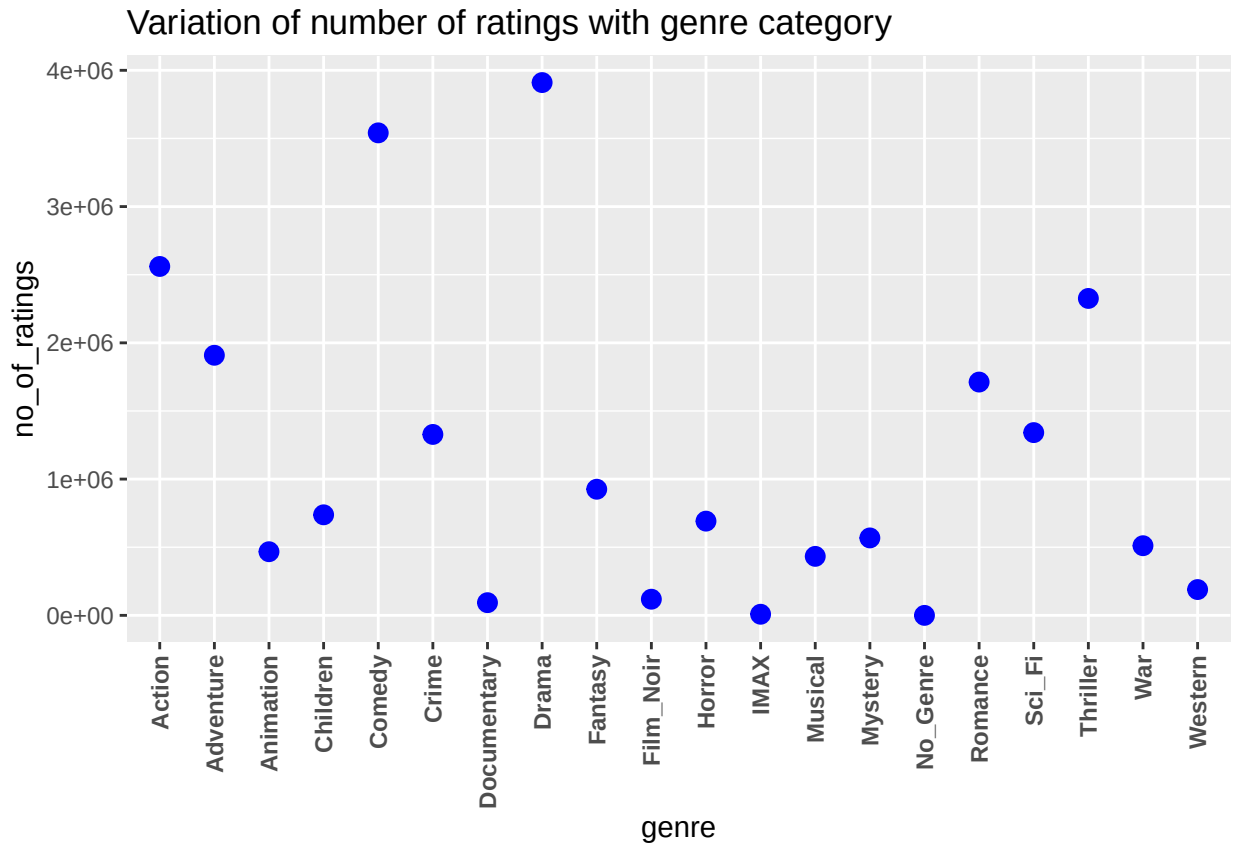
```
  genre_sum[20] <- sum(edx_tidy$No_Genre)
  filtered_data <- edx_tidy %>% filter(No_Genre == 1)
  genre_rating[20] <- round(mean(filtered_data$rating),digits = 2)
```

```
genre_data <- data.frame(genre = colnames(edx_tidy[,-c(1:5)]),no_of_ratings =
                                 genre_sum,rating_mean = genre_rating)
genre_data <- genre_data[order(genre_data$no_of_ratings,decreasing = TRUE),]
genre_data
```

```
##          genre no_of_ratings rating_mean
## 8        Drama       3910127        3.67
## 5       Comedy       3540930        3.44
## 1       Action       2560545        3.42
## 17    Thriller       2325899        3.51
```

```
## 2    Adventure       1908892        3.49
## 15     Romance       1712100        3.55
## 16      Sci_Fi       1341183        3.40
## 6        Crime       1327715        3.67
## 9      Fantasy        925637        3.50
## 4     Children        737994        3.42
## 11      Horror        691485        3.27
## 14     Mystery        568332        3.68
## 18         War        511147        3.78
## 3    Animation        467168        3.60
## 13     Musical        433080        3.56
## 19     Western        189394        3.56
## 10   Film_Noir        118541        4.01
## 7  Documentary         93066        3.78
## 12        IMAX          8181        3.77
## 20    No_Genre             7        3.64
```

```
genre_data %>% ggplot(aes(genre,no_of_ratings)) + geom_point(size = 3, color = "blue") +
  theme(axis.text.x = element_text(angle = 90, face = "bold",vjust = 0.5,hjust = 1)) +
  ggtitle("Variation of number of ratings with genre category")
```



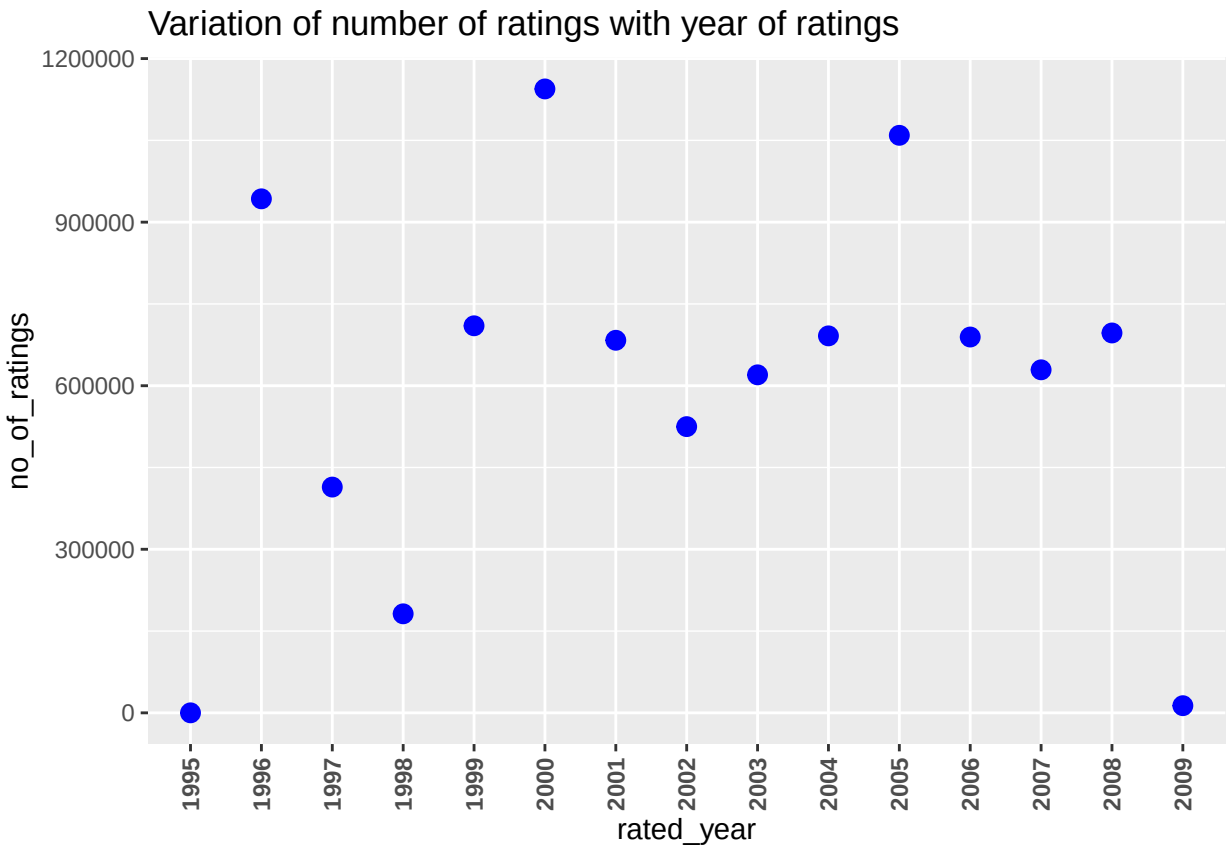
We can see that there are some genres which are rated more than others. The Drama, Comedy and Action are the most three rated genres. The average rating varies between 3.27 for Horror movies and 4.01 for Film_Noir movies. The Film_Noir movies are on the 17th place in the number of rating list. There is a small variation regarding average rating between the movie genre.

b. Rated Years Analysis

```
year_data <- edx_tidy %>% group_by(rated_year) %>% summarize(no_of_ratings = n(),
                                                             average_rating = mean(rating))
year_data[order(year_data$no_of_ratings,decreasing = TRUE),]
```

```
## # A tibble: 15 x 3
##    rated_year no_of_ratings average_rating
##         <dbl>         <int>          <dbl>
##  1       2000       1144349           3.58
##  2       2005       1059277           3.44
##  3       1996        942772           3.55
##  4       1999        709893           3.62
##  5       2008        696740           3.54
##  6       2004        691429           3.43
##  7       2006        689315           3.47
##  8       2001        683355           3.54
##  9       2007        629168           3.47
## 10       2003        619938           3.47
## 11       2002        524959           3.47
## 12       1997        414101           3.59
## 13       1998        181634           3.51
## 14       2009         13123           3.46
## 15       1995             2           4
```

```
year_data %>% ggplot(aes(factor(rated_year),no_of_ratings)) + geom_point(size = 3,
                                                                         color = "blue") +
 theme(axis.text.x = element_text(angle = 90, face = "bold",vjust = 0.5,hjust = 1)) +
 xlab("rated_year") +
 ggtitle("Variation of number of ratings with year of ratings")
```



With this summary we can see that years 2000 and 2005 has the most number of ratings. The average rating varies between 3.42 for 2004 year and 4 for 1995.

c. Movies Analysis

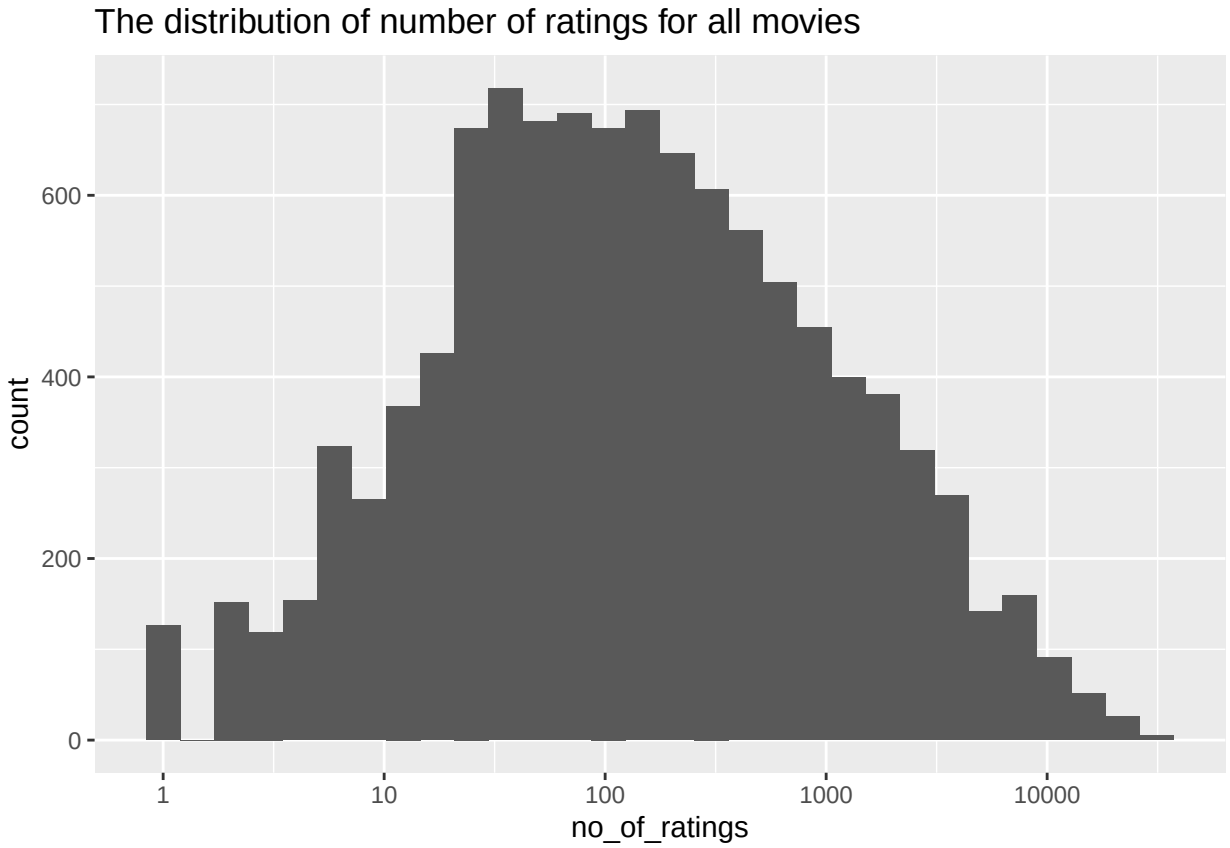
```
movieId_data <- edx_tidy %>% group_by(movieId) %>% summarize(no_of_ratings = n(),
                                                             average_rating = mean(rating))

movieId_data <- left_join(movieId_data,unique(edx_tidy %>% select(movieId,title)),
                                                                        by = "movieId")


movieId_data <- movieId_data[order(movieId_data$no_of_ratings,decreasing = TRUE),]

head(movieId_data)
```

```
## # A tibble: 6 x 4
##   movieId no_of_ratings average_rating title
##     <dbl>         <int>          <dbl> <chr>
## 1     296         31362           4.15 Pulp Fiction (1994)
## 2     356         31079           4.01 Forrest Gump (1994)
## 3     593         30382           4.20 Silence of the Lambs, The (1991)
## 4     480         29360           3.66 Jurassic Park (1993)
## 5     318         28015           4.46 Shawshank Redemption, The (1994)
```

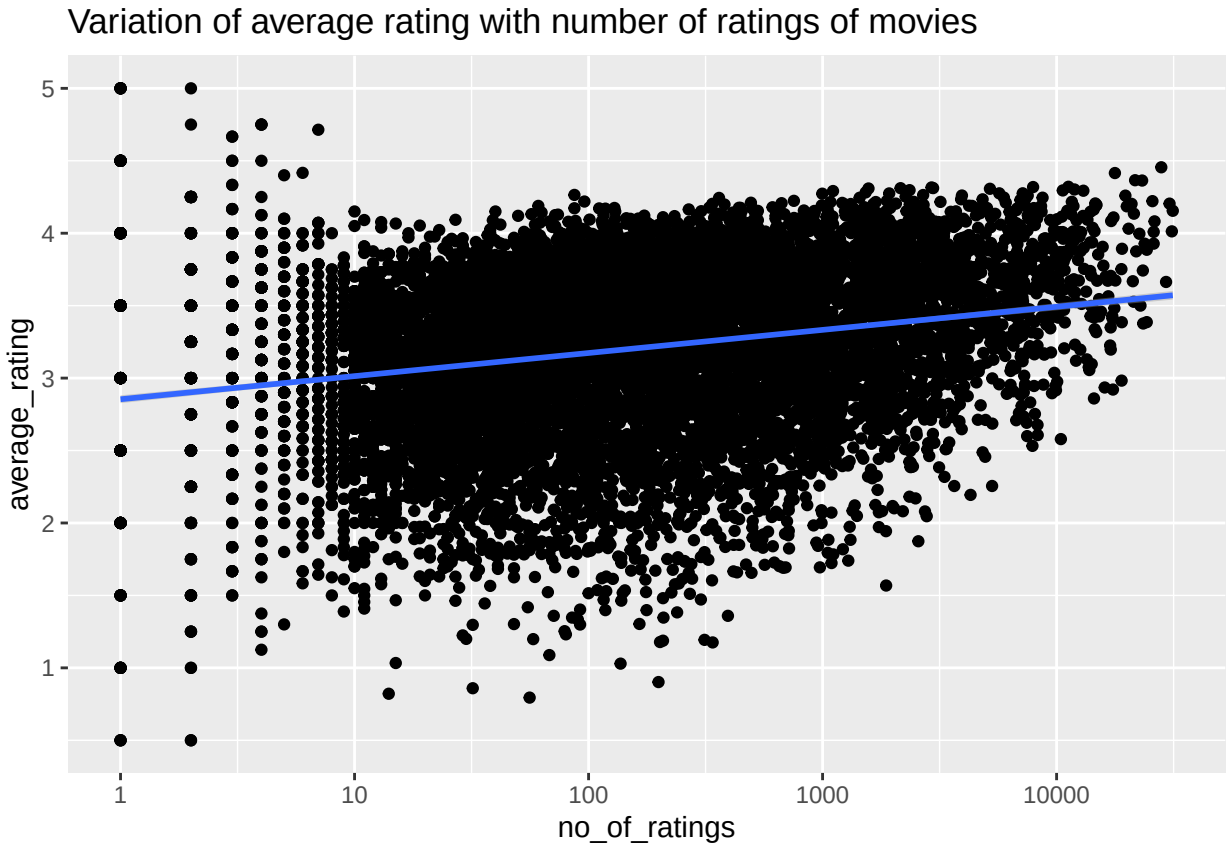
```
## 6     110         26212           4.08 Braveheart (1995)
```

There are movies that are more rated than others. Pulp Fiction, Forrest Gump and Silence of The Lambs are the first three rated movies.

```
movieId_data %>% ggplot(aes(no_of_ratings)) +
  geom_histogram(bins = 30) +
  scale_x_log10() +
  ggtitle("The distribution of number of ratings for all movies")
```




```
movieId_data %>% ggplot(aes(no_of_ratings,average_rating)) +
  geom_point() +
  scale_x_log10() +
  geom_smooth(method='lm') +
  ggtitle("Variation of average rating with number of ratings of movies")
```



We can see from this plot there is a positive influence of numbers of ratings of movies on average rating of movies.The slope of linear regression line is very small.

d. Users Analysis

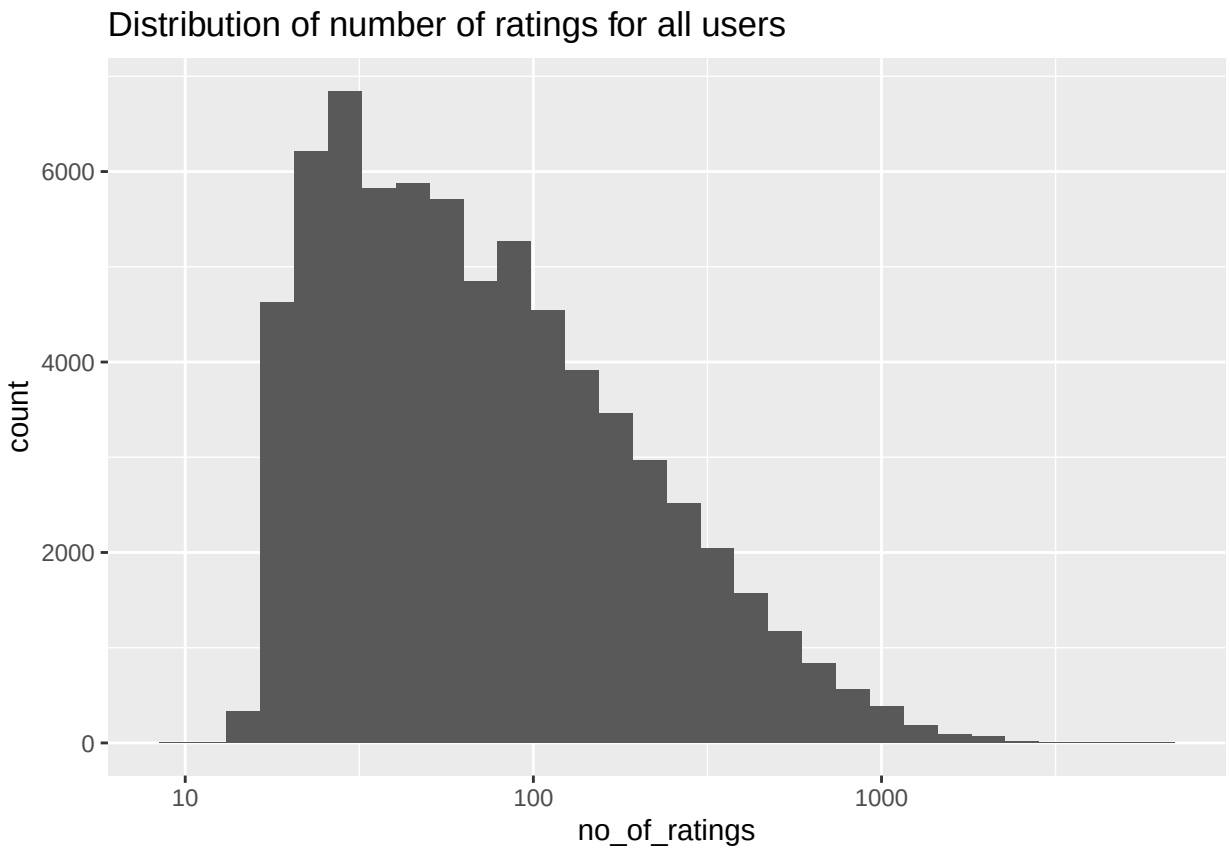
```
userId_data <- edx_tidy %>% group_by(userId) %>% summarize(no_of_ratings = n()
                                                           ,average_rating = mean(rating))
userId_data <- userId_data[order(userId_data$no_of_ratings,decreasing = TRUE),]
head(userId_data)
```

```
## # A tibble: 6 x 3
##   userId no_of_ratings average_rating
##    <int>         <int>          <dbl>
## 1  59269          6616           3.26
## 2  67385          6360           3.20
## 3  14463          4648           2.40
## 4  68259          4036           3.58
## 5  27468          4023           3.83
## 6  19635          3771           3.50
```

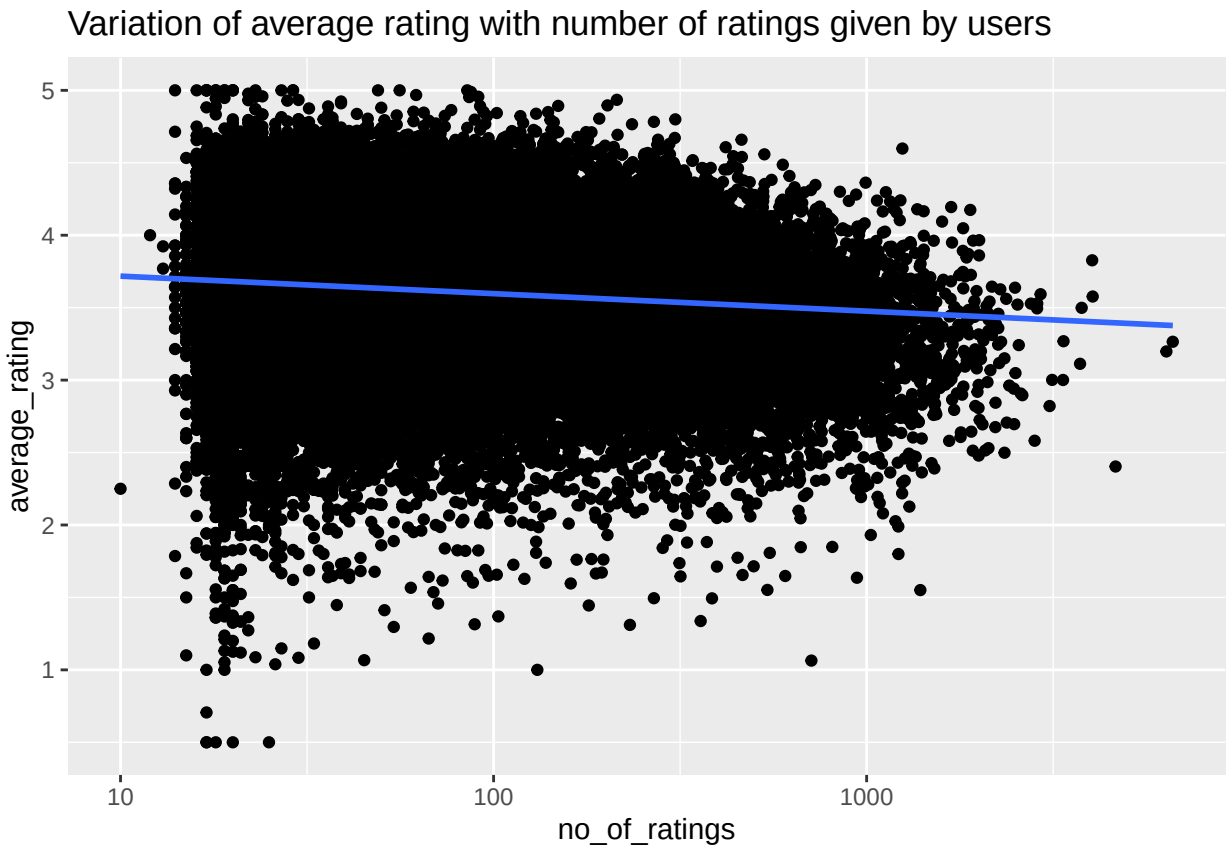
There are users who give more ratings than others

```
userId_data %>% ggplot(aes(no_of_ratings)) +
  geom_histogram(bins = 30) +
```

```
  scale_x_log10() +
  ggtitle("Distribution of number of ratings for all users")
```




```
userId_data %>% ggplot(aes(no_of_ratings,average_rating)) +
  geom_point() +
  scale_x_log10() +
  geom_smooth(method = 'lm') +
  ggtitle("Variation of average rating with number of ratings given by users")
```



Modeling Approach

For this model i use the regularisation based approach. We saw from the exploratory data analysis that there are movies who a rated very few times and there are users who give very few ratings. So I put a penalty term in the formula for calculating the biases.

Accuracy of the model will be evaluated using the residual mean squared error (RMSE):

RMSE = sqrt(mean((data_rating - predicted_rating)^2))

- data rating = rating for movie i by user u,
- predicted rating = sum of average rating and movies biases,users biases, year biases and genre biases.

I start by predicting the same rating for all movies and all users.The value who minimizes the RMSE is the average value of all ratings.

```
validation_tidy <- validation %>% separate(genres,c("1_genre","2_genre",
                                                    "3_genre","4_genre",
                                                    "5_genre","6_genre",
                                                    "7_genre","8_genre"),
                                                    "\\|")

individual_genre <- unique(c(unique(validation_tidy$`1_genre`),
                             unique(validation_tidy$`2_genre`),
                             unique(validation_tidy$`3_genre`),
```

```
                              unique(validation_tidy$`4_genre`),
                              unique(validation_tidy$`5_genre`),
                              unique(validation_tidy$`6_genre`),
                              unique(validation_tidy$`7_genre`),
                              unique(validation_tidy$`8_genre`)))
individual_genre <-individual_genre[order(individual_genre)]

validation_tidy <- validation_tidy %>% mutate(Action = 0,Adventure = 0,
                                              Animation = 0,Children = 0,
                                              Comedy = 0,Crime = 0,
                                              Documentary = 0,Drama = 0,
                                              Fantasy = 0,Film_Noir = 0,
                                              Horror = 0,IMAX = 0,Musical = 0,
                                              Mystery = 0,Romance = 0,
                                              Sci_Fi = 0,Thriller = 0,
                                              War = 0, Western = 0,
                                              No_Genre = 0)


temp <- validation_tidy$`1_genre` == "Action"
validation_tidy$Action[temp] <- 1

temp <- validation_tidy$`1_genre` == "Adventure"
validation_tidy$Adventure[temp] <- 1

temp <- validation_tidy$`1_genre` == "Animation"
validation_tidy$Animation[temp] <- 1
```

————————etc.

```
temp <- validation_tidy$`8_genre` == "War"
validation_tidy$War[temp] <- 1

temp <- validation_tidy$`8_genre` == "Western"
validation_tidy$Western[temp] <- 1

temp <- validation_tidy$`8_genre` == "(no genres listed)"
validation_tidy$No_Genre[temp] <- 1


validation_tidy <- validation_tidy[,-c(6:13)]
validation_tidy <- validation_tidy%>% mutate(rated_year =
                     year(as_datetime(validation_tidy$timestamp)))
validation_tidy <- validation_tidy[,-4]
validation_tidy <- validation_tidy[,c(1:3,25,4:24)]
```

```
avg_rating_edx <- mean(edx_tidy$rating)


avg_rating_edx
```

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

a.Movie effect

Calculating the movie bias b_i:

```
movie_effect <- edx_tidy %>% group_by(movieId) %>%
  summarize(b_i = sum(rating-avg_rating_edx)/(n()+5))
head(movie_effect)
```

```
## # A tibble: 6 x 2
##   movieId    b_i
##     <dbl>  <dbl>
## 1       1  0.415
## 2       2 -0.307
## 3       3 -0.365
## 4       4 -0.646
## 5       5 -0.443
## 6       6  0.303
```

```
edx_tidy <- edx_tidy %>% left_join(movie_effect, by = 'movieId')
head(edx_tidy[,c(1:4,26)])
```

```
##    userId movieId rating rated_year         b_i
## 1:      1     122      5       1996 -0.65238168
## 2:      1     185      5       1996 -0.38298900
## 3:      1     292      5       1996 -0.09442186
## 4:      1     316      5       1996 -0.16274038
## 5:      1     329      5       1996 -0.17494804
## 6:      1     355      5       1996 -1.02361857
```

Populating the validation set with the movies bias b_i:

```
validation_tidy <- left_join(validation_tidy,movie_effect, by = 'movieId')
head(validation_tidy[,c(1:4,26)])
```

```
##    userId movieId rating rated_year         b_i
## 1:      1     231      5       1996 -0.57716427
## 2:      1     480      5       1996  0.15103088
## 3:      1     586      5       1996 -0.45664758
## 4:      2     151      3       1997  0.01758101
## 5:      2     858      2       1997  0.90264647
## 6:      2    1544      3       1997 -0.56680008
```

b.User effect

Calculating the user bias b_u:

```
user_effect <- edx_tidy %>% group_by(userId) %>%
  summarize(b_u = sum(rating - b_i - avg_rating_edx)/(n()+5))
head(user_effect)
```

```
## # A tibble: 6 x 2
##   userId     b_u
##    <int>   <dbl>
## 1      1  1.33
## 2      2 -0.183
## 3      3  0.228
## 4      4  0.571
## 5      5  0.0803
## 6      6  0.306
```

```
edx_tidy <- edx_tidy %>% left_join(user_effect, by = 'userId')
head(edx_tidy[,c(1:4,26,27)])
```

```
##    userId movieId rating rated_year         b_i      b_u
## 1:      1     122      5       1996 -0.65238168 1.329212
## 2:      1     185      5       1996 -0.38298900 1.329212
## 3:      1     292      5       1996 -0.09442186 1.329212
## 4:      1     316      5       1996 -0.16274038 1.329212
## 5:      1     329      5       1996 -0.17494804 1.329212
## 6:      1     355      5       1996 -1.02361857 1.329212
```

Populating the validation set with the users bias b_u:

```
validation_tidy <- left_join(validation_tidy,user_effect,by = 'userId')
head(validation_tidy[,c(1:4,26,27)])
```

```
##    userId movieId rating rated_year         b_i        b_u
## 1:      1     231      5       1996 -0.57716427  1.3292115
## 2:      1     480      5       1996  0.15103088  1.3292115
## 3:      1     586      5       1996 -0.45664758  1.3292115
## 4:      2     151      3       1997  0.01758101 -0.1827251
## 5:      2     858      2       1997  0.90264647 -0.1827251
## 6:      2    1544      3       1997 -0.56680008 -0.1827251
```

c. Year effect

Calculating the year effect b_y:

```
year_effect <- edx_tidy %>% group_by(rated_year) %>%
  summarize(b_y = sum(rating - b_i - b_u - avg_rating_edx)/(n()+5))
head(year_effect)
```

```
## # A tibble: 6 x 2
##   rated_year     b_y
##        <dbl>   <dbl>
## 1       1995 0.0620
## 2       1996 0.00899
## 3       1997 0.00815
## 4       1998 0.00130
## 5       1999 0.00572
## 6       2000 0.00464
```

```
edx_tidy <- edx_tidy %>% left_join(year_effect, by = 'rated_year')
head(edx_tidy[,c(1:4,26,27,28)])
```

```
##    userId movieId rating rated_year         b_i      b_u         b_y
## 1:      1     122      5       1996 -0.65238168 1.329212 0.008985188
## 2:      1     185      5       1996 -0.38298900 1.329212 0.008985188
## 3:      1     292      5       1996 -0.09442186 1.329212 0.008985188
## 4:      1     316      5       1996 -0.16274038 1.329212 0.008985188
## 5:      1     329      5       1996 -0.17494804 1.329212 0.008985188
## 6:      1     355      5       1996 -1.02361857 1.329212 0.008985188
```

Populating the validation set with the years bias b_i:

```
validation_tidy <- left_join(validation_tidy,year_effect,by = 'rated_year')
head(validation_tidy[,c(1:4,26:28)])
```

```
##    userId movieId rating rated_year         b_i        b_u         b_y
## 1:      1     231      5       1996 -0.57716427  1.3292115 0.008985188
## 2:      1     480      5       1996  0.15103088  1.3292115 0.008985188
## 3:      1     586      5       1996 -0.45664758  1.3292115 0.008985188
## 4:      2     151      3       1997  0.01758101 -0.1827251 0.008145839
## 5:      2     858      2       1997  0.90264647 -0.1827251 0.008145839
## 6:      2    1544      3       1997 -0.56680008 -0.1827251 0.008145839
```

d. Genre effect

There are 20 individual genres. The genre bias is the sum of each individual genre.

```
edx_tidy <- edx_tidy %>% mutate(b_genre = 0)
validation_tidy <- validation_tidy %>% mutate(b_genre = 0)


filtered_data <- edx_tidy %>% filter(Action == 1)

effects_value <- filtered_data$rating - filtered_data$b_i - filtered_data$b_u -
                    filtered_data$b_y -  avg_rating_edx - filtered_data$b_genre

genre_bias_value <- sum(effects_value)/(nrow(filtered_data)+5)

edx_tidy <- edx_tidy %>% mutate(b_genre = ifelse(Action == 1,
                                                 genre_bias_value,0)+b_genre)

validation_tidy <-validation_tidy %>% mutate(b_genre = ifelse(Action == 1,
                                           genre_bias_value,0) + b_genre)


filtered_data <- edx_tidy %>% filter(Adventure == 1)

effects_value <- filtered_data$rating - filtered_data$b_i - filtered_data$b_u -
                    filtered_data$b_y -  avg_rating_edx - filtered_data$b_genre
```

```
genre_bias_value <- sum(effects_value)/(nrow(filtered_data)+5)

edx_tidy <- edx_tidy %>% mutate(b_genre = ifelse(Adventure == 1,
                                                 genre_bias_value,0)+b_genre)

validation_tidy <-validation_tidy %>% mutate(b_genre = ifelse(Adventure == 1,
                                               genre_bias_value,0) + b_genre)
```

————————etc.

```
filtered_data <- edx_tidy %>% filter(No_Genre == 1)

effects_value <- filtered_data$rating - filtered_data$b_i - filtered_data$b_u -
                    filtered_data$b_y -  avg_rating_edx - filtered_data$b_genre

genre_bias_value <- sum(effects_value)/(nrow(filtered_data)+5)

edx_tidy <- edx_tidy %>% mutate(b_genre =
                              ifelse(No_Genre == 1,genre_bias_value,0)+b_genre)

validation_tidy <-validation_tidy %>% mutate(b_genre =
                            ifelse(No_Genre == 1,genre_bias_value,0) + b_genre)

head(validation_tidy[,c(1:4,26:29)])
```

```
##    userId movieId rating rated_year         b_i        b_u         b_y
## 1:      1     231      5       1996 -0.57716427  1.3292115 0.008985188
## 2:      1     480      5       1996  0.15103088  1.3292115 0.008985188
## 3:      1     586      5       1996 -0.45664758  1.3292115 0.008985188
## 4:      2     151      3       1997  0.01758101 -0.1827251 0.008145839
## 5:      2     858      2       1997  0.90264647 -0.1827251 0.008145839
## 6:      2    1544      3       1997 -0.56680008 -0.1827251 0.008145839
##         b_genre
## 1:  0.002707893
## 2: -0.028548633
## 3: -0.012231164
## 4: -0.007999940
## 5:  0.024068076
## 6: -0.025141707
```

Pred_rating = avg_rating + b_i + b_u +b_y + b_genre

```
pred_rating <- avg_rating_edx + validation_tidy$b_i + validation_tidy$b_u +
                                validation_tidy$b_y + validation_tidy$b_genre
```

Result

Residual mean squared error RMSE is:

```
RMSE(validation_tidy$rating,pred_rating)
```

```
## [1] 0.8646609
```

Conclusion

The RMSE is 0.8646 less than 0.8649 the minimum value put for the project and it means that we can trust our predicted values for movie ratings.