The Visualization of Movie Datasets Analysis and the Realization of Recommendation System

Zhao Yeyu 21620018

11/20/2016

1 Description

This report attempts to use R language to do some useful and deep data analysis. R is used to analyze the movies data, users data and ratings data included in the online movie review websites. Based on the results of statistical analysis, this paper then constructs the TopN recommendation model. The report is divided into two parts. In the first part, the data sets are merged, transformed and analyzed using some built-in functions and self-defined functions after data importing and preprocessing. For more convenient to be observed, the results are visualized by using visualization functions such as ggplot. In the second part, the user similarity data set is constructed based on the dataset. In addition, recommendation system is created by using the collaborative filtering algorithm based on user history dataset.

2 Datasets

There are three datasets included in this report. The entire datasets contain more than 1 million ratings from 6,040 users on 4,000 movies.

2.1 Movies

Attributes	Description
movieid	MovieIDs range between 1 and 3952
title	Titles are identical to titles provided by the IMDB (including year of release)
genres	Each movie can belong to multiple genres

Genres are pipe-separated and are selected from the following genres:

- Action
- Adventure
- Animation
- Children's
- Comedy
- Crime
- Documentary
- Drama
- Fantasy
- Film-Noir
- Horror
- Musical
- Mystery
- Romance

- Sci-Fi
- Thriller
- War
- Western

2.2 Users

Attributes	Description
userid	UserIDs range between 1 and 6040
gender	Gender is denoted by a "M" for male and "F" for female
age	Age is divided into seven age groups according to the range
job	0-20 represent 21 different occupations respectively
zipcode	Every user's zipcode of location

Age is chosen from the following ranges:

- 1: "Under 18"
- 18: "18-24"
- 25: "25-34"
- 35: "35-44"
- 45: "45-49"
- 50: "50-55"
- 56: "56+"

Occupation is chosen from the following choices:

- 0: "other" or not specified
- 1: "academic/educator"
- 2: "artist"
- 3: "clerical/admin"
- 4: "college/grad student"
- 5: "customer service"
- 6: "doctor/health care"
- 7: "executive/managerial"
- 8: "farmer"
- 9: "homemaker"
- 10: "K-12 student"
- 11: "lawyer"
- 12: "programmer"
- 13: "retired"
- 14: "sales/marketing"
- 15: "scientist"
- 16: "self-employed"
- 17: "technician/engineer"
- 18: "tradesman/craftsman"
- 19: "unemployed"
- 20: "writer"

2.3 Ratings

Attributes	Description
userid	UserIDs range between 1 and 6040
movieid	MovieIDs range between 1 and 3952
rating	Ratings are made on a 5-star scale (whole-star ratings only)
timestamp	Timestamp is represented in seconds since the epoch as returned by time(2)

3 Import data

As the R language can not directly read the ".dat" file, I use the MySQL to read the ".dat" file into database firstly. Then, RMySQL package is used to read data from the database.

4 Data Analysis of "movies.dat"

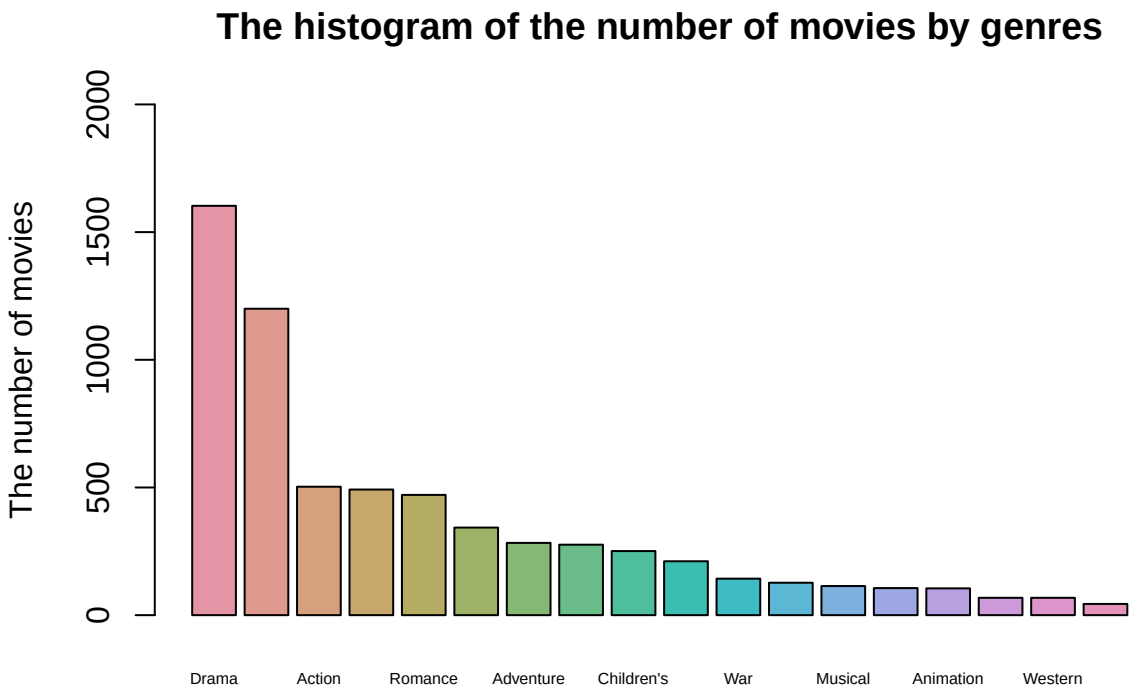
4.1 Data Preprocessing

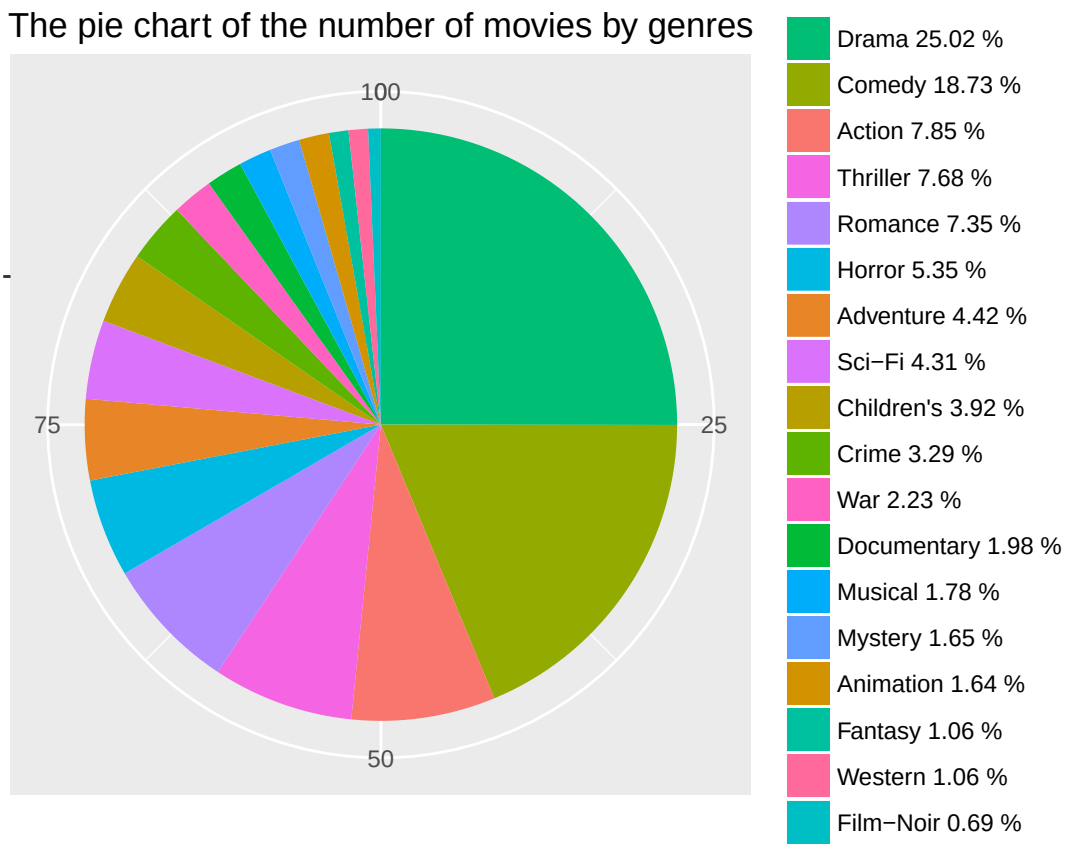
After importing the data, data preprocessing needed to be employed. There are two aspects in data preprocessing. First, I used the regular expression to extract the movie release year from the column of "title", and generated a new column named "year" to store year data. Then, the column of "genres" in dataset is not convenient for me to process, so that I need to convert a categorical variable into a "dummy" or "indicator" matrix. If a column in a dataframe has k distinct values, we would derive a matrix or dataframe containing k columns containing all 1's and 0's. As a result, I have converted the column into "dummy" matrix, and combined the matrix with movie dataset. The output dataset is called "movies_final", which is ordered by "year".

4.2 Data Analysis

4.2.1 Statistics of movies genres

First of all, I counted the number of different genres of movies in the data set, and then plotted the histogram and pie chart respectively.



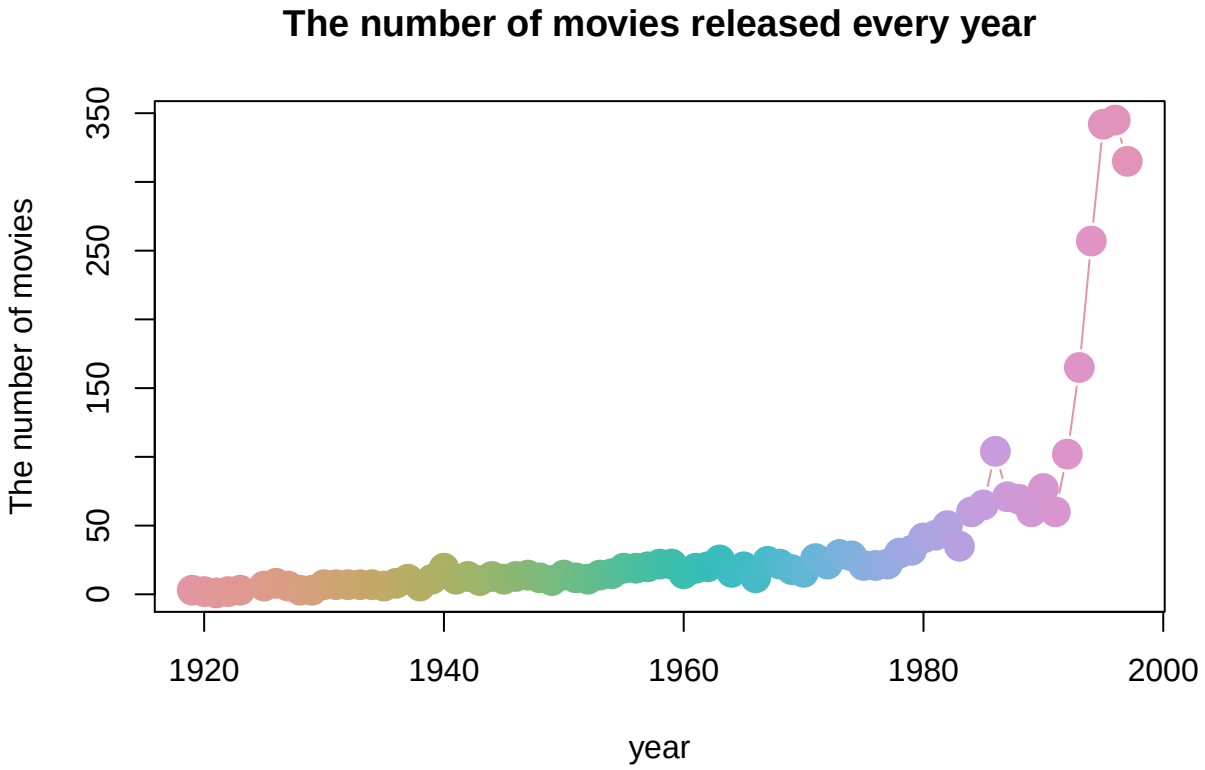


From the histogram, we learn that Drama and Comedy accounted for the majority in the dataset. Compared to the minority, these two types of movies released far more than the Fantasy, Western and other types. From the pie chart, it's obvious that the number of Drama, Comedy and Action movies accounted for more than

50% of all movies. While the smallest number of films is less than 1% of all movies. Thinking about the actual experience, it is not difficult to find the reason. As a result of cultural differences between domestic and abroad, foreigners have stronger preferences to Drama, Comedy and Action, and thus the release of such movies would attract more custormers, indicating that the number of such movies released more; Compared with popular genres of movies, Western and Firm-Noir were rarely released.

4.2.2 Time Series statistics of movies released

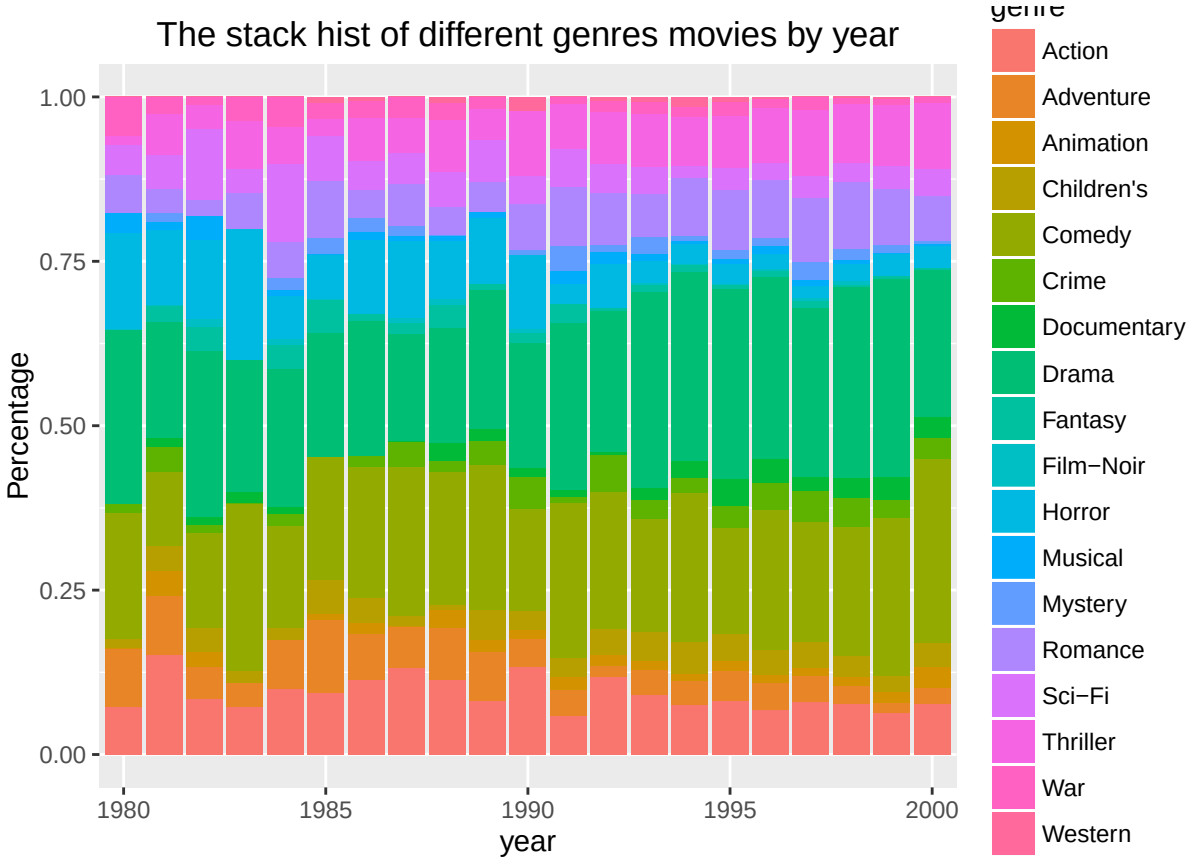
In accordance with the time series (year), I summed up the number of movies released each year and plotted the line graph.



From the line graph, we can see the number of movies released each year that in the dataset showed an upward trend. Although there have been some small fluctuations in the middle.

4.2.3 Time Series statistics of different genres of movies released

In this part, I would analyze the changes in the number of movies of different genres over time. In order to eliminate differences of the number of movies released each year, I present the data in the form of a pile-up scale histogram.



From the results, most of the different genres of movies released each year remained stable in proportion from 1990 to 2000. But there have been some changes.

- From 1980 to 2000, the release of the Comedy gradually increased in general, indicating that the market have more preferences to comdey;
- From 1980 to 2000, the release of the Musical gradually decreased in general. There are two main reasons. On the one hand, with the social development, the market acceptance of the Musical gradually reduced; on the other hand, the Musical can not give producer sufficient profits;
- The proporion of Drama, Action, Romance and other popular movie genres that are in line with public tastes is almost unchanged.

5 Data Analysis of "ratings.dat"

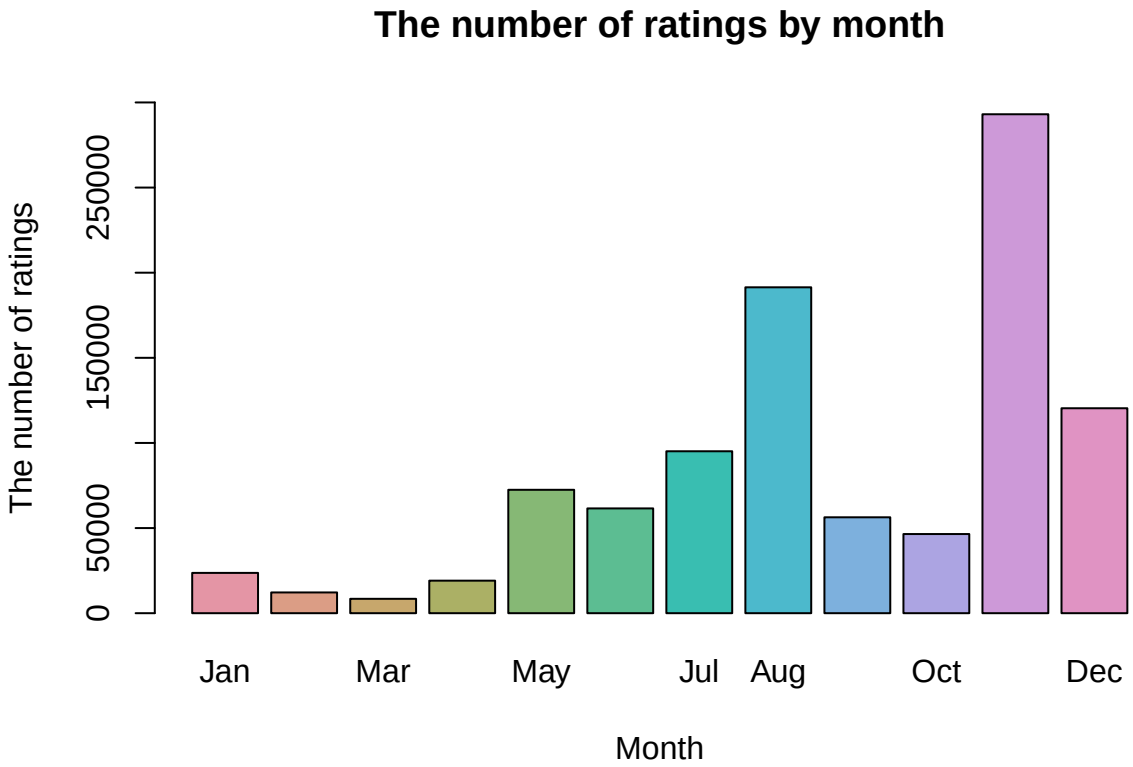
5.1 Data Preprocessing

The column "timestamp" of ratings dataset represents seconds since midnight Coordinated Universal Time (UTC) of January 1, 1970. First, in order to process the data conveniently, I converted the format of "timestamp" to "yyyy-mm-dd hh:mm:ss", and saved these data as new columns.

5.2 Data Analysis

5.2.1 Statistics of "Peak Month"

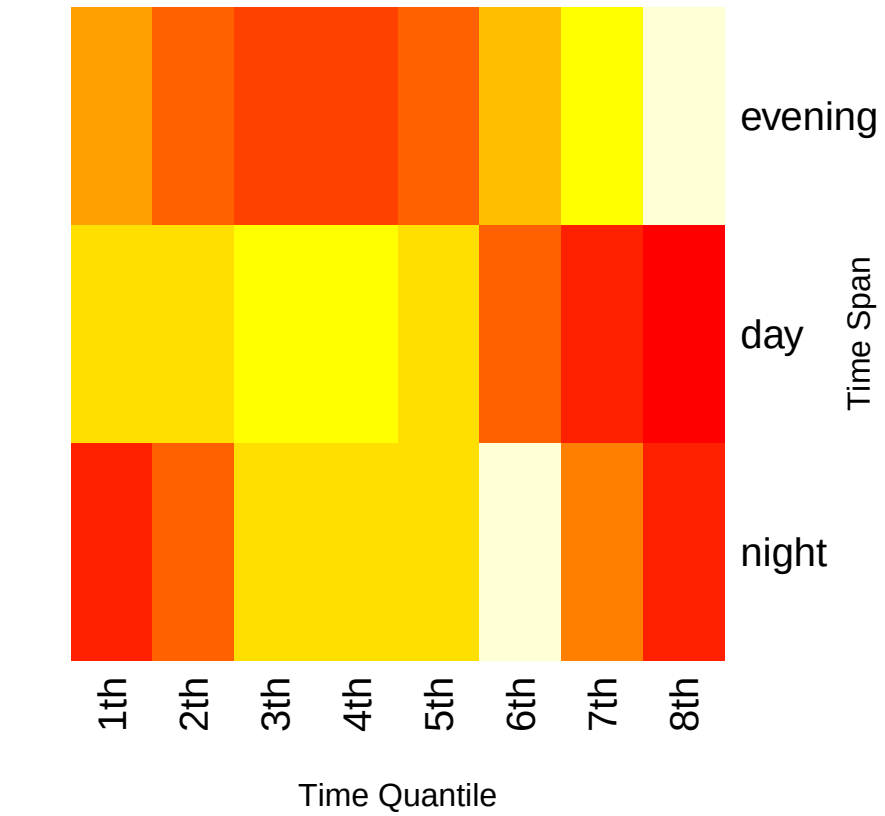
In this part, I want to understand when the users post their comments. First, I count the number of comments that users posted by month, and presented the result by hisogram.



From the histogram, we can see that the number of comments posted each month varies widely. July, August, November and December are the peak months that users posted their comments, indicating that most users choose to watch online movies in the winter and summer vacation.

5.2.2 Statistics of "Rush Hour"

After finishing the statistics of "peak month", the next step is to calculate the "rush hour" of users' comments. Here, I use heatmap to visualize the results.



In the heat map, each row represents a time period (night, day, evening), the horizontal axis represents the hour of current time period. Night is from 0:00 to 7:00(1th to 8th), day is from 8:00 to 15:00(1th to 8th), evening is from 16:00 to 23:00(1th to 8th). We learn that the rush hour of posting comments is among midnight, 7am, 2pm - 3pm and 6pm - 7pm.

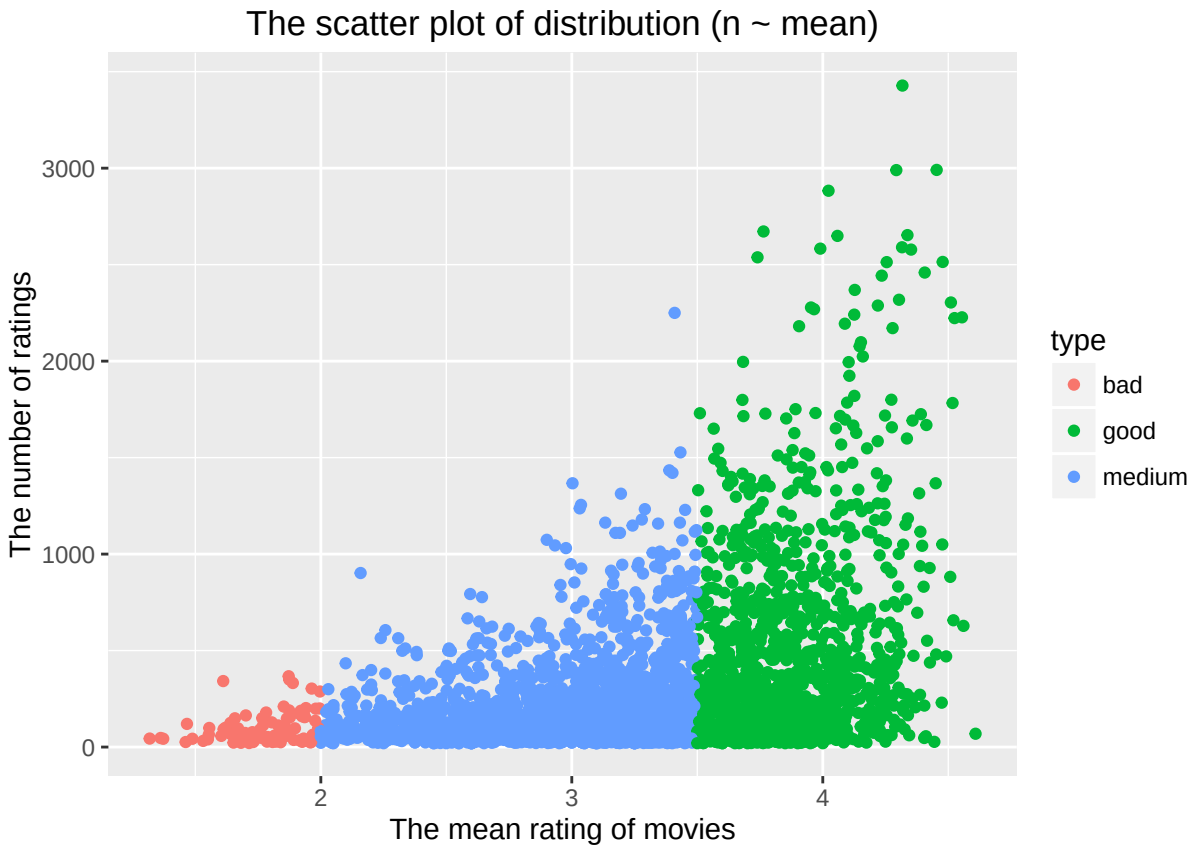
6 Data Analysis of Merged Dataset

6.1 Data Preprocessing

The preprocessing need to be employed before analysis. In order to avoid the large deviation of results, I removed the movies which has the small number of ratings less than 20. Next, I calculate the average score for each movie, and classify the movies according to the average score. Movies with ratings greater than 3.5 are classified as "good," and those with a score less than 2 are classified as "bad", others are classified as "medium".

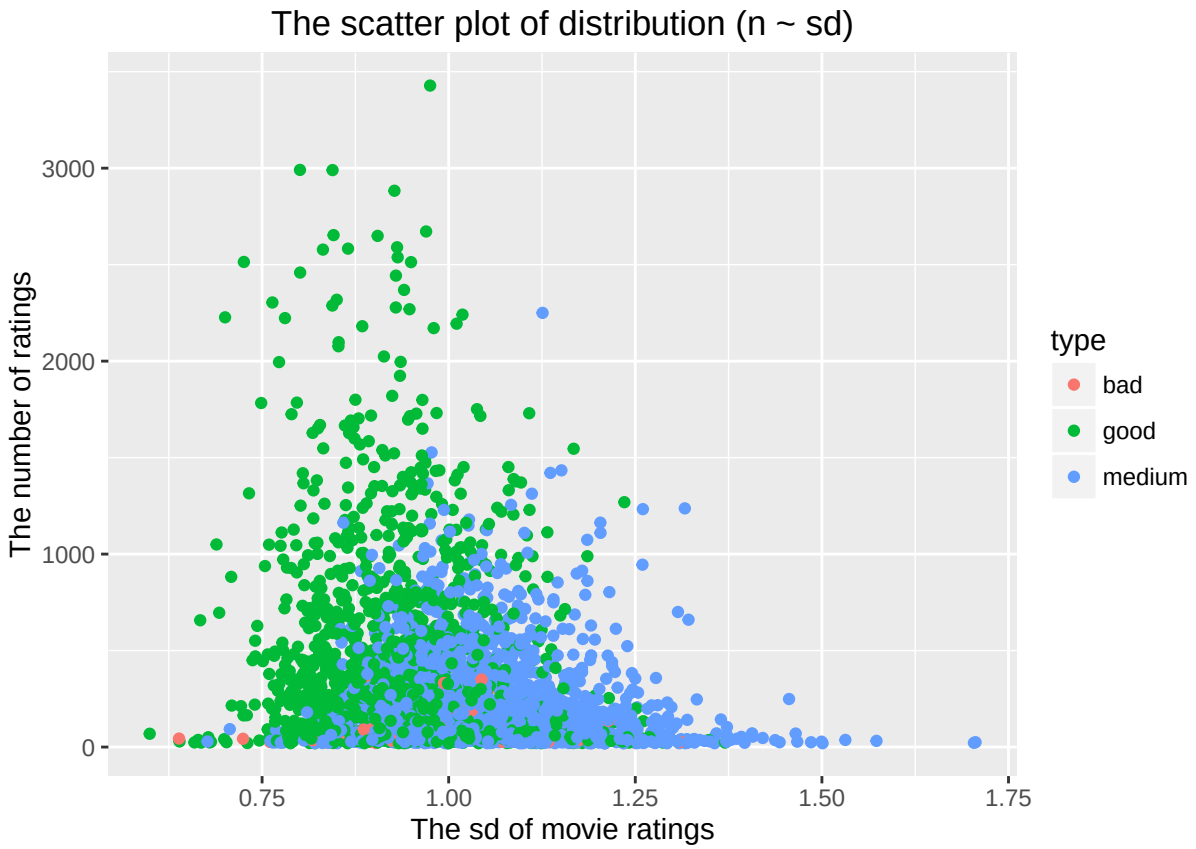
6.2 Data Analysis

After data preprocessing, I will start to calculate the mean, standard deviation, and the number of reviews for each movie, and visualize the results.



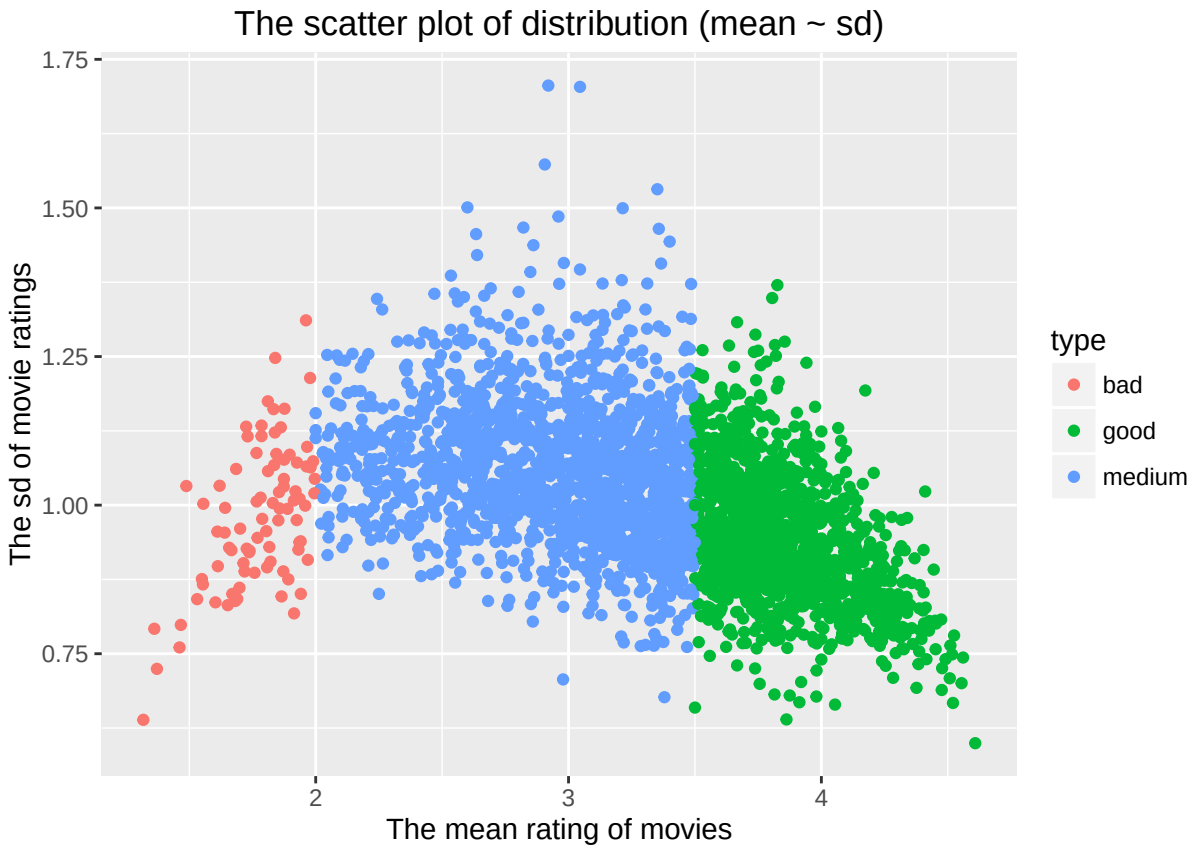
It can be seen from the figure that the number of ratings of movies has also increased with the increase in movie ratings. Compared with the "bad" movies, a number of "good" movies have more than 1,000 ratings. The number of ratings of all "bad" movies is below 500.

The scatter plot of the number of ratings and sd is as follows.



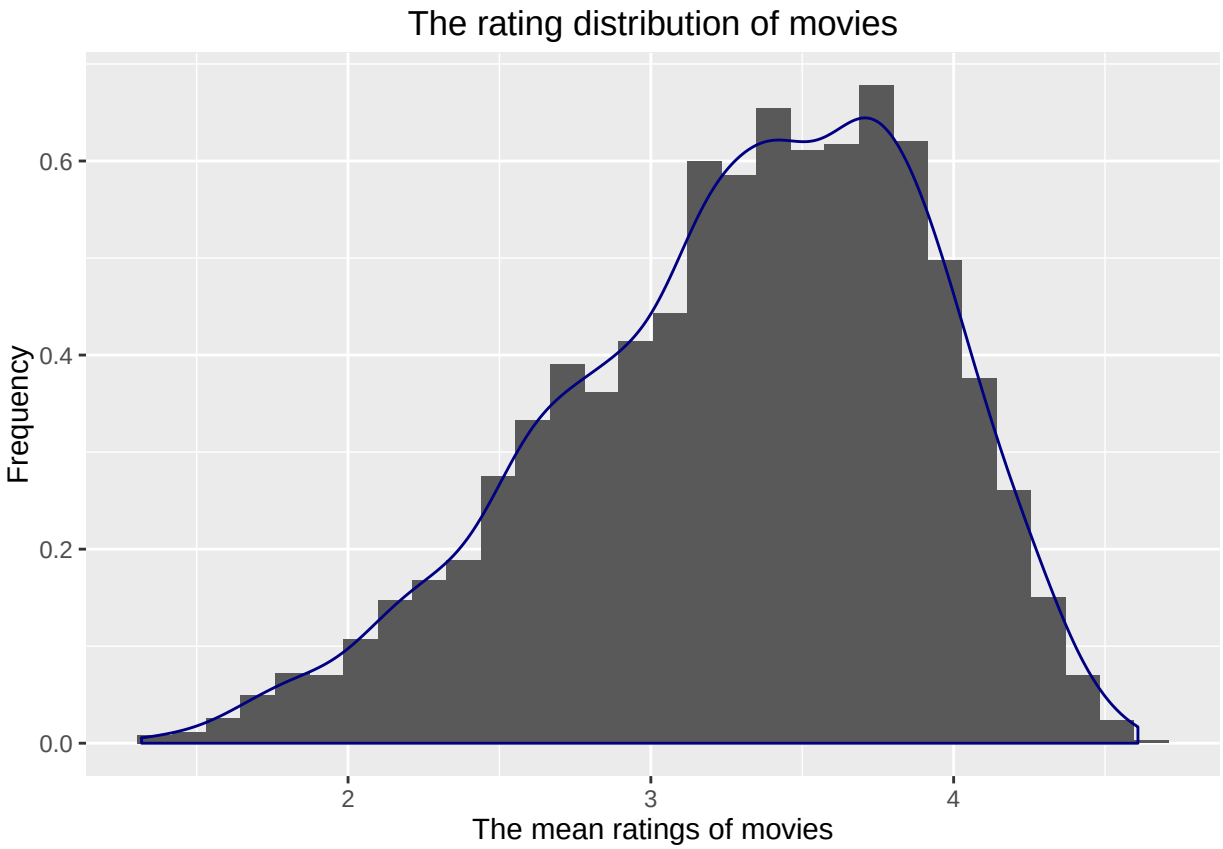
As can be seen from the scatter plot, the standard deviation of most movies are less than 1.5, only a few of the movie ratings differ largely(sd> 1.5). Good movies are basically distributed in left of the figure, indicating that user ratings for good movies are more consistent. In contrast, the differences in the ratings of "medium" movies are relatively large.

The scattor plot of the mean of ratings and sd is as follows.



It can be seen clearly from the figure that the approximate distribution of the scatters is close to parabolic. The differences of scores in "good" movies and "bad" movies are relatively small, while the differences of scores in "medium" movies are larger. This is because the user acceptance of these two types of movies("bad" and "good") is relatively consistent, while for movies which have relatively modest scores, evaluations from different users diverge largely.

Finally, I would like to plot the distribution histogram of the average score of the movies.



As can be seen from the figure, the distribution of the movie score roughly obeys the normal distribution. The overall average score is about 3.7, indicating that most movies have a relatively high score.

7 Data Analysis of "users.dat"

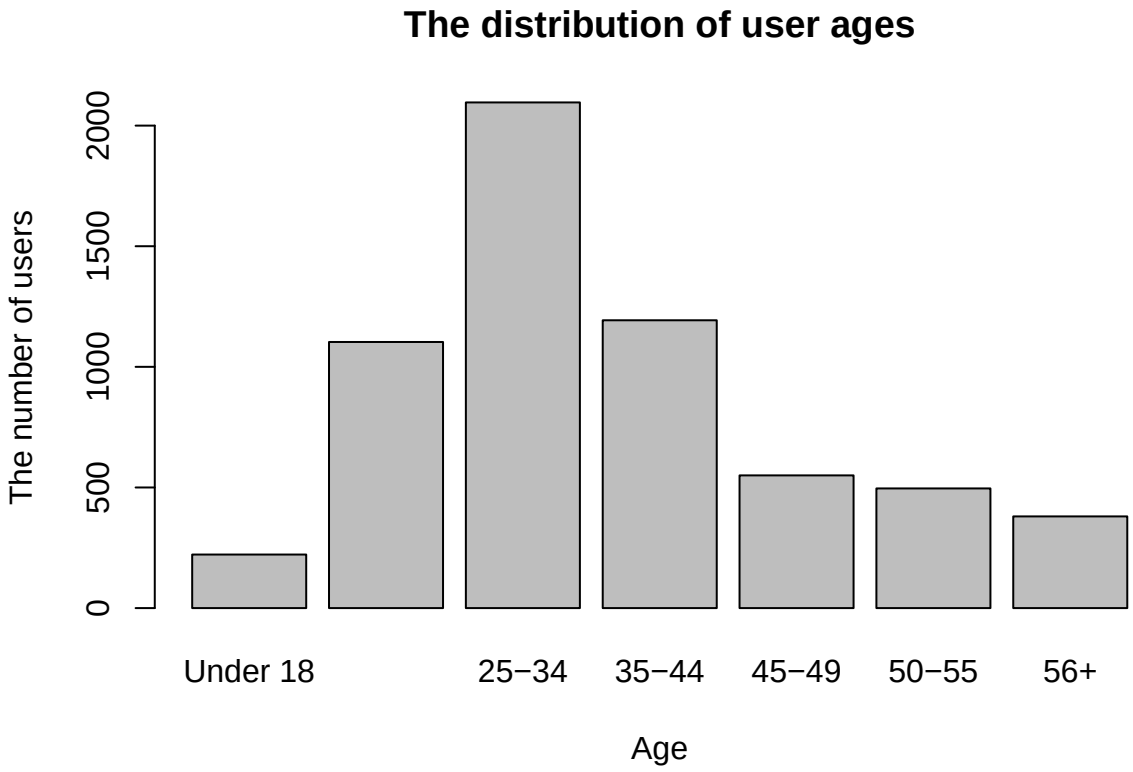
7.1 Data Preprocessing

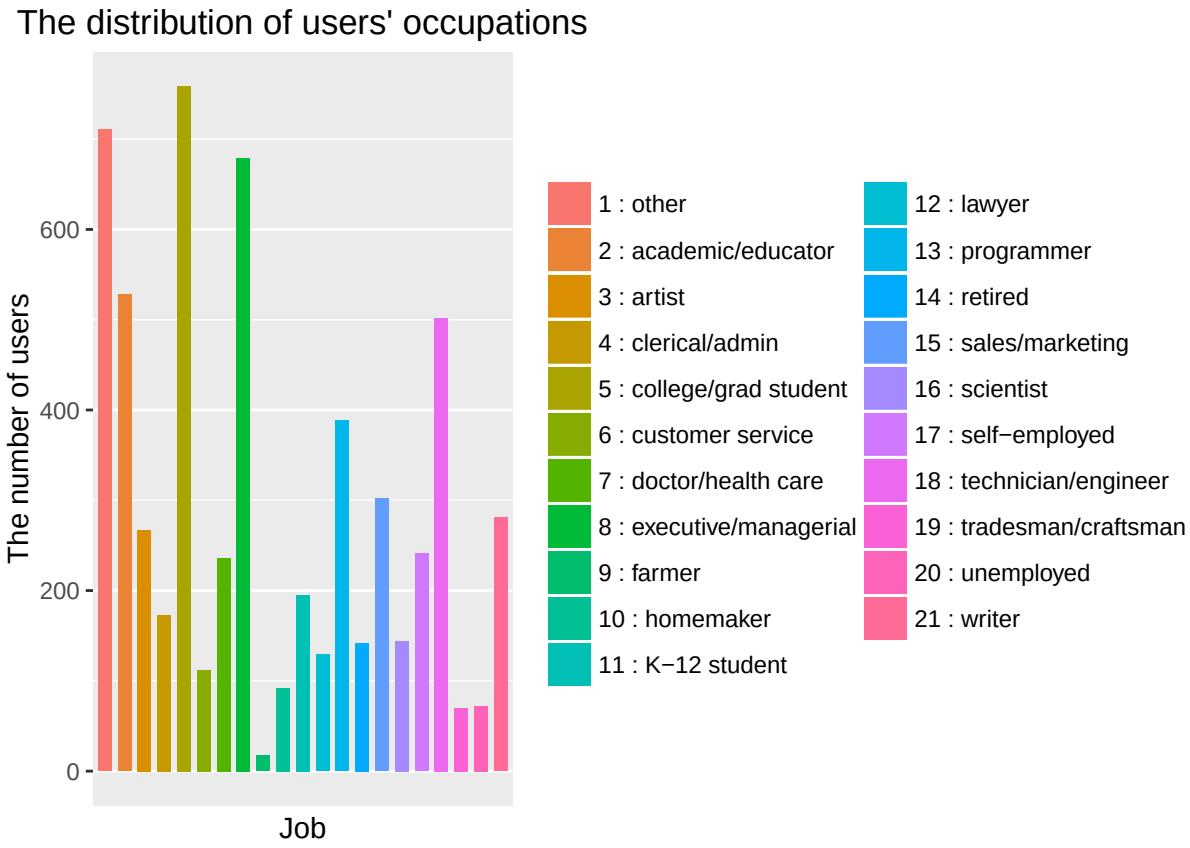
Before data analysis, I merge table "movies_ratings" and "users" into one table firstly.

7.2 Data Analysis

7.2.1 User Information Analysis

In this part, I carry out basic statistics on user information firstly, including the user's age distribution and the user's occupational distribution. The distribution diagrams are shown below.



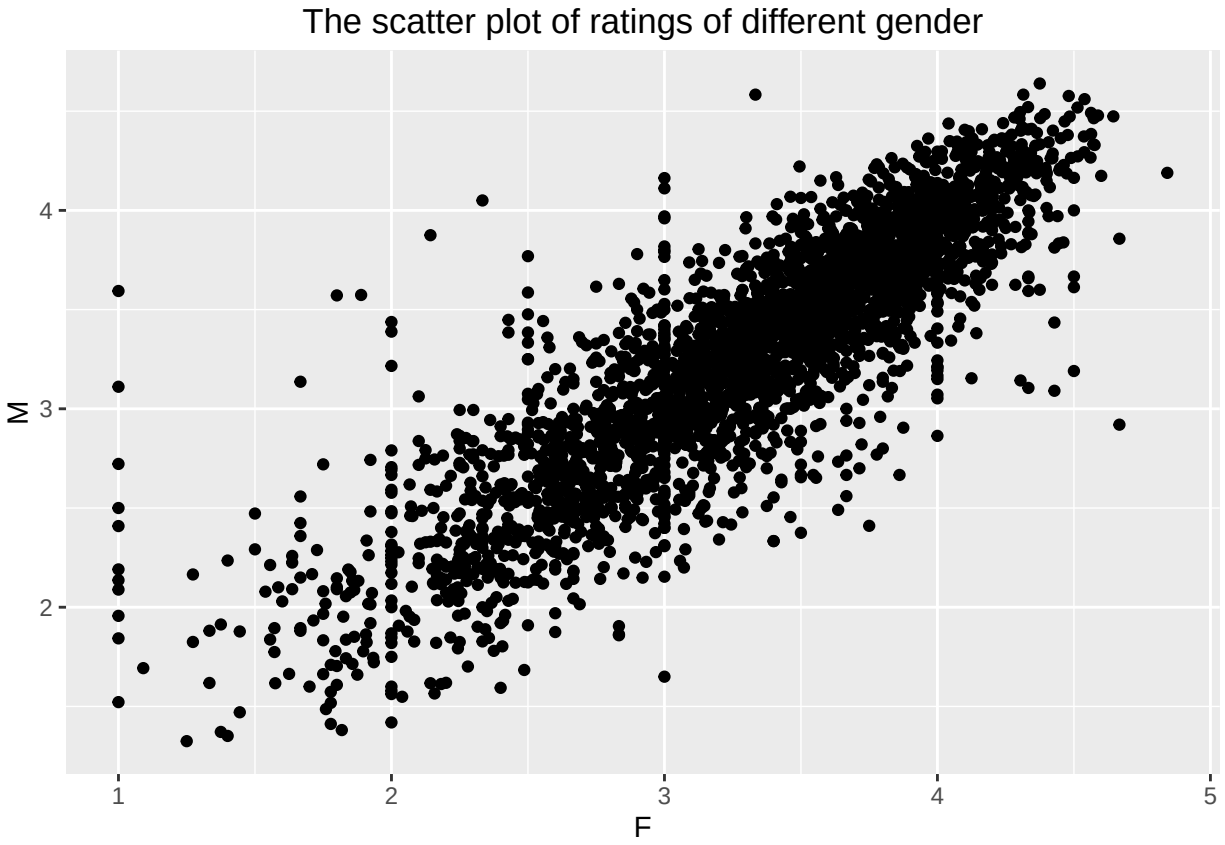


From the distribution map, we learn that users whose age between 25 and 34 years old account for the majority, and users whose age under 18 years old account for the least proportion, indicating that young people are more involved in online movie reviews. When it comes to occupational distribution, college/graduate students

account for the majority of user groups. In addition, the eductors and writers also account for a certain proportion. But to our surprise, people who engage in some busy jobs, such as programer, technician/engineer and executives/managers, also spend some time on online movie reviews.

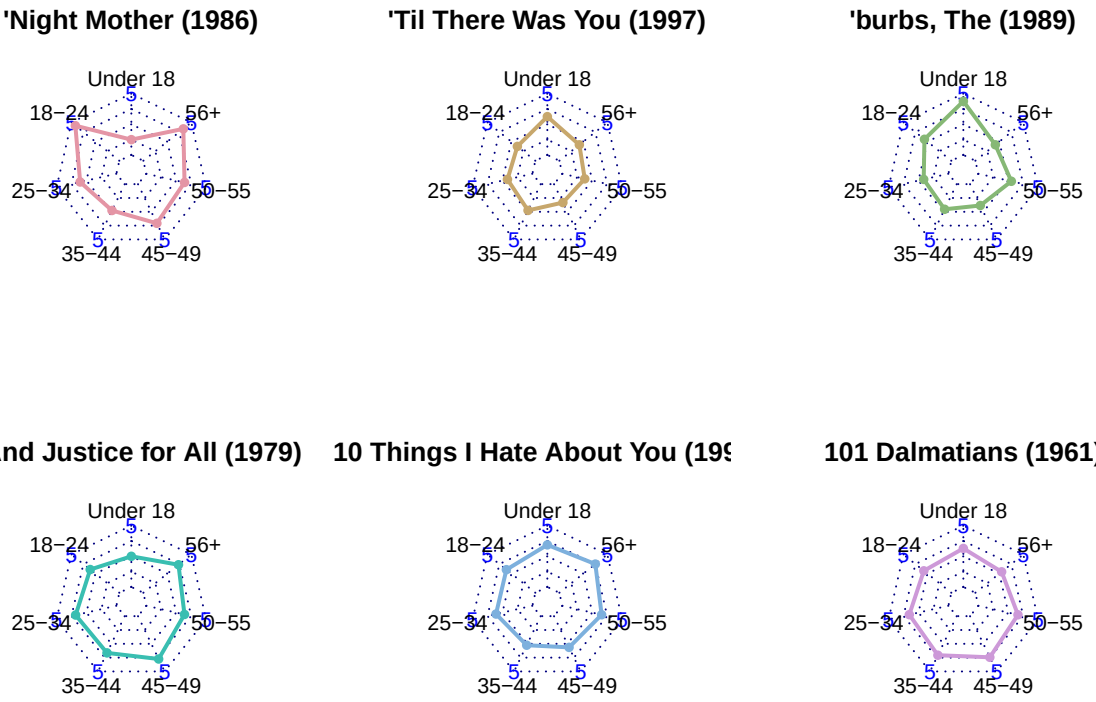
7.2.2 User Rating Analysis

Here I want to analyze the scores of different users in different gender. I calculate the average score of each movie which rated by female and the average score which rated by male.



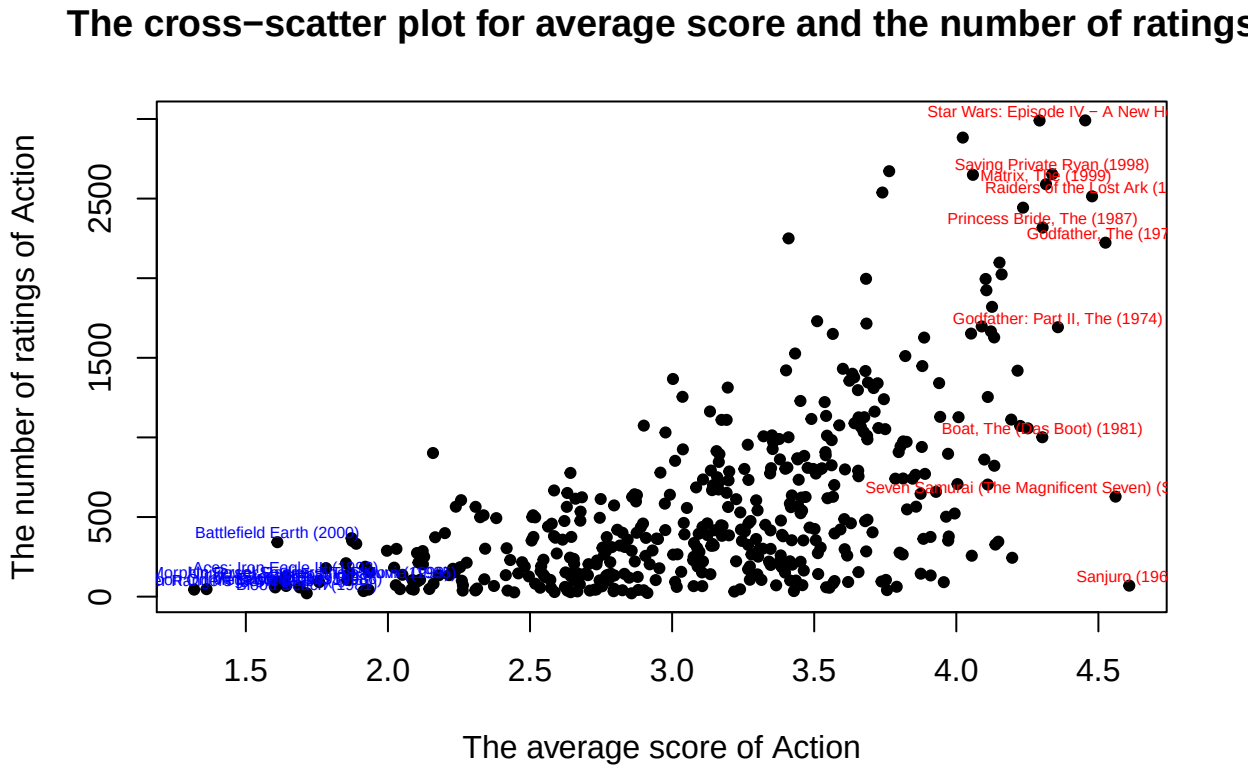
It can be seen from the scatter plot, the distribution of ratings from users of different gender is roughly in line with the function y = x, indicating that the average score of the same movie from users in different gender is roughly the same, only a few average score of movies exist some differences. For example, in the left part of the scatter plot, the men score high on the movies, while women score very low.

In order to understand the score from users in different ages, I use radarchart to visualize the result. I randomly select six movies to do visualization.



From the radarcharts, we can clearly see that different age groups give the same movie different scores. For "Night mother", users whose age is between 18 and 24 and users whose age is greater than 56 give it a higher score, while users whose age is under 18 years old give it the lowest score.

In order to view popular movies and boring movies more intuitively in a certain category of movies, I plot cross-scatter plots and add labels to the graph.



From the scatter plot, we can clearly see the distribution of Action movies. In the top right corner of the

map, the top 10 popular action movies are marked, and in the bottom left of the chart, the top 10 boring action movies are marked.

8 Recommendation System

8.1 Introduction

After the data analysis, we have a more complete understanding of the data set. In this section, I would like to use the entire data set to build a movie recommendation system, using object-based collaborative filtering algorithm. In the R language, the library "Recommenderlab" can achieve this function.

8.2 Modeling

In general, the two common algorithms used in recommendation system include user-based collaborative filtering and item-based collaborative filtering.The user-based filtering will probably work well for a few thousand people or items, but a very large site like Amazon has millions of customers and products—comparing a user with every other user and then comparing every product each user has rated can be very slow. Also, a site that sells millions of products may have very little overlap between people, which can make it difficult to decide which people are similar. However, in cases with very large datasets, item-based collaborative filtering can give better results, and it allows many of the calculations to be performed in advance so that a user needing recommendations can get them more quickly. In conclusion, in order to get a better result, I would like to build the model based on item-based collaborative filtering.

```
#Import library
library(registry)
library(reshape)
library(recommenderlab)
#Convert long to wide
ratings_long <- ratings[,c(1,2,3)]
ratings_wide <- cast(ratings_long, userid ~ movieid, value = "rating")
#convert format
class(ratings_wide) <- "data.frame"
useritem <- as.matrix(ratings_wide)
#convert into realRatingMaxtrix
rating_matrix <- as(useritem, "realRatingMatrix")
#change the column name
colnames(rating_matrix) <- paste("movie", 1:3707, sep = "")
```

Then I use the function Recommender to create model, "method = 'IBCF' " means that recommender algorithm that I use is item-based collaborative filtering.there are total 6040 users in the dataset, so I use the first 6,020 users' records to train the model, and do recommendations for last 10 users.

```
recommend_model <- Recommender(rating_matrix[1:6020], method = "IBCF")
```

8.3 Prediction

After modeling, I can do prediction based on the model. The prediction method mainly includes top-n prediction and rating prediction. Amazon former scientist Greg Linden has published an essay in 2009, this article pointed out that the purpose of movie recommendation is to dig the movies that users are interested in, rather than predict the rating that user would like to assign. Therefore, TopN recommendations are more

in line with the actual application requirements. There may be a movie that user would like to give a high score after watching, but it has a low possibility for the user to watch.

```
library(stringr)
predict1 <- predict(recommend_model, rating_matrix[6040], n = 5)
predict_result <- as(predict1, 'list')[1]
#Extract the movieid
recom <- str_extract_all(predict_result, "[0-9]+")[[1]]
recommendations <- data.frame(movieid = recom)
```

Using the 6040th user as an example, we can see that the recommendation results are as follows:

```
merge(movies, recommendations, by = 'movieid')
```

```
##   movieid                                        title
## 1      60         Indian in the Cupboard, The (1995)
## 2     106 Nobody Loves Me (Keiner liebt mich) (1994)
## 3     116               Anne Frank Remembered (1995)
## 4     121            Boys of St. Vincent, The (1993)
## 5     130                              Angela (1995)
##                          genres year
## 1 Adventure|Children's|Fantasy 1995
## 2                 Comedy|Drama 1994
## 3                  Documentary 1995
## 4                        Drama 1993
## 5                        Drama 1995
```

8.4 Model Assessment

In addition to item-based collaborative filtering, the recommendation algorithm also includes popularity-based recommendation and user-based collaborative filtering algorithm. This part will evaluate these three recommendation algorithms by using the results of score prediction.I divide the dataset into training set and test set according to 9: 1 ratio.

```
##               RMSE        MAE
## POPULAR 143.373210 38.3385846
## UBCF    187.896155 27.5916507
## IBCF      1.210951  0.8752066
```

In model assessment, RMSE(Root Mean Squared Error) and MAE(Mean Absolute Error) are used as the measure of estimated error. From the results, we can see that the prediction error of IBCF(item-based collaborative filtering) is the least, confirming what I have mentioned in the "Modeling" part.

9 Report Summary

In terms of the content, this report is mainly divided into two aspects, including data analysis and recommendation system. For the first part, I did some statistics on the datasets and visualized the results. And for the other part, recommendation systems are built based on the library "recommenderlab" to recommend movies to users. On the technical side, in addition to R language, MySQL and Python are also used to aid analyzing data. Because R cannot read ".dat" file directly, so MySQL is used to store the datasets. Besides,

the memory management mechanism of R is poor, resulting that it is very slow to process big data, but Python can perform better. At the beginning, I decided to use the full-size datasets(more than 20 million pieces of data) to do the analysis, but it is too hard for R to analyze such large datasets. As a result, smaller datasets (more than 1 million pieces of data) are used in this report. Although R is poor in dealing with bid data, but its functions of data visualization are very powerful, so that R and Python can be combined for data analysis in the future.

The detail description of datasets and the MySQL data operation statements are all posted to my GitHub [1].

[1]Statement: All the contents and ideas of this report are original, the relevant information has been posted to GitHub (Nelson ZHao).