Report of the Movielens Project

071970225 ZhouKang

2022-11-12

1 Introducion

1.1 The information about Dataset Movielens

The Movielens dataset contains the ratings users gave to different movies.First of all let me acquaint you with the overall contents of this 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
## 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
```

There are six columns in the “movielens” dataset.Notice that he release year is contained in the title column.The column timestamp is the time of users gave their ratings,and is stored as seconds it happened after the standard start time which is January 1,1970.In the analysis the timestamp will be transformed into weeks to diminish the time point.

To check some basic information of the table.

```
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 :35741   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
##
##
##
```

1.2 The goal of the project

The movielens dataset came from a campaign launched by Netflix to improve its recommendation system.Our goal is to predict the ratings give by a user on a specific movie.The dataset will be split onto two partition:the "edx" training set and the "temp" testing set.An important rule for the prediction is avoiding using the data in the testing set in any circumstance to prevent the overfitting or overtraining.

2 Methods and Analysis

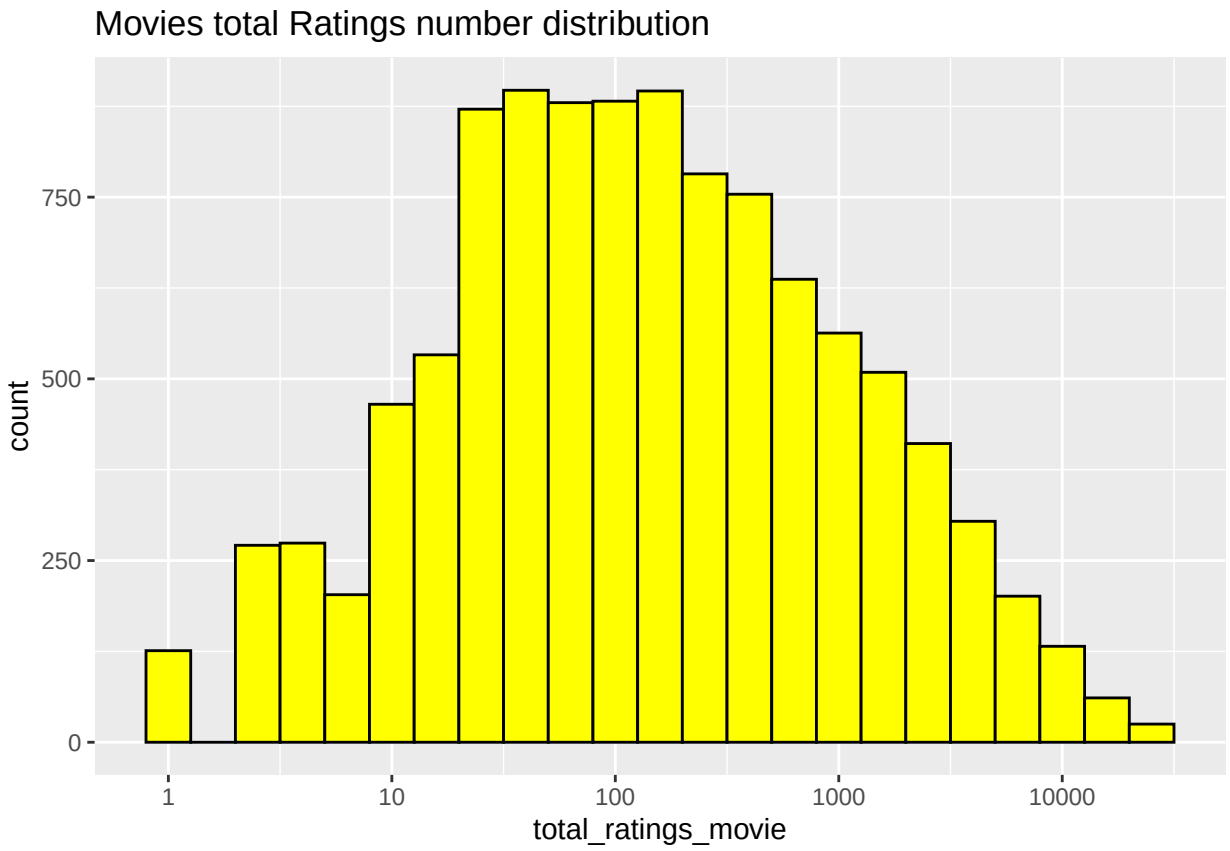
2.1 Methdology

The mean of all ratings is computed and function as the basic value of the ratings.We assume that all ratings are around this value but are influenced by other factor for every specific rating.

2.2 Visualization

Visualization is a good way to search for some relations between the ratings and predictors.

2.2.1 The distribution of the total number of ratings of each movie.



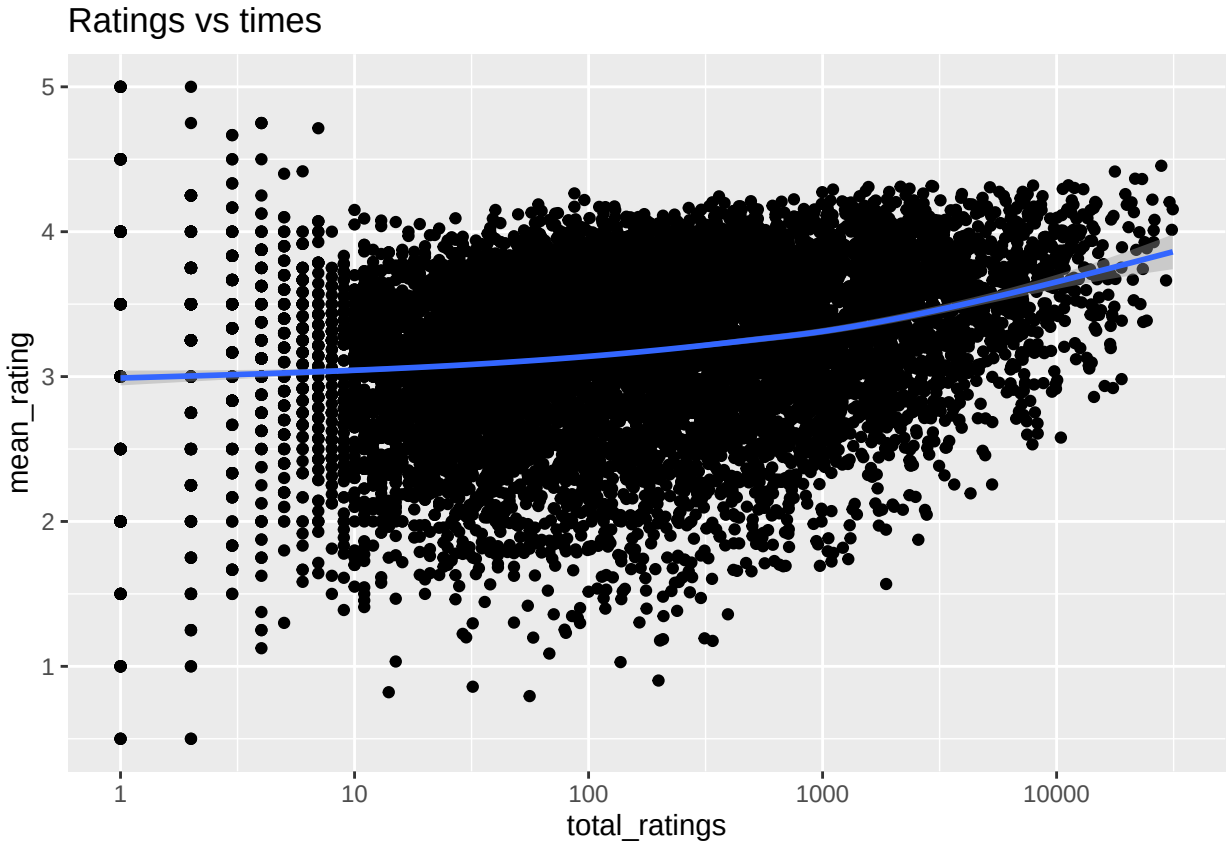
The range of n is very huge so we modify the x scale into the norm of log10(x) so the image will not be too long.Due to the deformation of the x scale,it is not a normal distribution as the image shows though the image seems like normal distribution.

The quantile of the total ratings is as below:

```
##      0%     10%     20%     30%     40%     50%     60%     70%     80%     90% 
##     1.0    10.0    23.0    40.0    70.0   122.0   210.0   397.0   832.8  2150.2 
```

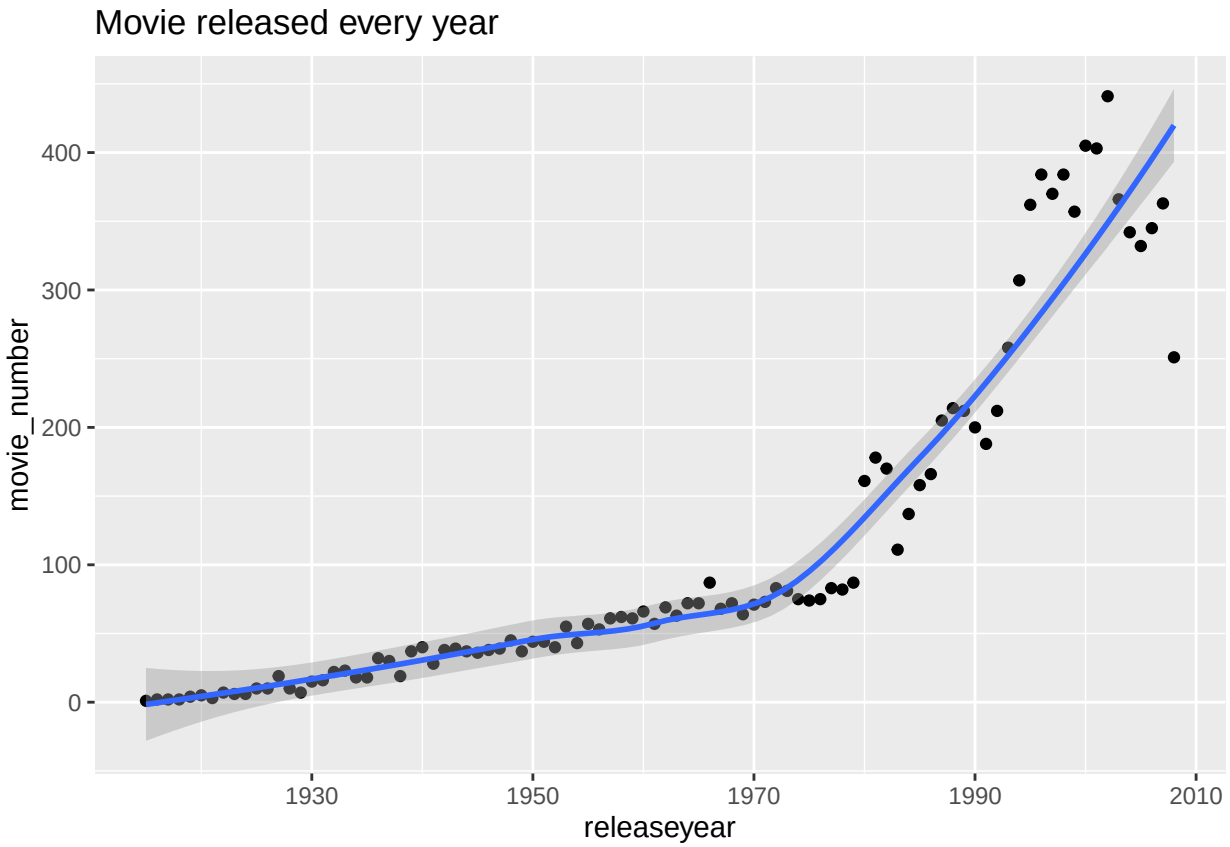
```
##    100%
## 31362.0
```

2.2.2 The mean ratings of a movie versus the times it was rated



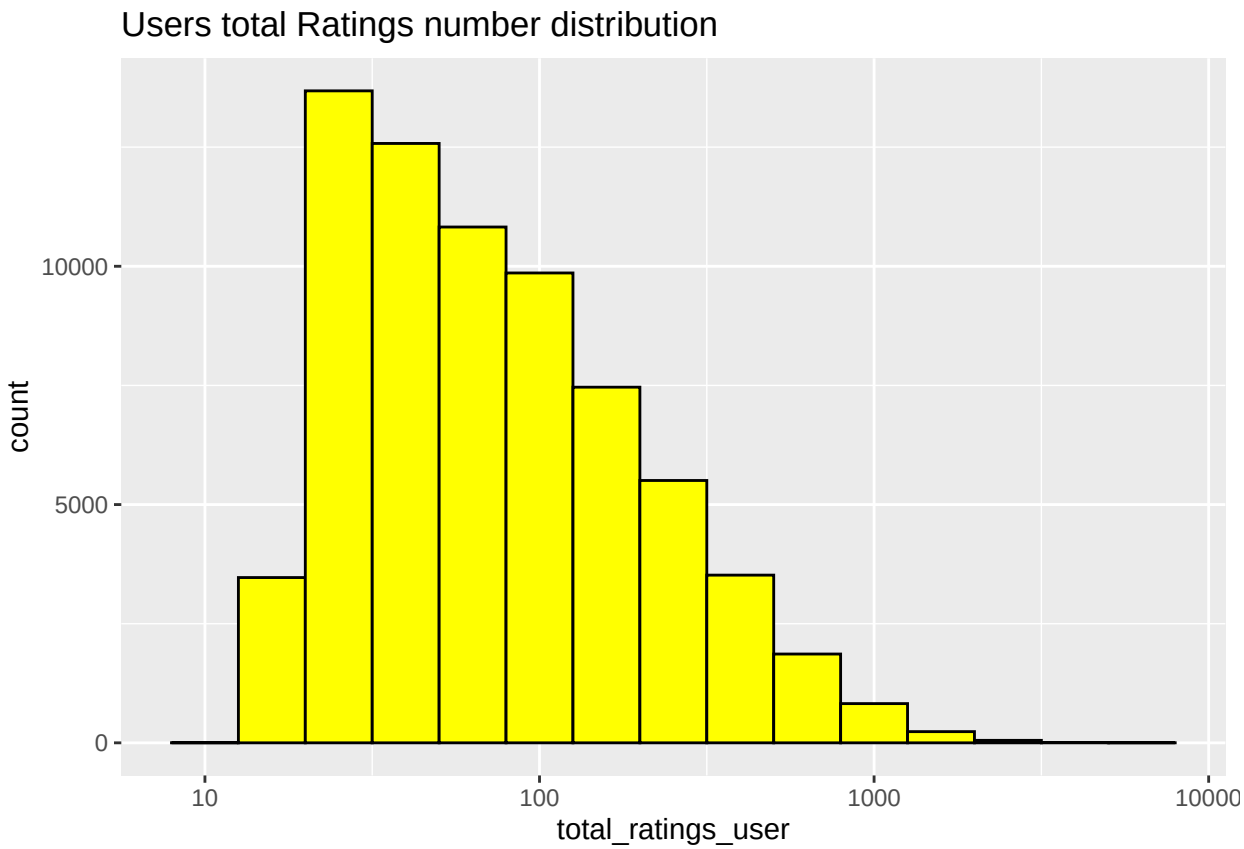
We can see that more ratings indicate a higher and more stable score.Regression should be employed to modify the movies that were quite obscure and got less ratings.

2.2.3 The number of movies released every year



We can see a sudden increase in movie number in around 1980s.

2.2.4 The distribution of the total number of ratings of each user.

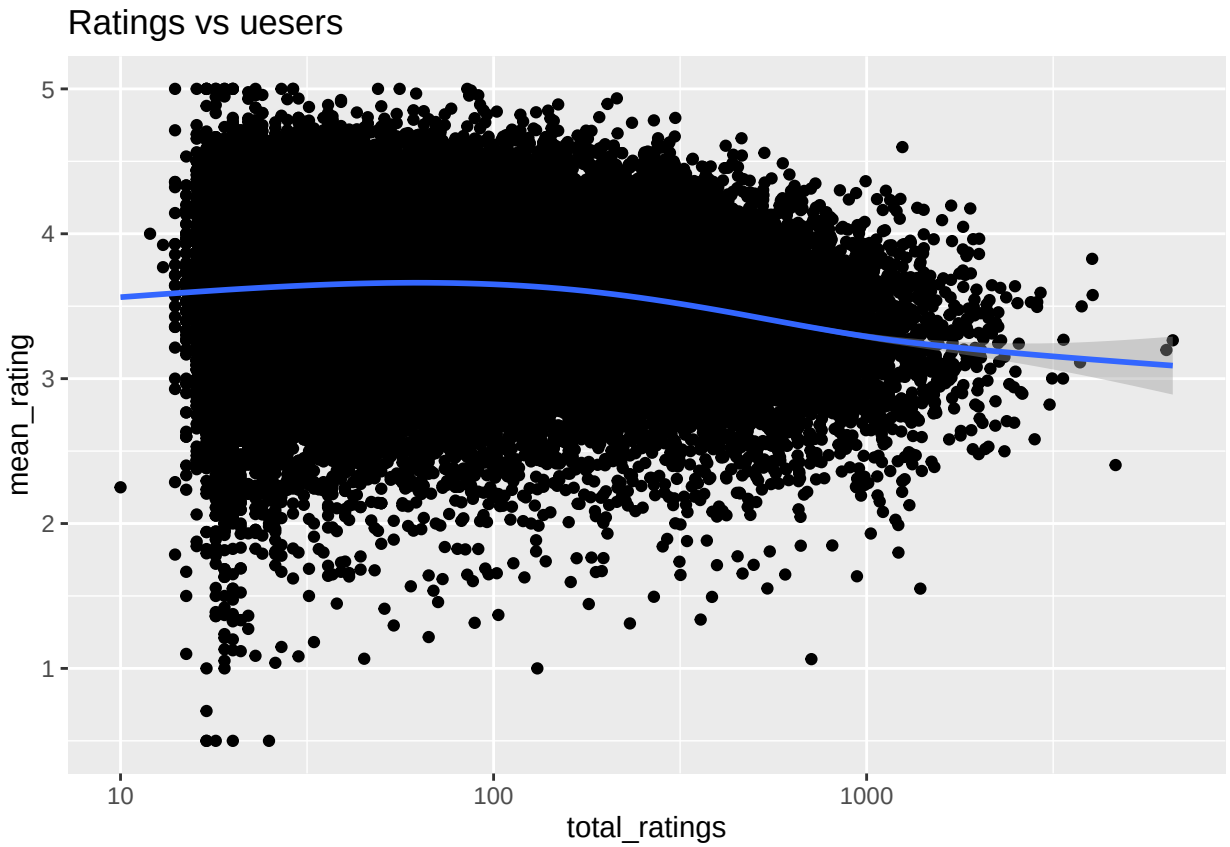


It is obvious that most users rated movies less than 1000.The quantile of the total ratings of every user is as below:

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%  90% 100%
##   10   22   28   36   47   62   85  116  176  301 6616
```

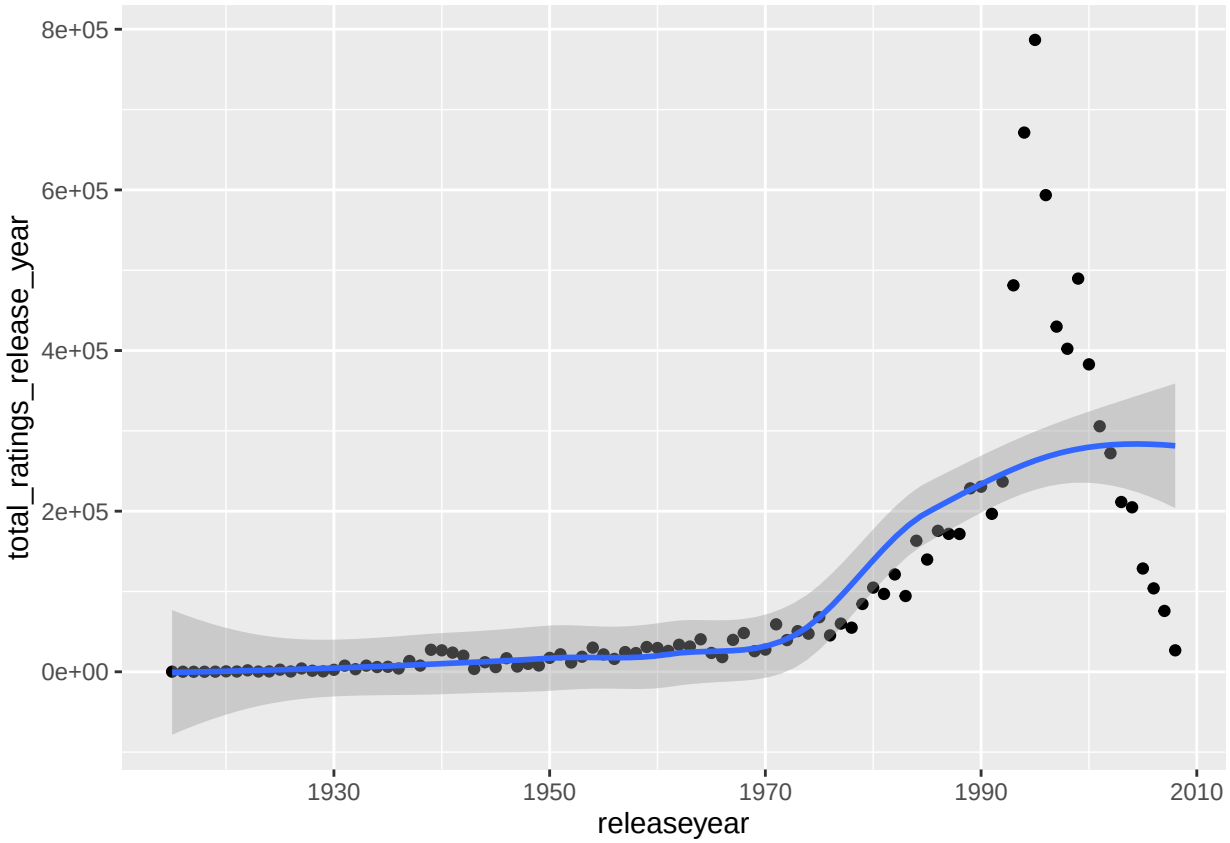
According to the result we know that 90% users rated less than 301 movies.

2.2.5 The mean ratings of a user gave to movies



The condition is quite similar with the total number a movie was rated.Users who made more ratings are much more stable in their mean ratings given to movies.

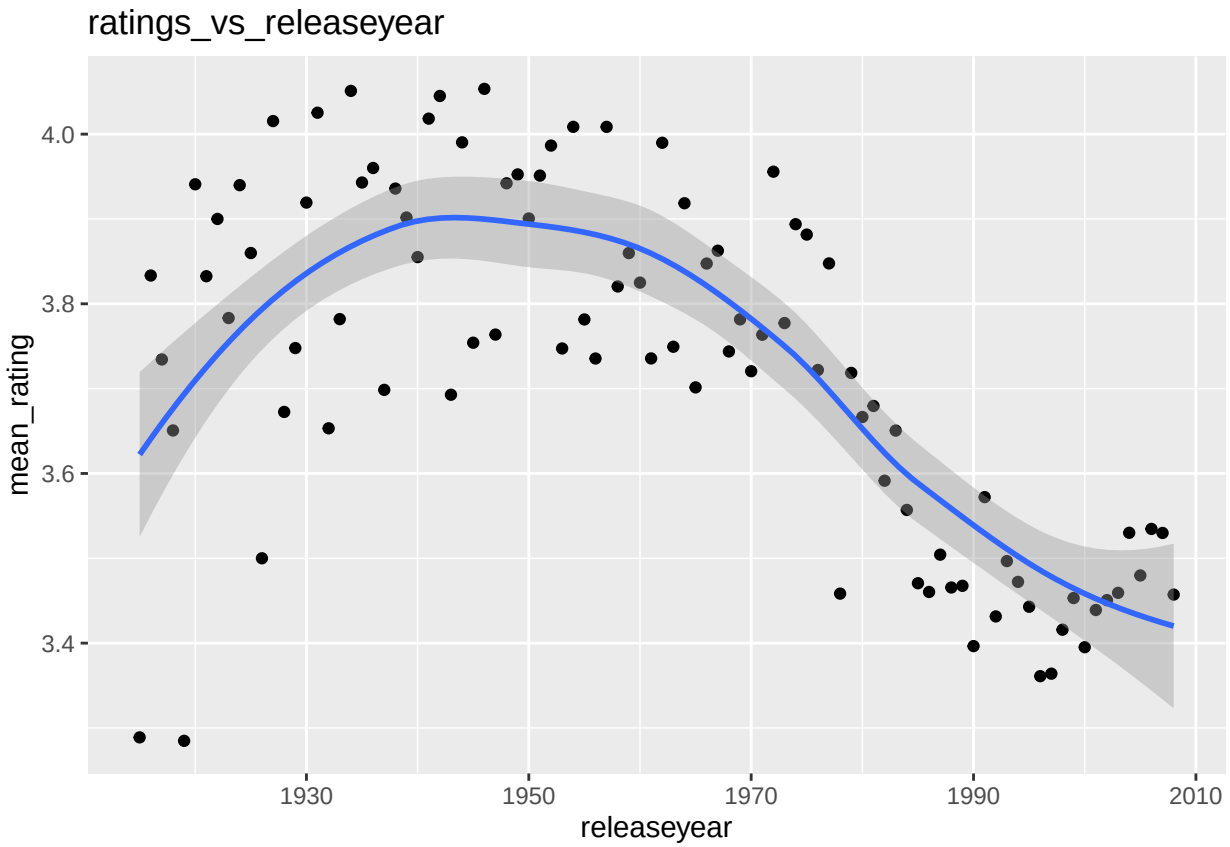
2.2.6 The distribution of the total number of ratings of movies released in every year.




```
## $title
## [1] "Total ratings vs release year"
##
## attr(,"class")
## [1] "labels"
```

Compare this with the the movie released every year.More movies were released after 1990s,but the rating numbers conversely was decreasing.

2.2.7 The mean ratings of movies released in different year.



Obviously there is some relation between the mean ratings of a movie and the year it was released.

2.2.8 The relation between the ratings and movie genre.

Most movies belong to more than just one genre and makes it difficult to split them into different types. So we will not use it for the prediction.

To see first 10 movie's genre and all genres:

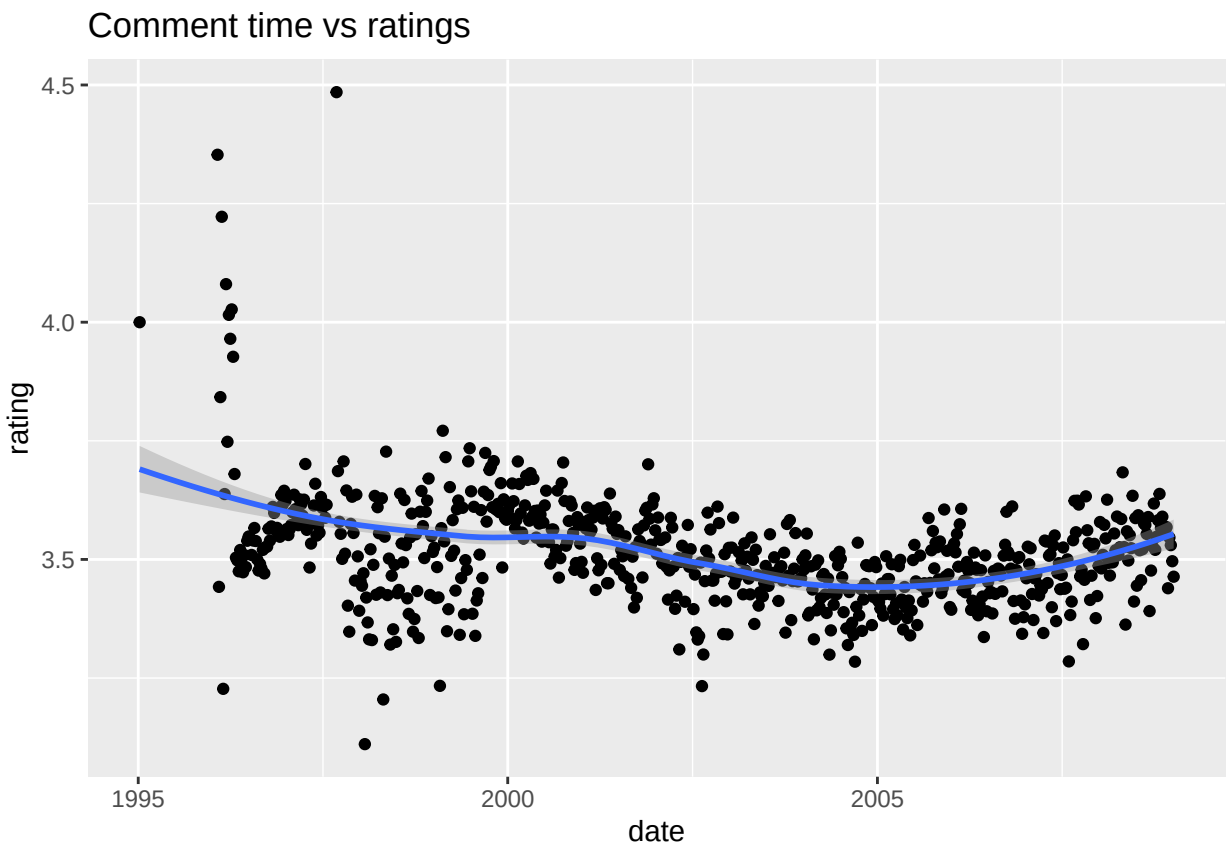
```
##  [1] "Comedy|Romance"
##  [2] "Action|Crime|Thriller"
##  [3] "Action|Drama|Sci-Fi|Thriller"
##  [4] "Action|Adventure|Sci-Fi"
##  [5] "Action|Adventure|Drama|Sci-Fi"
##  [6] "Children|Comedy|Fantasy"
##  [7] "Comedy|Drama|Romance|War"
##  [8] "Adventure|Children|Romance"
##  [9] "Adventure|Animation|Children|Drama|Musical"
## [10] "Action|Comedy"
```

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

It is quite complicated and contains too many genres.

2.2.9 The relation between the comment time and ratings.

```
edx%<>%mutate(date = as_datetime(timestamp))
edx%>%mutate(date=round_date(date, unit = "week"))%>%
  group_by(date) %>%
  summarize(rating = mean(rating)) %>%
  ggplot(aes(date, rating)) +
  geom_point() +
  geom_smooth()+ggtitle("Comment time vs ratings")
```



The connection between the time and ratings is not strong.

2.2.10 Summary of the predictors

Due to the visualization of different factors and consider the significant connection.The predictors include movie ID,user Id and the release year.

3 The build of the model

3.1 The prediciton of ratings.

We assume that all the movies has the same basic ratings and is influenced by the chosen predictors.And the ratings are given by the formula: ratings=basic ratings + bias_movie+bias_user+bias_release year.

For the estimate of the model,we build the RMSE to compute the distance between the estimated ratings and the true ratings.

To compute basic ratings of all movies

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

Now we can check the RMSE for the first time to see our precision.Define the function RMSE as below:

```
RMSE<-function(pred_value,true_value)
{sqrt(mean((pred_value-true_value)^2))
}
```

And the RMSE for the basic ratings and true ratings is listed:

```
RMSE(mu,edx$rating)
```

```
## [1] 1.060331
```

3.2 The bias of movie ID with regularization

First try to narrow the scope of alpha for regularization.As the start,limit the alpha to start at 0 and increase by 10 until 100.

```
alpha<-seq(0,100,10)
bias_pre<-edx%>%group_by(movieId)%>%summarise(sum=sum(rating-mu),n=n())
Regular<- sapply(alpha,function(alpha)
  {
  bias<- bias_pre%>%left_join(edx,.,by = "movieId")%>%
  mutate(bias=sum/(n+alpha))%>%
  mutate(pred=mu+bias)
  return(RMSE(bias$pred,edx$rating))
  })
```

The best alpha is 0, meaning no regularization.

```
alpha[which.min(Regular)]
```

```
## [1] 0
```

```
min(Regular)
```

```
## [1] 0.9423475
```

Then narrow the scope of alpha step by step.And the result showed that alpha equals to 0.

Now we see that alpha=0 is the best.So the bias_movie is certain now

```
alpha_movie<-0
bias_reg<-edx%>%group_by(movieId)%>%summarise(bias_movie=sum(rating-mu)/(n()+alpha_movie))%>%left_join(
###To see the first several lines and the RMSE
head(bias_reg)
```

```
##    userId movieId rating timestamp                  title
## 1:      1     122      5 838985046              Boomerang
## 2:      1     185      5 838983525               Net, The
## 3:      1     292      5 838983421               Outbreak
## 4:      1     316      5 838983392               Stargate
## 5:      1     329      5 838983392 Star Trek: Generations
## 6:      1     355      5 838984474       Flintstones, The
##                            genres releaseyear                date  bias_movie
## 1:                Comedy|Romance        1992 1996-08-02 11:24:06 -0.65387934
## 2:         Action|Crime|Thriller        1995 1996-08-02 10:58:45 -0.38313118
## 3:  Action|Drama|Sci-Fi|Thriller        1995 1996-08-02 10:57:01 -0.09445454
```

```
## 4:       Action|Adventure|Sci-Fi        1994 1996-08-02 10:56:32 -0.16278816
## 5: Action|Adventure|Drama|Sci-Fi        1994 1996-08-02 10:56:32 -0.17500816
## 6:       Children|Comedy|Fantasy        1994 1996-08-02 11:14:34 -1.02467799
##        pred
## 1: 2.858586
## 2: 3.129334
## 3: 3.418011
## 4: 3.349677
## 5: 3.337457
## 6: 2.487787
```

```
RMSE(bias_reg$pred,edx$rating)
```

```
## [1] 0.9423475
```

We can see an improvement in RMSE comparing with the basic ratings.

3.3 The bias of user with regularization

On the basis of the bias of movie and in similar way we find the bias of user.The best alpha is still 0.Then we can get the bias of users and RMSE get smaller.

```
alpha_user<-0
bias_reg<-bias_reg%>%group_by(userId)%>%summarise(bias_user=sum(rating-pred)/(n()+alpha_user))%>%
  left_join(bias_reg,.,by="userId")%>%mutate(pred=pred+bias_user)
head(bias_reg)
```

```
##    userId movieId rating timestamp                  title
## 1:      1     122      5 838985046              Boomerang
## 2:      1     185      5 838983525               Net, The
## 3:      1     292      5 838983421               Outbreak
## 4:      1     316      5 838983392               Stargate
## 5:      1     329      5 838983392 Star Trek: Generations
## 6:      1     355      5 838984474       Flintstones, The
##                            genres releaseyear                date  bias_movie
## 1:                Comedy|Romance        1992 1996-08-02 11:24:06 -0.65387934
## 2:         Action|Crime|Thriller        1995 1996-08-02 10:58:45 -0.38313118
## 3:  Action|Drama|Sci-Fi|Thriller        1995 1996-08-02 10:57:01 -0.09445454
## 4:       Action|Adventure|Sci-Fi        1994 1996-08-02 10:56:32 -0.16278816
## 5: Action|Adventure|Drama|Sci-Fi        1994 1996-08-02 10:56:32 -0.17500816
## 6:       Children|Comedy|Fantasy        1994 1996-08-02 11:14:34 -1.02467799
##        pred bias_user
## 1: 4.537821  1.679235
## 2: 4.808569  1.679235
## 3: 5.097245  1.679235
## 4: 5.028912  1.679235
## 5: 5.016692  1.679235
## 6: 4.167022  1.679235
```

```
RMSE(bias_reg$pred,bias_reg$rating)
```

```
## [1] 0.8567039
```

3.4 The bias of release year with regularization

Under the consideration of the bias of movie and the bias of user,the alpha of release year is still 0.Then we get the bias of release year include and made another progress in reducing the RMSE.Until now all these

modeling were based on the training set edx and the testing set remain untouched.

```
alpha_releaseyear<-0
bias_reg<-bias_reg%>%group_by(releaseyear)%>%
  summarise(bias_releaseyear=sum(rating-pred)/(n()+alpha_releaseyear))%>%
  left_join(bias_reg,.,by="releaseyear")%>%mutate(pred=pred+bias_releaseyear)
head(bias_reg)
```

```
##    userId movieId rating timestamp                  title
## 1:      1     122      5 838985046              Boomerang
## 2:      1     185      5 838983525               Net, The
## 3:      1     292      5 838983421               Outbreak
## 4:      1     316      5 838983392               Stargate
## 5:      1     329      5 838983392 Star Trek: Generations
## 6:      1     355      5 838984474       Flintstones, The
##                            genres releaseyear                date  bias_movie
## 1:                Comedy|Romance        1992 1996-08-02 11:24:06 -0.65387934
## 2:         Action|Crime|Thriller        1995 1996-08-02 10:58:45 -0.38313118
## 3:  Action|Drama|Sci-Fi|Thriller        1995 1996-08-02 10:57:01 -0.09445454
## 4:       Action|Adventure|Sci-Fi        1994 1996-08-02 10:56:32 -0.16278816
## 5: Action|Adventure|Drama|Sci-Fi        1994 1996-08-02 10:56:32 -0.17500816
## 6:       Children|Comedy|Fantasy        1994 1996-08-02 11:14:34 -1.02467799
##        pred bias_user bias_releaseyear
## 1: 4.544804  1.679235      0.006983816
## 2: 4.770650  1.679235     -0.037918647
## 3: 5.059327  1.679235     -0.037918647
## 4: 4.996909  1.679235     -0.032003181
## 5: 4.984689  1.679235     -0.032003181
## 6: 4.135019  1.679235     -0.032003181
```

```
RMSE(bias_reg$pred,bias_reg$rating)
```

```
## [1] 0.8563777
```

4 Results and conclusion.

4.1 Test the model on the validation dataset temp.

We compute the prediction of ratings with the formula:pred = mu + bias_movie + bias_user + bias_releaseyear

First We compute mu+bias_movie

```
pred_temp_1<- bias_reg%>%group_by(movieId)%>%summarise(bias_movie=mean(bias_movie))%>%
            left_join(temp,.,by = "movieId")%>%mutate(pred=mu+bias_movie)
```

The RMSE is

```
RMSE(pred_temp_1$pred,temp$rating)
```

```
## [1] 0.9439087
```

Then we include the bias_user

```
pred_temp_2<-bias_reg%>%group_by(userId)%>%summarise(bias_user=mean(bias_user))%>%
            left_join(pred_temp_1,.,by = "userId")%>%mutate(pred=pred+bias_user)
```

The RMSE reduces to

```
## [1] 0.8653488
```

At last it is the release year

```
pred_temp_3<-bias_reg%>%group_by(releaseyear)%>%
  summarise(bias_releaseyear=mean(bias_releaseyear))%>%
  left_join(pred_temp_2,.,by = "releaseyear")%>%mutate(pred=pred+bias_releaseyear)
```

The RMSE is smaller now

```
## [1] 0.8650043
```

After checking the range of the ratings,some unnormal values were transformed to their nearest acceptable value, The number smaller than 0 to 0 and that larger than 5 to 5.

```
range(pred_temp_3$pred)
```

```
## [1] -0.4937364  6.0493942
```

```
pred_temp<-pred_temp_3%>%mutate(pred=ifelse(pred<0,0,ifelse(pred>5,5,pred)))
```

Now we can check the final RMSE of the model.

```
RMSE(pred_temp$pred,temp$rating)
```

```
## [1] 0.8648403
```

The final RMSE is 0.8648403 and is quite satisfying and we find a acceptable estimation of the ratings. We can also have the accuracy of predict the exact correct rating.

```
mean(floor(pred_temp$pred)+ifelse(mod(pred_temp$pred,1)<0.25,0,ifelse(mod(pred_temp$pred,1)<0.75,0.5,1))
```

```
## [1] 0.2493792
```

The rate is around 0.25.We can compare this with simply guessing the mean ratings.

```
mu_temp<-mean(temp$rating)
mean(floor(mu_temp)+ifelse(mod(mu,1)<0.25,0,ifelse(mod(pred_temp$pred,1)<0.75,0.5,1))==temp$rating)
```

```
## [1] 0.08814009
```

Our prediction is obviously doing much better.

4.2 Conclusion

After modify the basic ratings step by step, the RMSE was reduced and a acceptable model was find.While some factors were not considered like the genres.Though the final RMSE

is in a good condition while the accuracy of prediction is not very high.

In addition,the regularization seems not to function in this project and all the alpha for three predictors are 0.