Movielens recommendation project

Judy Bowen

September 14, 2021

The purpose of this project is to use R to build a recommendation system that helps people find a movie to watch. The data is provided as a subset of movies rated by viewers, as generated by the GroupLens research lab[1]. The dataset for this project consists of 9,000,055 observations of six variables (userId, movieId, rating, timestamp, title and genres). Each row of data represents a rating given by one user to one movie. The movie recommendation system here was created using a model based upon the BellKor solution to the Netflix Grand Prize[2], [3]. The BellKor solution uses a number of techniques – linear regression models, factor analysis, gradient boosted decision trees. In this paper, I make use of the linear modelling techniques, where the regression model was defined as a loss function; the purpose of the model is to minimize the RMSE, where y_u,i is the rating for movie i by user u, with the movie prediction defined as yhat_u,i. The RMSE is defined as:

sqrt(mean((true_ratings - predicted_ratings)^2)).

In this paper the relationship between each movie by each user has the greatest impact on the model RMSE. The model here also incorporates the time effect, as described by the BellKor solution, where the time variable is binned into 10 week sections to take into account changes in tastes over time. Also included was a frequency variable which estimates the effect of rating a number of movies on a single timestamp – if many movies are reviewed at one time, the BellKor group hypothesized that the mood and tastes of a user at the one time would weight more heavily on the measures of movie preferences. Genres were also included in an attempt to relate tastes for a broader category of movies into the model.

The beta estimates obtained from the model were transformed by standardizing these estimates, as described in the BellKor paper, such that a penalty was imposed on movies with a low count of ratings. The rating and the frequency variable were transformed, prior to running any model, by scaling these to a z-distribution. It turned out, in my models, that scaling the ratings would yield the greatest improvement to the RMSE. Prior to scaling the ratings, I could not obtain an RMSE below 0.86, even with standardization applied to the beta estimates. The final RMSE, obtained from the validation set, was measured at 0.8155.

Dataset Description and Analysis

The following describes the dataset in general terms (using methods as described in Data Analysis and Prediction Algorithms with R[4]).
In general terms, the database is described as having the following unique users, movies, genres and timestamps:

Table 1: Count of unique values for each variable

n_users	n_movies	n_genres	n_timestamp
69878	10677	797	6519590

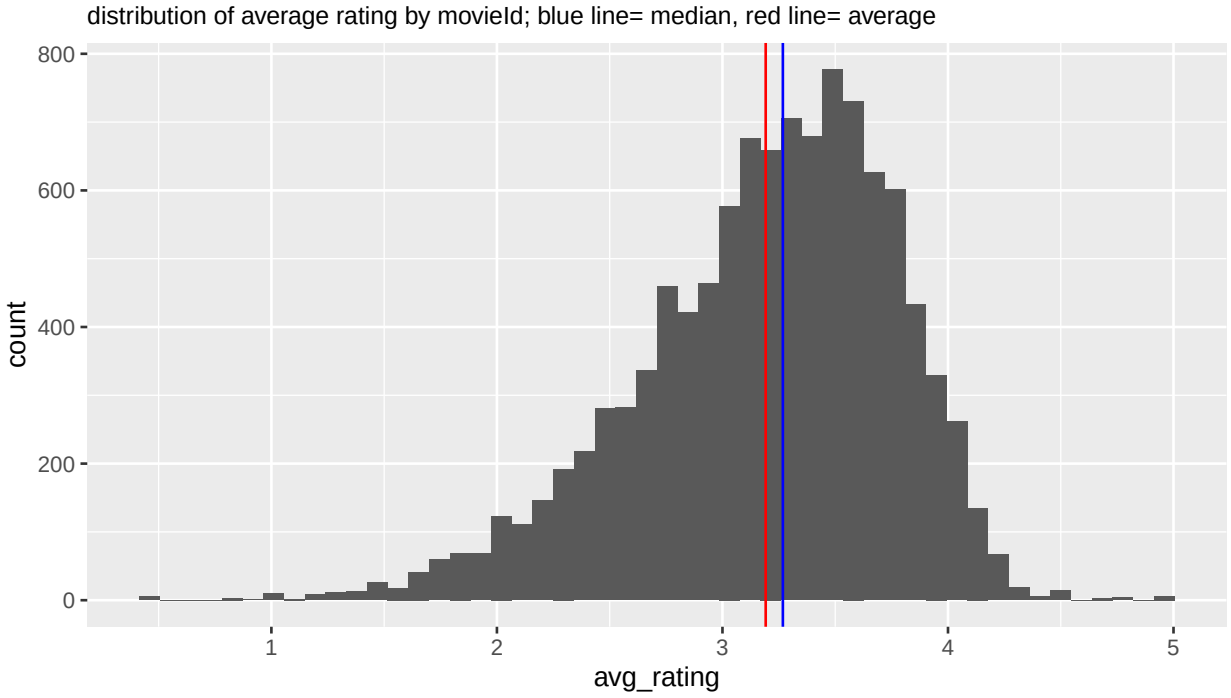
[1] https://grouplens.org/
[2] https://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf
[3] http://blog.echen.me/2011/10/24/winning-the-netflix-prize-a-summary/
[4] https://rafalab.github.io/dsbook/

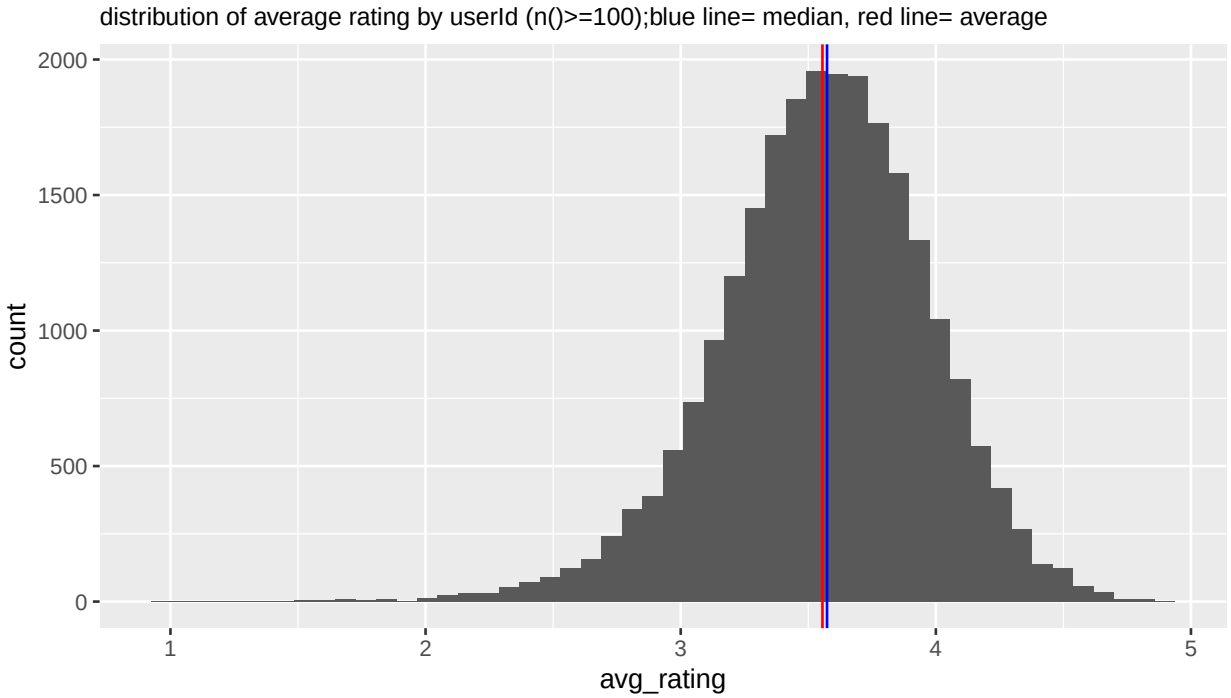
The mean of the distribution of ratings in the edx movielens database is 3.512465. The median is located to the right of the average rating, 4. As the mean of the distribution is less than the median, the distribution is negatively skewed. The measure of skewness in the edx data set is -0.5958884, also indicating that the distribution of the ratings is skewed towards the left.

Table 2: Mean, median and skewness measures of the rating variable

measure	value
mean	3.5124652
median	4.0000000
skewness	-0.5958884

The distribution of the average rating is illustrated in the histograms, below for average rating by movieId and then average rating by userId (for userId which have 100 or more ratings). The average rating by movieId and the median rating by movieId also illustrate evidence of negative skew in the dataset:





The definition of a user and a movie title (and associated movieId) are straightforward and do not need much further description to add to understanding these data points. However, the genre descriptors and the timestamps were investigated to determine what information they contribute to understanding the dataset.

The genres

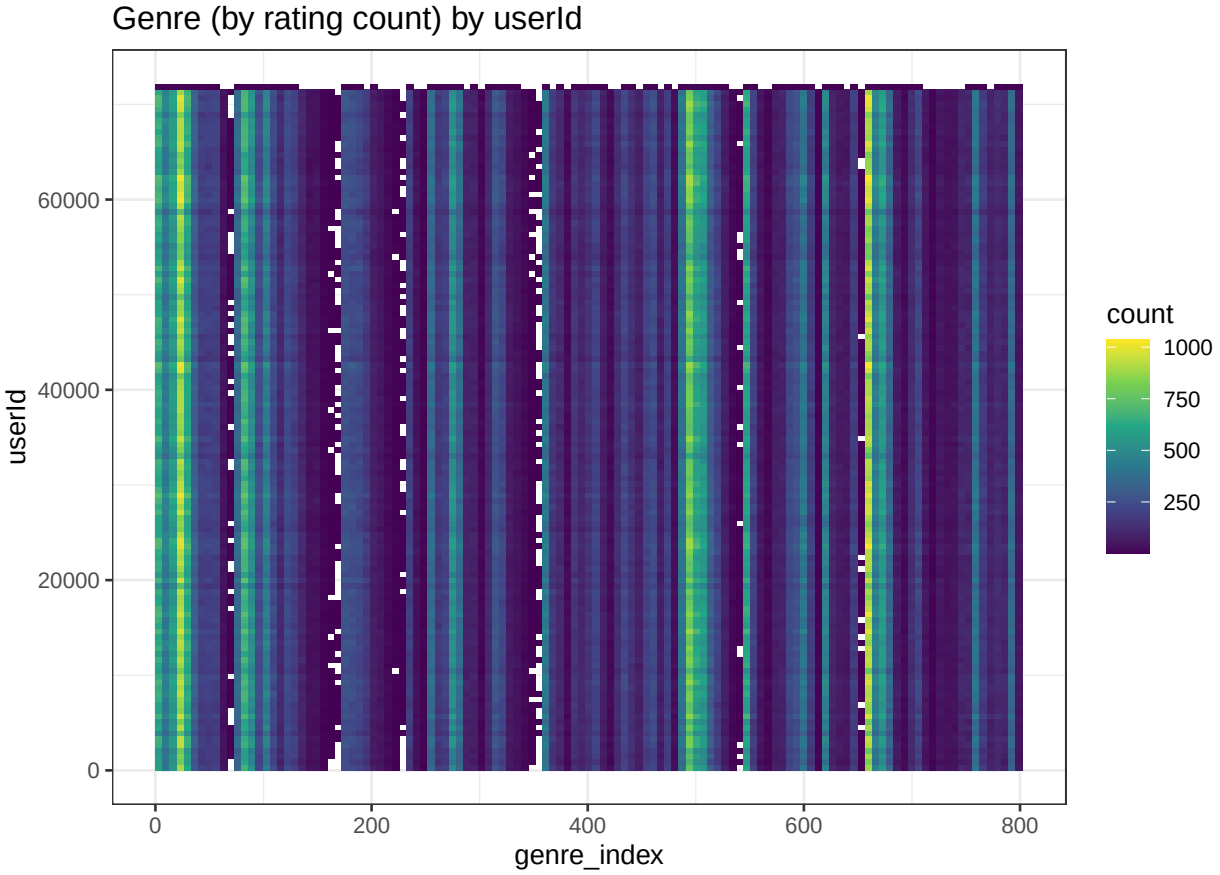
The genres are single or multiple word descriptors of the movie content. In all, there are 20 different genre descriptors, used as single word descriptors or combined in groups of up to 8 different descriptors. For example, Dead Poets Society (1989) is described as a Drama. However, many movies are described by more than one genre: Lion King, The (1994) is described by five genres (Adventure|Animation|Children|Drama|Musical). As described above, there are 797 unique genre word combinations (which includes seven movies with no genre description and one movie (with 256 ratings) with eight genres describing its content (Host, The (Gwoemul)(2006)). The top five genre descriptors are shown in the following table (note that this table does not tell us that the largest categories are rated highly for all movies included in the category; it merely tells us how many movies are described by the genre)(genre_count in this table represents the number of words in the genre description):

Table 3: Top 5 genres by rating count

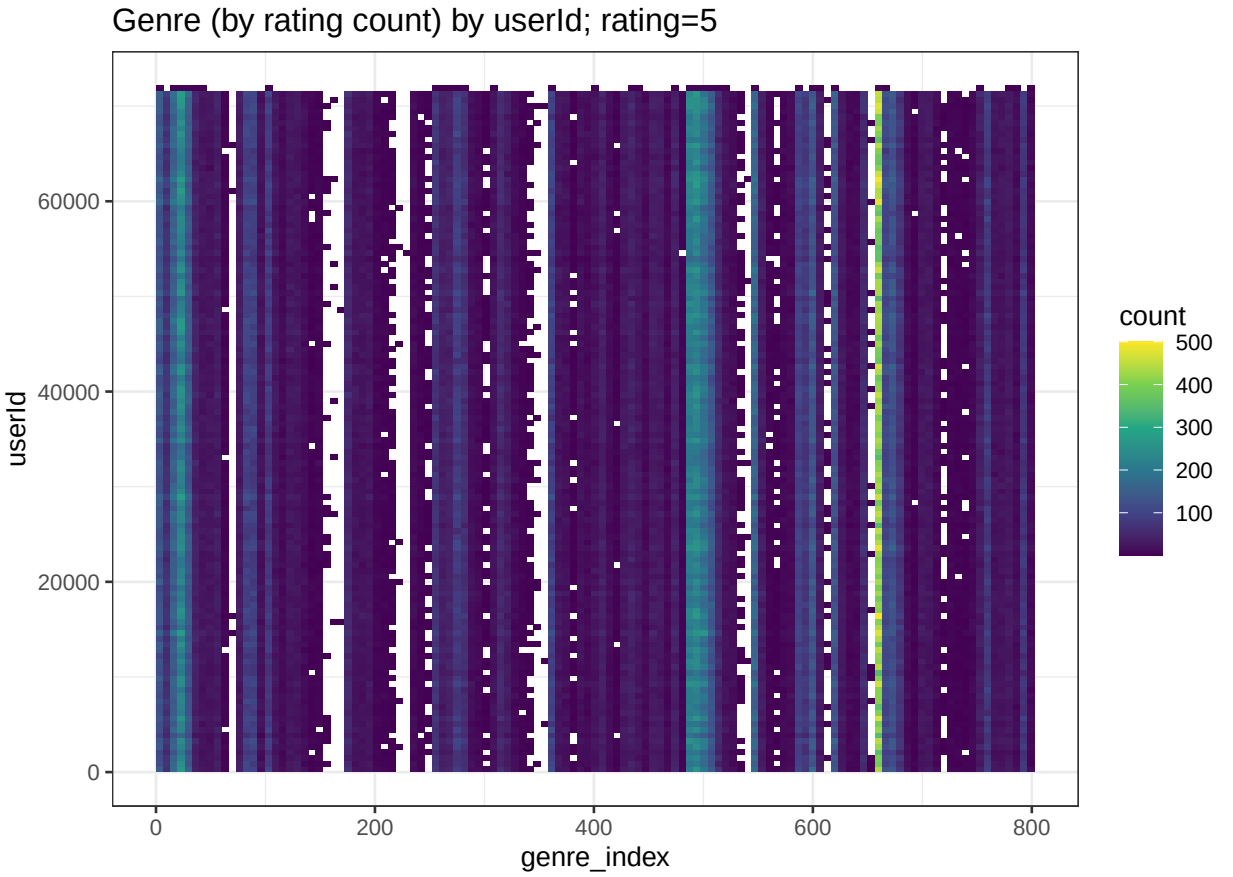
genres	count_ratings	count_descriptor
Drama	733296	1
Comedy	700889	1
Comedy|Romance	365468	2
Comedy|Drama	323637	2
Comedy|Drama|Romance	261425	3

To try to understand the distribution of the genres in general terms, a histogram was created that plotted the genres (by count) for each userId (for all userId with greater than 100 ratings). As there are 797 different genres, it was necessary to create an number index in order to create a histogram in readable form. The index number was assigned to the genre list in alphabetical order…so that genres beginning with Action were found in the lower numbers of the index and the genres beginning with War were at the end of the index.

The histogram reveals that some genres are almost never reviewed (the white space on the histogram has empty data; the dark blue indicates a low count of ratings). As indicated in the tables, there are only a few categories that stand out for preferred viewing. But this histogram does not indicate that these genres would necessarily be correlated to a good rating.



Attempting to visualize the histogram a 'good' rating, the same plot was created, but this time, only records with a rating equal to five were included in the visualization. This time, with the somewhat smaller sample size, it remains clear that some genres are preferred by all users and rated more highly and that these genres are also the genres with more ratings (and therefore, selected for viewing).



That higher counts of ratings seem to coincide with more favourable ratings is easy enough to explain: if the reviewers favour a genres they are more likely to select those genres for viewing so that review counts alone may reflect a bias in tastes.

The genre descriptors are in alphabetical order. That means that a movie described by Comedy|Drama|Romance is not mostly a Comedy any more than it is a Comedy with some Romance or primarily a Romance with Drama and a touch of Comedy. Some wordy genre descriptors are rarely observed: Action|Adventure|Animation|Comedy|Sci-Fi is applied to only one **not** well known movie (Dead Leaves (2004)) which has only three reviews (the reviews weren't bad…two 4.5 and one 3.0).

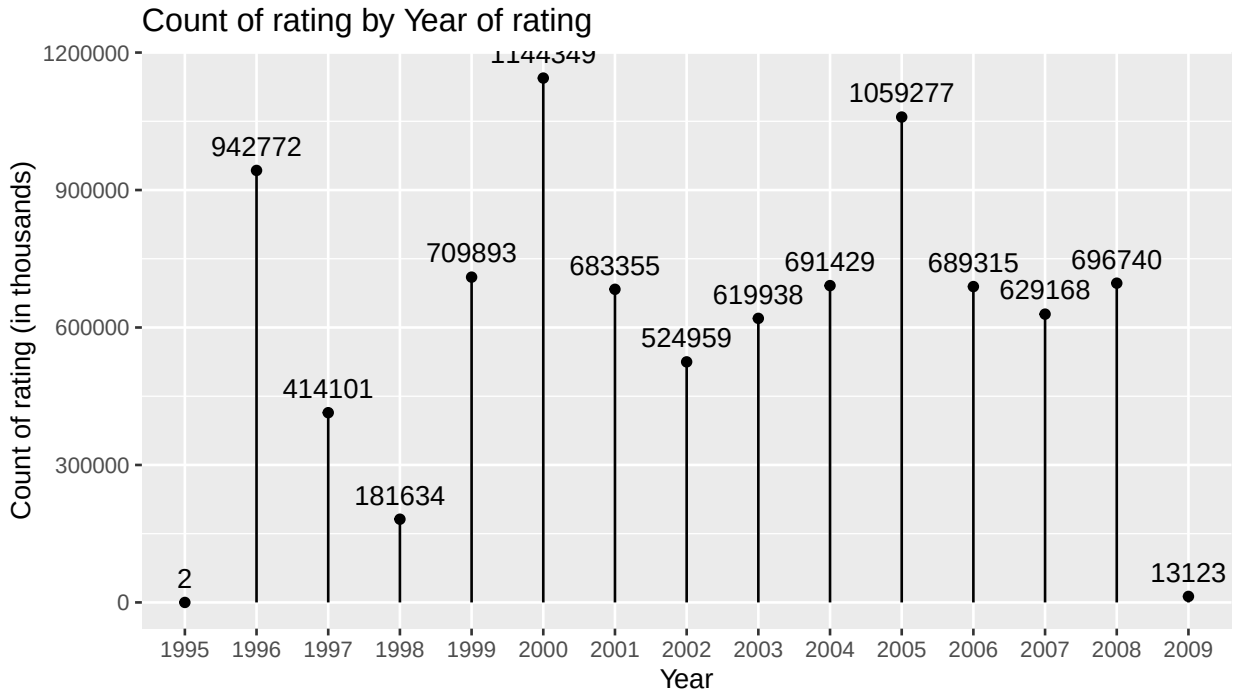
The matter of genres required a bit more research. So, I went to Netflix! Netflix has a number of movie genres so viewers can select a movie based on their genre preferences. A movie that falls under multiple genre descriptors will be found on a number of different genre menu selections. For example, the movie Sahara (2005) is found listed as a Comedy under the Action menu selection. It is also found as an Action&Adventure based on Books movie under the Comedies menu selection. (Good work Netflix movie recommendation programmers!). It seems that in a movie recommendation model, the genre descriptor would be more suited to nonlinear decision tree programming than to a linear least squares model. However, genre preference will be related to userId and may be a good rough indicator for movie recommendations even in a linear model.

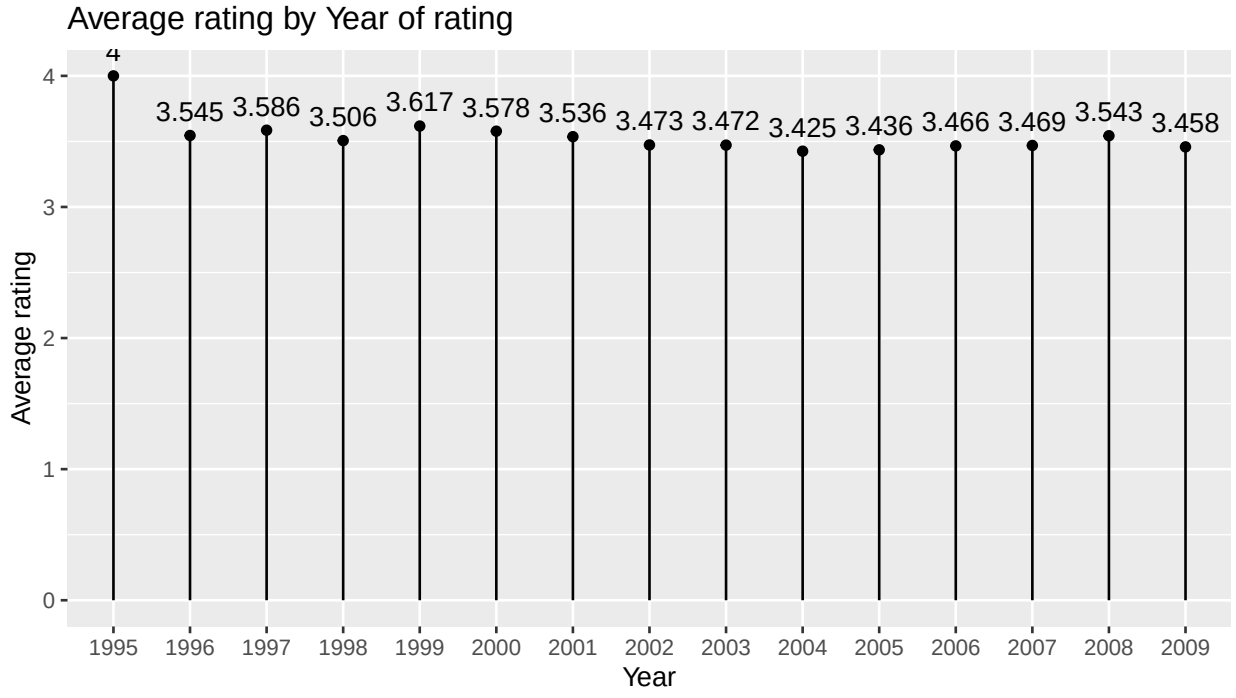
The Timestamp

As described above, there are a little over 6.5 million time stamps in the database. The raw data is presenting a timestamp which represents the time and date on which the rating was provided. The units are seconds since January 1, 1970. Investigation of this variable is easier if the timestamp were presented in a form more easily interpreted. Therefore, the timestamps were converted to year, month, week , day and hour of day, to see if there are patterns to be discovered which explain ratings.

Viewing the data of count of ratings by year reveals that the values in 1995 and 2009 are obvious outliers;

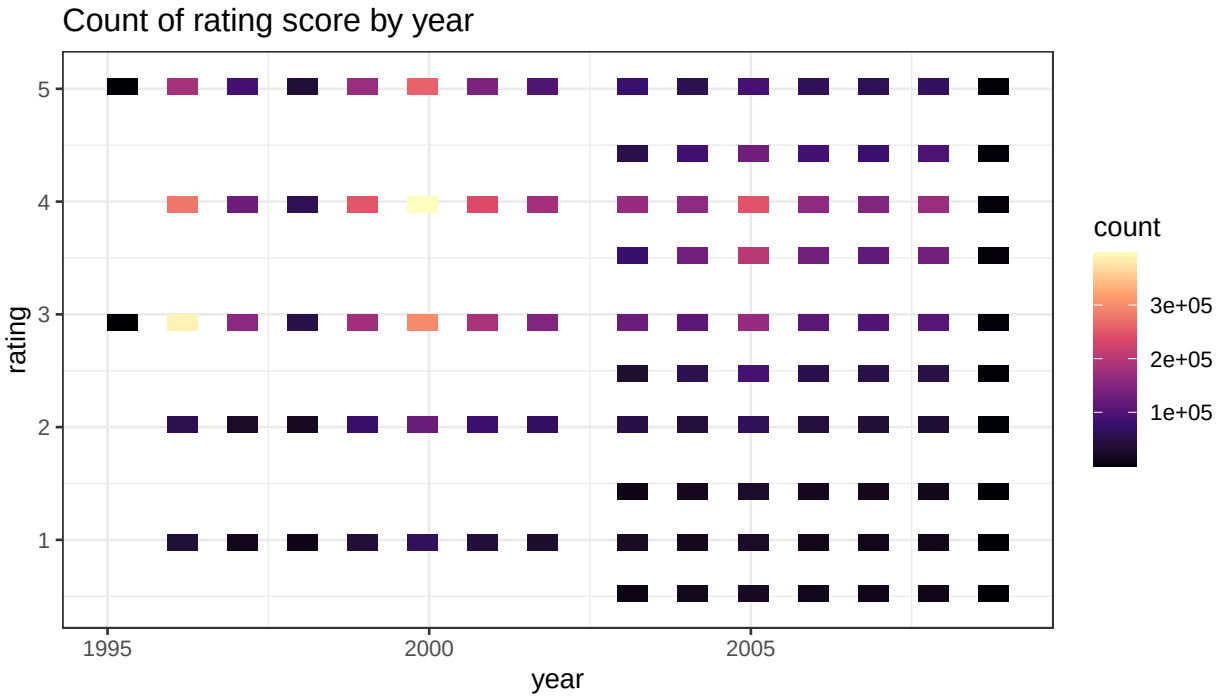
removing these years from the dataset shows more linear relationship between year and count of movie ratings but it is still pretty rocky.





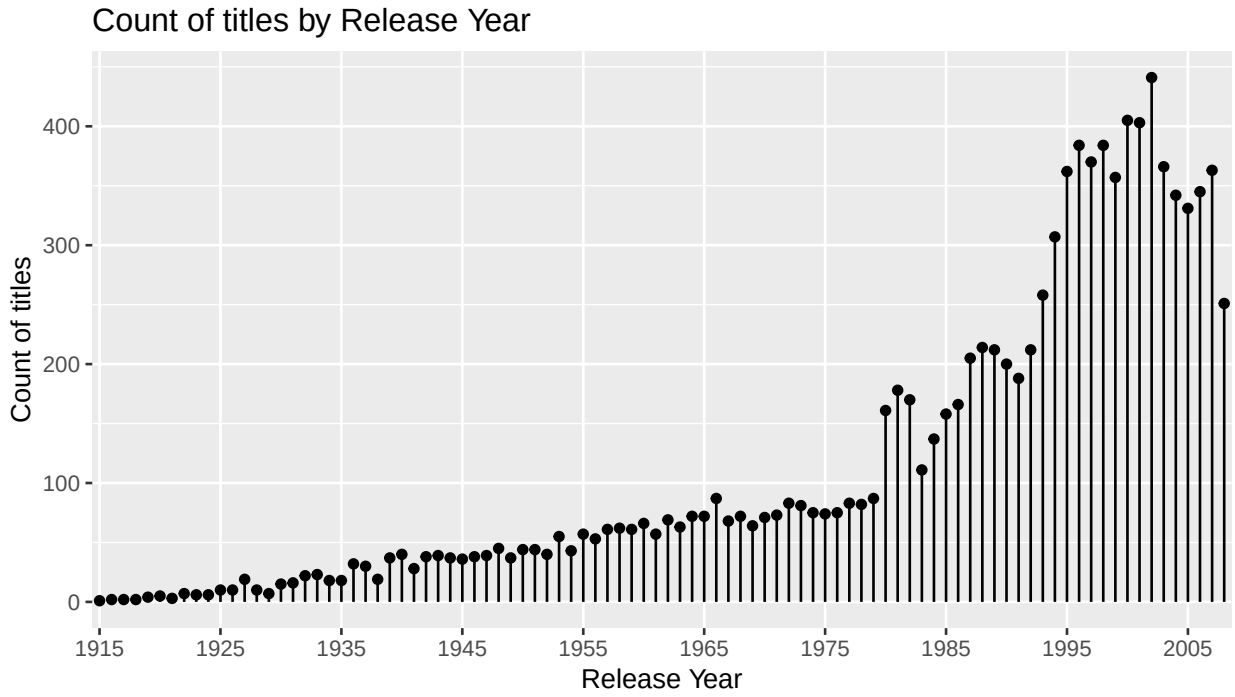
The plot of average rating vs year (above) indicates that there is a slight decline in average rating over time, by year.

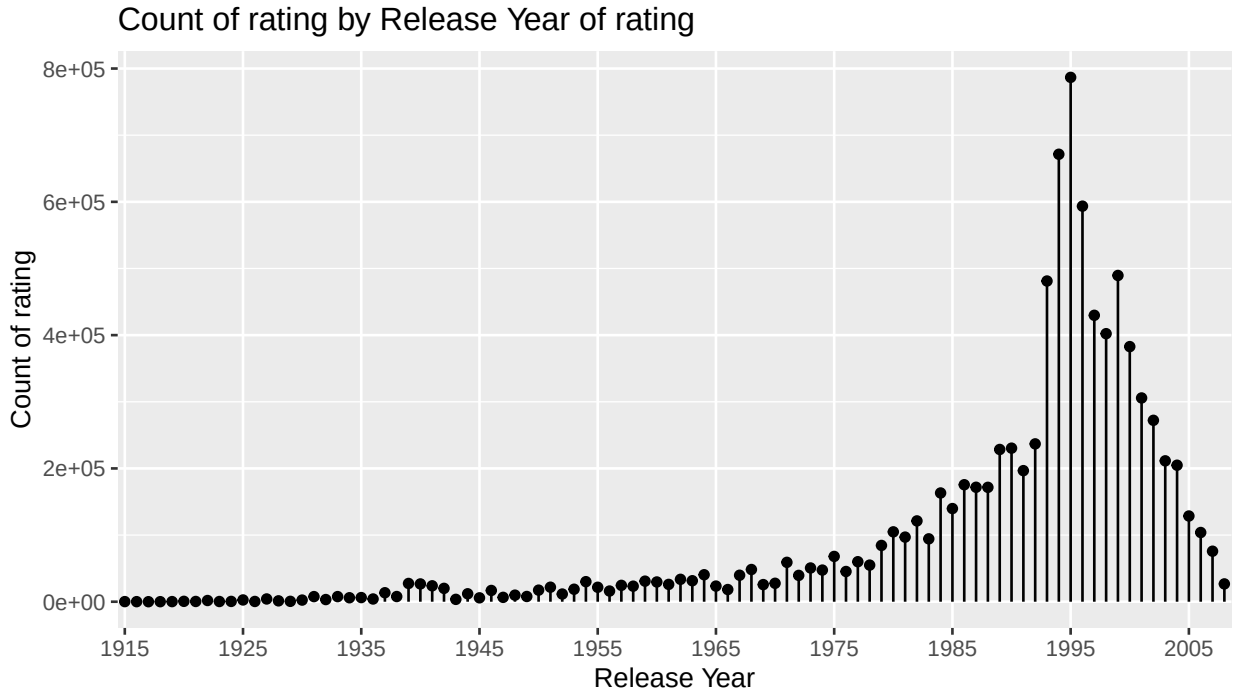
Next, we have a plot of average rating vs count of ratings.



We can see that over the years, there are fewer counts of ratings four and over. The view is obscured by the fact that the rating system changed in 2003, so that half grades (e.g. 0.5, 1.5, 2.5, 3.5, 4.5) are possible. Such a change might impact the user's ratings if a rating option between 3.0, 3.5 or 4.0 were easier (3 might bump up to 3.5; 4 might bump down to 3.5 and so on), and therefore the user appears to become more critical over time. We can see that the users are not handing out many scores of '5' or '4' after the rating system changes. Also, we know that counts in 1995 and 2009 are outliers; therefore, these years are not good indicators of a trend in category of rating counts.

Release year of the movie has some interesting information to share. First, looking at count of rating by release year, it seems that the users like to review some of the really old movies. It also seems that there is less and less interest in reviewing movies released after 1995.

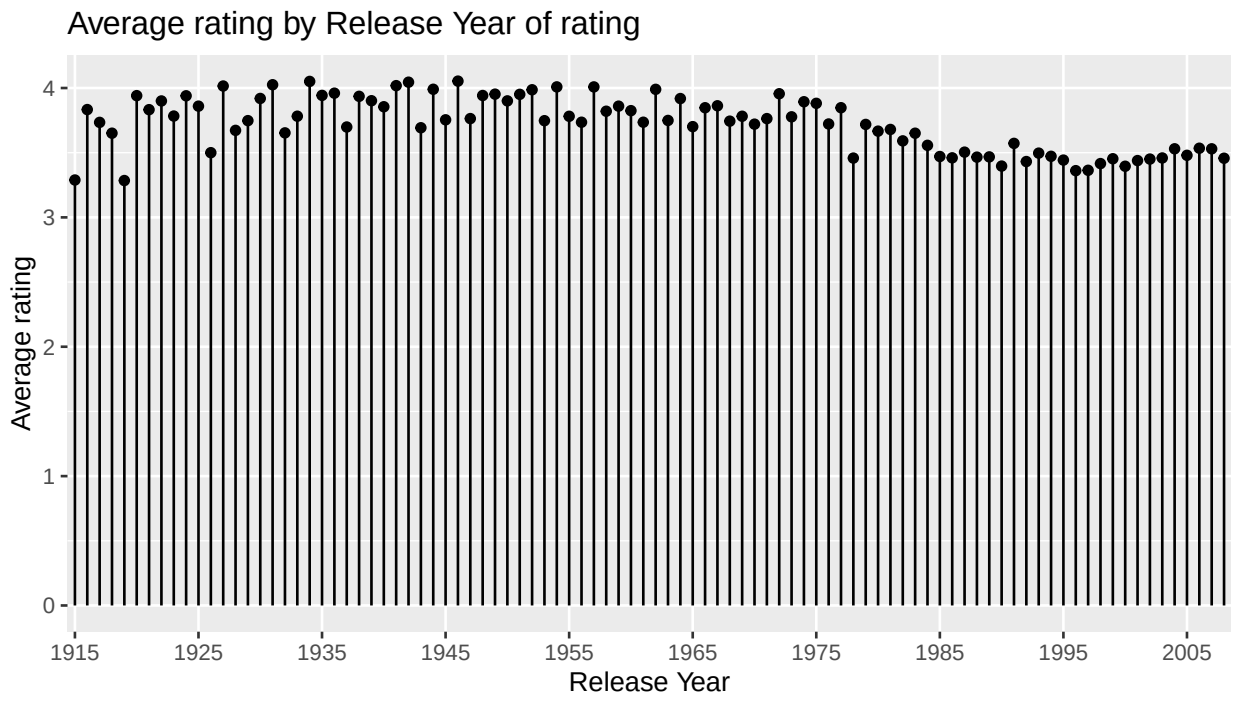




It looks like there has been no decline in the number of movies release every year since 1995 (very roughly speaking). We could suggest that users are just not increasing the number of reviews in proportion to the number of movies released. However, if it is true that users are less and less interested in reviewing more movies as more movies are released , then are recommendations based upon movie reviews going to be biased towards the tastes and preferences of those people who like to review movies and who take the time to fill in movie rating forms? Difficult to answer that question, as we do not have any information about the personal characteristics of the reviewers.

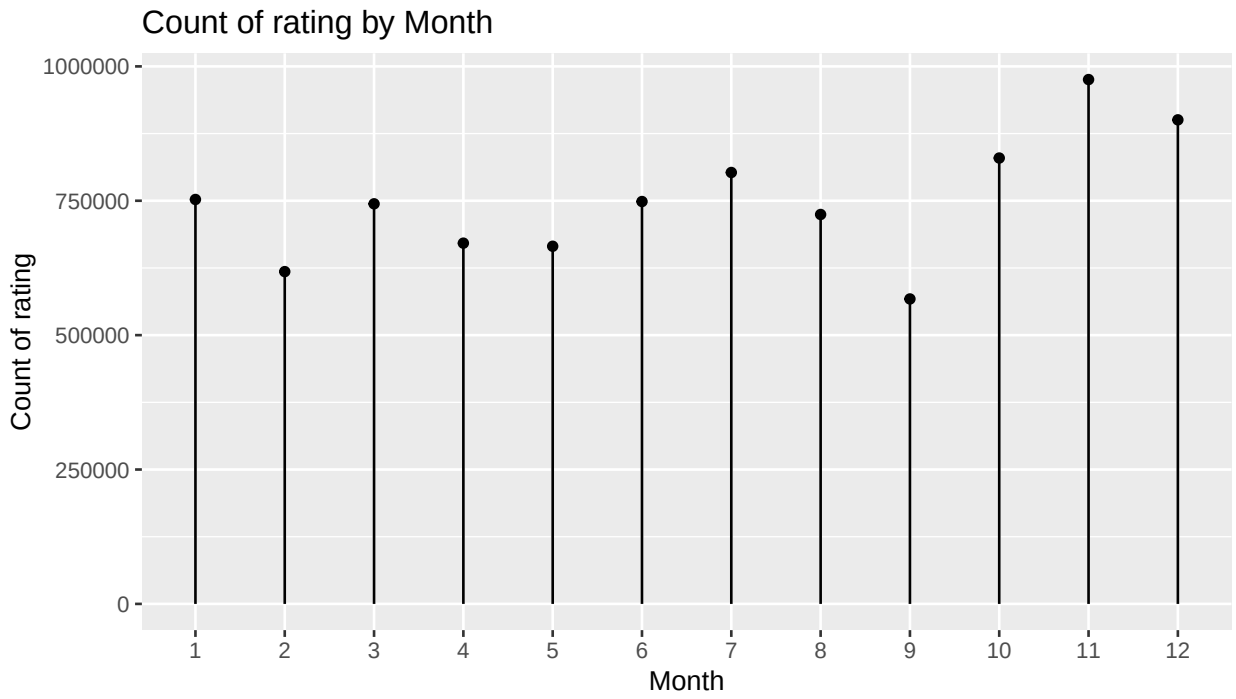
Looking for a trend in the average rating by release year, a brief view of the visual presentation reveals a

confusing story: The average rating has shown a noticeable decline, starting in about 1985. Which is a bit confusing because the trend presented here has quite a few decades to establish. So, I am wondering if the change in the rating system (as described earlier) has resulted in a more critical review.



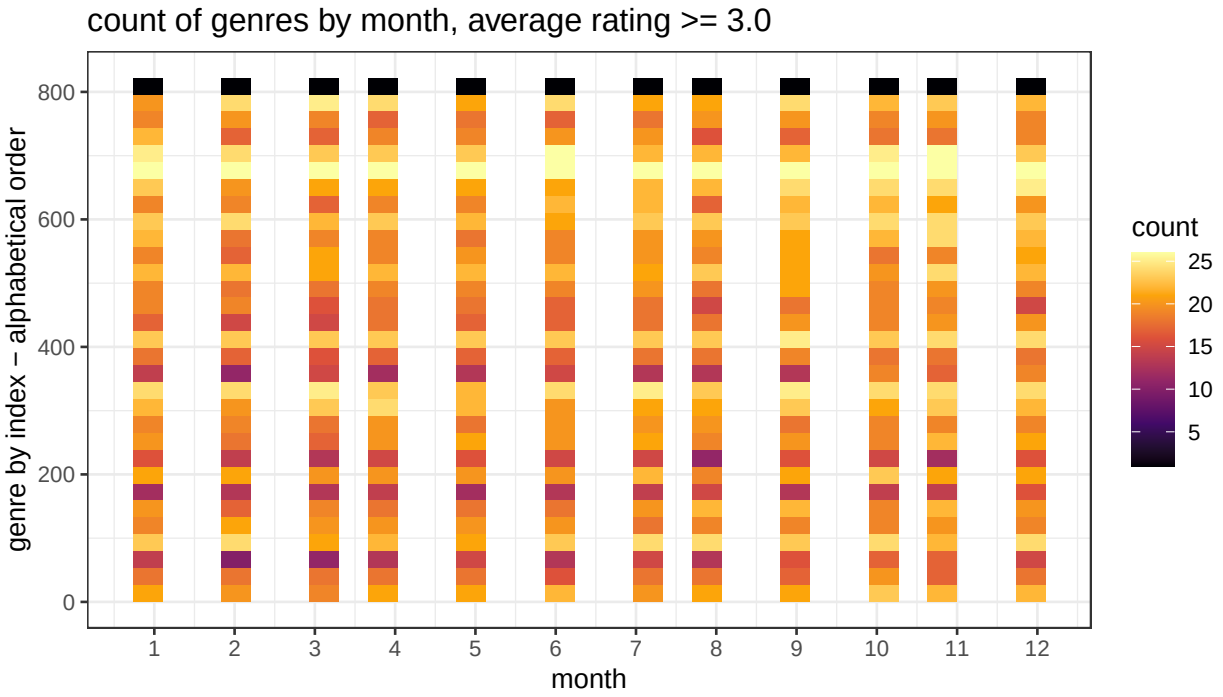
A bit of research reveals that 'blockbuster' movies are released on a fairly predictable timeline during the year[5]. Movies are typically released in June, July and August to take advantage of the fact that children are out of school and parents are looking for summer vacation activities. The next sizeable release is November and December for movies looking to win at the Oscar awards and to take advantage of the Christmas holiday season. With this in mind, the analysis of the timestamp, by month of rating, was carried out to see if there is a seasonal effect to the count of ratings or average rating.

[5]https://daveswallet.com/what-are-the-best-months-to-release-box-office-movies-and-why/

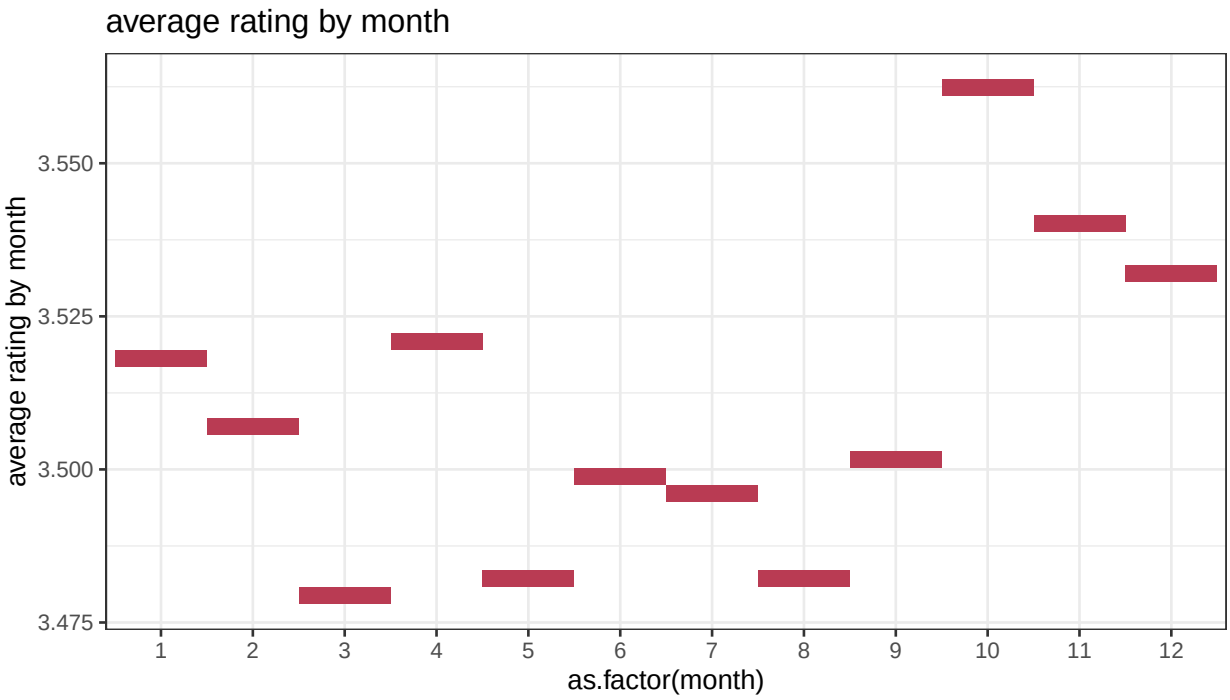


The visual above indicates, very roughly, that there may in fact be a seasonal component to the count of movie ratings. There is a gentle uptick in the count in the month of July (blockbuster releases) and then another uptick in November and December (Oscar worthy movies and then Christmas). It makes a bit of sense to discover that the movies out for release, looking for an Oscar would attract the users who like to rate movies, so the count of ratings goes up to its highest point. We don't have data on the users who rate the movies, so we can only guess at the characteristics of movie reviewers and how their preferences might be influencing the dataset.

A density histogram was created to see if a pattern of genres selection could be observed. Per the information presented in the genres section, we do have preferences. Would these preferences for a preferred genre be different in the summer months versus the months of fall (the 'blockbuster' release months versus the 'Oscar contenders' months). It is difficult to find a visualization technique that would take the entire data set and summarize it into something easy to read. Therefore, the histogram was slimmed down a bit by selecting for genres with an average rating greater than 3.0.



Looking at the histogram, we see that the genre preferences remain fairly steady over the course of a year. There is a bit of movement in the last quarter for the genres in the 200 range (looking at my index, these are the genres described by Action movies of various description…Action movies are described by index numbers from 2 to 253). Our favourite genres always seem to be included in the Drama categories: the only bright yellow line across all months is in the Drama genres(index number 658-731).



The average rating by month, when displayed on a scale that begins at 3.475, shows a discernible pattern. If we are expecting that the Oscar contenders are released in October and November, it could be that the

ratings reflect at the least, the expectation of better quality and perhaps a higher rating.

I examined the day of the month, day of the week and hour of the day, looking for useful patterns in count or average rating and did not find anything that I believed would be useful for my linear model. The work is not presented in this report.

Summing up what has been learned from the timestamp and the year of release of the movie: Reviewing the timestamp variable by disaggregating it into year and month is interesting and yields interesting bits of information about the database. There is very little revealed which indicates that a time variable based upon the day or hour of the review would improve a movie recommendation system. There is some indication that month of the year as related to blockbuster releases and Oscar contending releases, may impact upon average rating and counts of rating. The time variables indicate when movie ratings are counted and higher counts seem to indicate higher rating scores, as evidenced by the observation that the Oscar contender months, October and November, have an increase in both count of ratings and average rating. There is an observed change in rating practices after 2003 which also seems to have an impact on scoring patterns. There is some indication, based upon the year of release and year of rating, which indicate that there are shifting patterns in tastes, as exhibited by a general trend to lower average rating.

The Regression Models

The regression models were set up as described in https://rafalab.github.io/dsbook/large-datasets.html#movielens-data and the BellKor Solution to the Netflix Grand Prize. My procedure followed that described in the course text . I followed the procedures in the course text to familiarize myself with the R programming required and to document the improvements to the model possible when straightforward improvements to the model were put through the R programming.

The text describes the estimation of an RMSE obtained by estimating the mean of all test set ratings (mu_hat as the predicted_rating) and then estimating the RMSE as: sqrt(mean((true_ratings - predicted_ratings)^2)), where the true_ratings are the ratings of each movie i by each user u. The RMSE is interpreted similar to a standard deviation. If the standard error is greater than 1, the regression model generating the value is not good.

The BellKor Solution describes attempts at estimating the time effects of the changing tastes for the movies and the changing tastes of the individual movie raters. To account for changing tastes in movies over time, the BellKor team binned the time series into ten week intervals to achieve a beta for movieId that would have a preference for the movie plus a time factor that takes into account preferences over time. Their model was described by:

bi(t) = bi +bi_Bin(t)

In my model, I have binned the time of review into ten week intervals to get a beta estimate of 'bin', b_b. As noted previously in the data analysis section, there appears to be a shift in the rating scheme in 2003. Also shown in the data analysis section was a decline in rating score over time, as illustrated by the plots of the average rating by year of release of the movie. Also observed was that there are shifts in preferences by season or 'months', so perhaps binning in 10 week segments would pick up the seasonal influences. So, by binning the data, it may be possible to improve the RMSE as it captures a change in taste of movies and possibly takes into account the sudden shift in the rating that took place as the scoring scheme was adjusted.

The BellKor Solution also describes a time element of the model for individual movie raters, described by b_u(t) = bu + u $\cdot$ dev_u(t), where dev_u(t) = sign(t $-$t_u) $\cdot$ |t $-$t_u| , where is a constant (as estimated by BellKor) of = 0.4 and t is 'day of rating' and t_u is the average day of rating for user u. I estimated a variable dev_u(t), but found that my computer does not have the RAM nor sufficient processor speed to successfully process this variable. So, it was not used in my model. Although, it does make sense as a variable; I wish I could have seen it incorporated sucessfully.

The BellKor team also estimated a variable F_u,i the overall number of ratings that user u gave on day t_u,i. The analysis of the data, as previously described, supports the notion that users tend to rate a large count of movies one a single day. The BellKor team has explained that the practice of reviewing many movies at one go may influence ratings: movie watchers rating movies remember movies they like in very positive

terms and movies they dislike in very negative terms. Therefore, the frequency variable may pick up biases that exaggerate a strong like or dislike of a movie and could therefore explain user bias more thoroughly. Therefore I calculated my own 'freq' variable and its beta value b_F, where 'freq' is described by

f_u,i = (log_a (F_u,i)).

In the edx movielens database, I found that the mean of ratings per submission of ratings by users was a little above 21 ratings per submission (in other words, the ratings were lumped together on one day an average of 21 ratings submitted at one day). Estimating the effect of lumping ratings together was straightforward enough and could explain some variability in the ratings by a beta described by b_F for variable 'freq'.

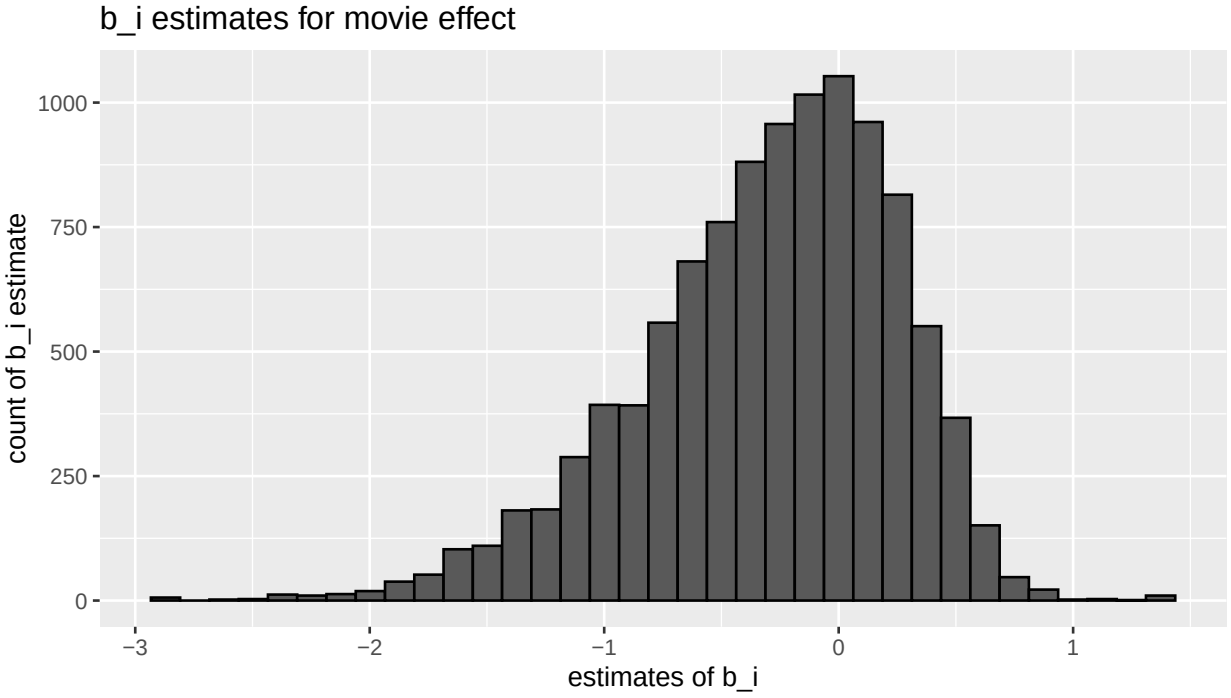
Most importantly, before splitting the edx movielens database into the training set and the test set, the independent numeric variables were scaled to a z-distribution, as recommended by a number of programming guides[6] [7]. Scaling the data to a z-distribution turned out to be the best way to obtain good beta estimates. The following variables were scaled using the formula (x-mean(x)/standard deviation (x)): rating, freq.

Finally, the data set was split into two segments (not including the adjustments made to create the 'validation' data set, as described in the course material); the train set and the test set. The test set consists of 20% of the edx movielens dataset.

The Results

The first regression is a measure of the model when we predict the same rating for all movies, regardless of the user or the tastes for the movie. This model, as in the course material, is the 'naive model'.

The second regression model obtains an RMSE resulting when the effects of individual movie ratings are included. The distribution of the movie effect beta appears to be normally distributed, although slightly skewed.

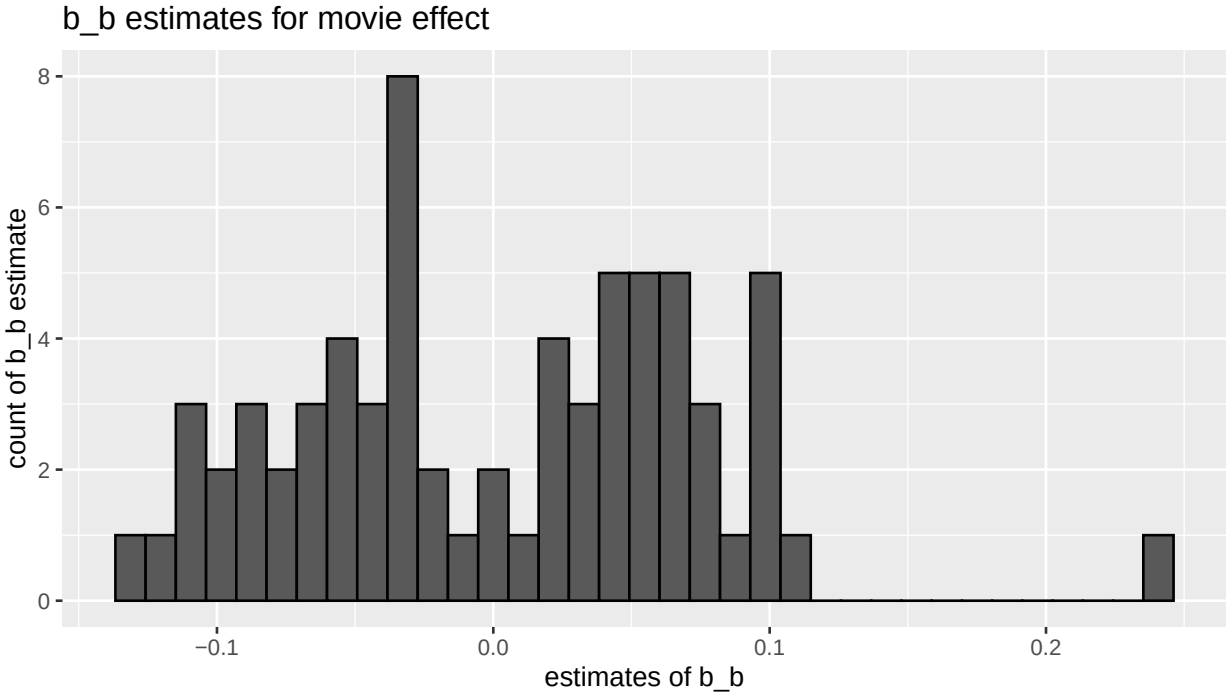


The RMSE for a model including the movie effect b_i is reduced from the initial 'naïve' model (as seen in the final results table at the end of this section).

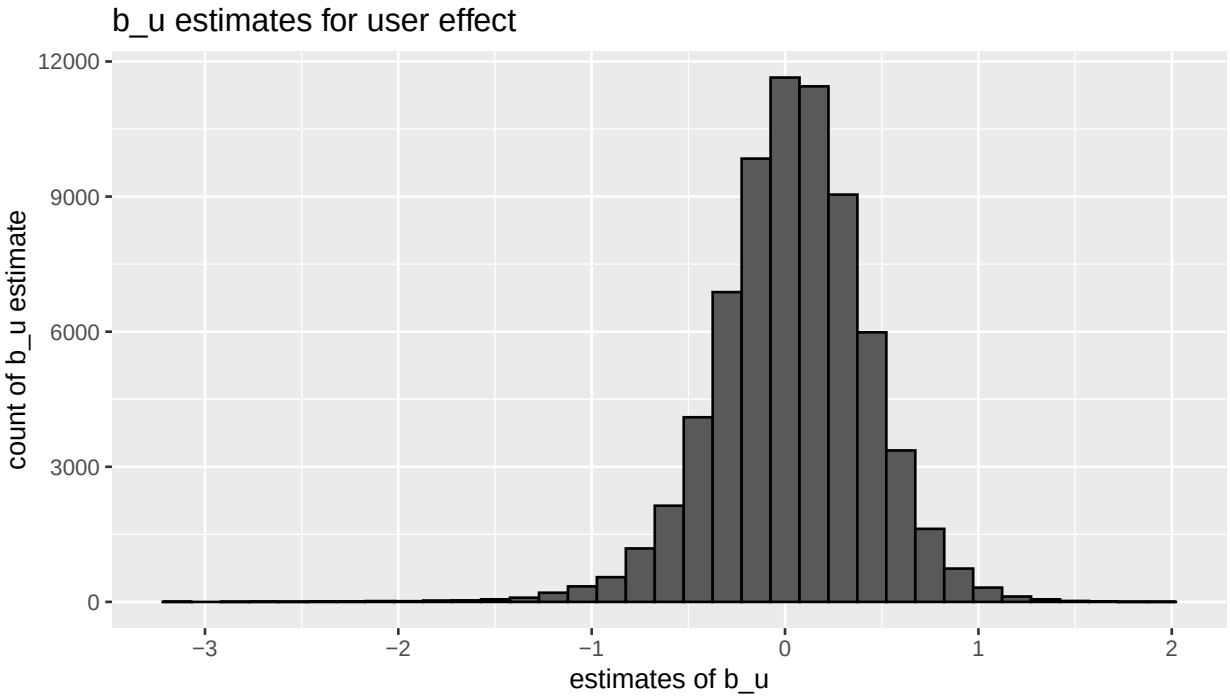
[6]https://www.programmingr.com/examples/neat-tricks/using-the-scale-function/

[7]https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/

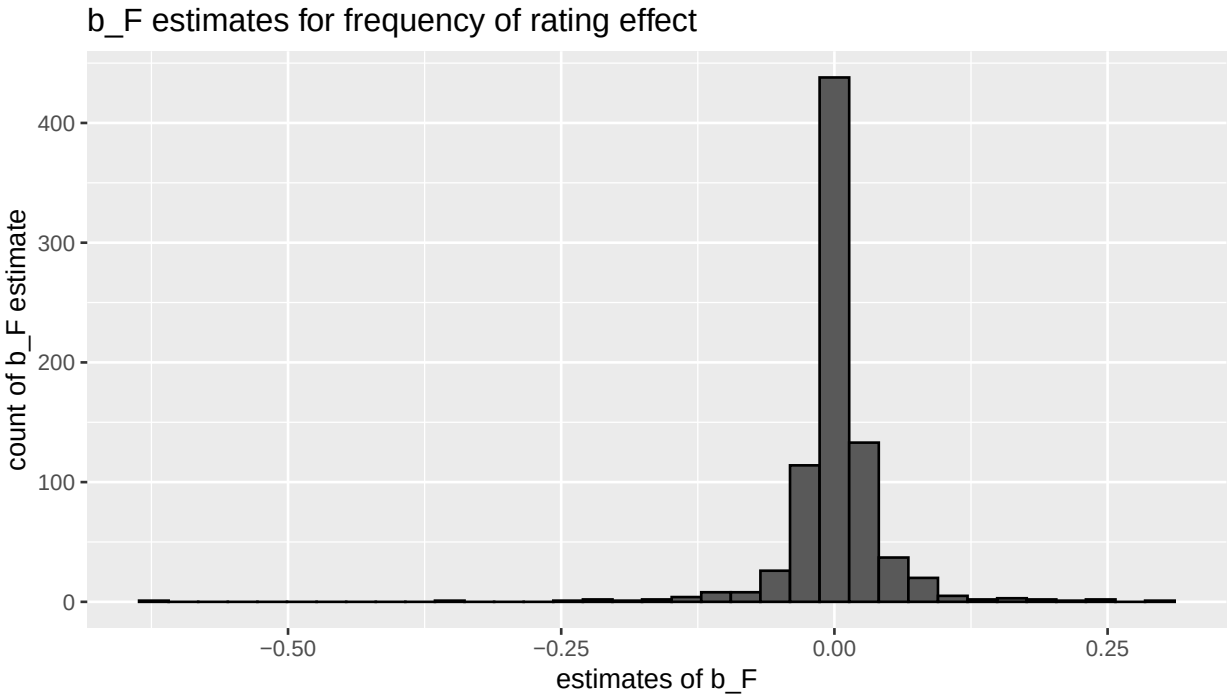
The next model estimated a beta for the binning effect, the change in the tastes for movies of different sorts, as measured over time. As can be seen, the distribution of the bin effect does not appear to be normal; there appears to be a binormal distribution, with one evident outlier. As discussed earlier, there was a change in the scoring of movies in 2003…it could be that the binning effect has captured the change, resulting in a shift in the beta. If this is the case, then the binning effect is misidentified and has not properly captured shifts in rating (or movie preferences) over time. If there were more time available to me, I would consider adding a dummy variable into the model which might capture the shift in scoring in 2003, which might lead to a better measure of the binning effect beta.



The third regression includes the effects of an individual user on the movie ratings. This model takes into account the fact that different movie watchers have different preferences or rating schemes and therefore don't rate each movie on exactly the same criteria. The distribution for the user effect is illustrated in a histogram; the distribution appears to be normal. Noted is that the user effect has the single greatest impact of any explanatory variable on reducing the RMSE in the model.



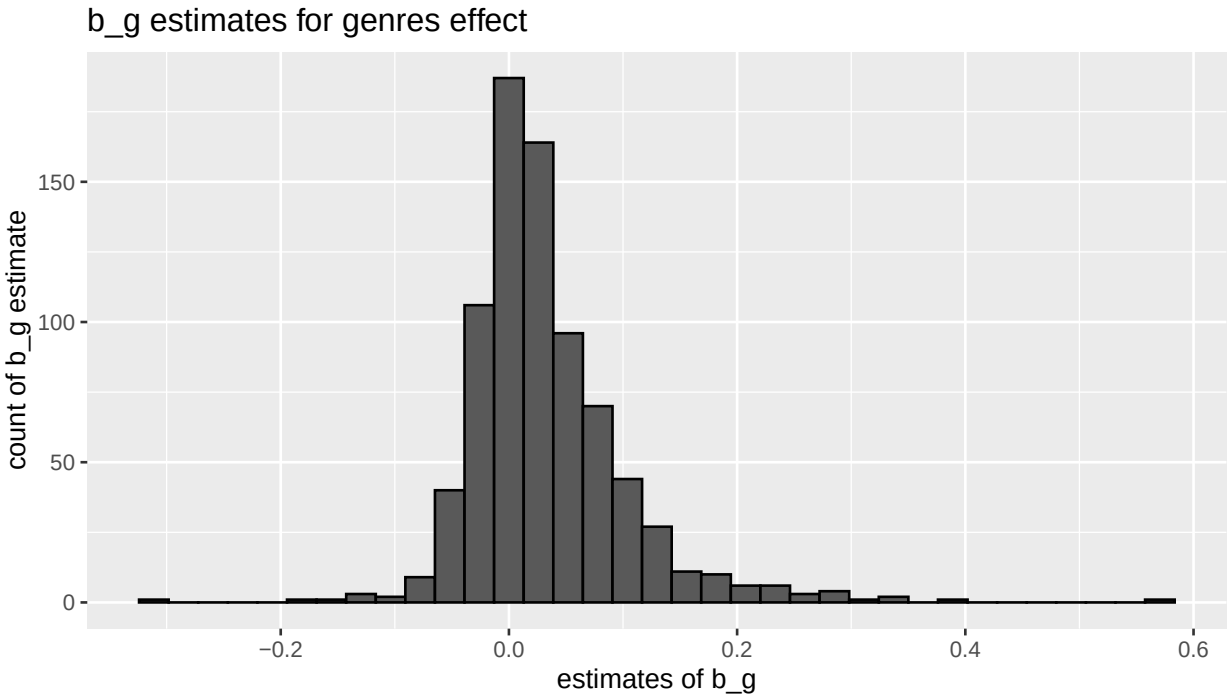
The next model includes the frequency of ratings effect. The frequency of ratings effect beta is shown in the histogram below:



The distribution of the freq beta appears to be normal, with very little spread. The frequency variable contributes to the reduction of the RMSE, but certainly does contribute to a dramatic drop in the RMSE as was documented by the BellKor Solution.

The regression model finally includes the genre effect. A histogram of the distribution of the genre effect beta is below. The distribution appears to be normal. It therefore makes sense to see that when the user

effect is present, the genres effect shows a positive effect; different users have different genres preferences and this may show up even in the genres of movies selected for reviewing.

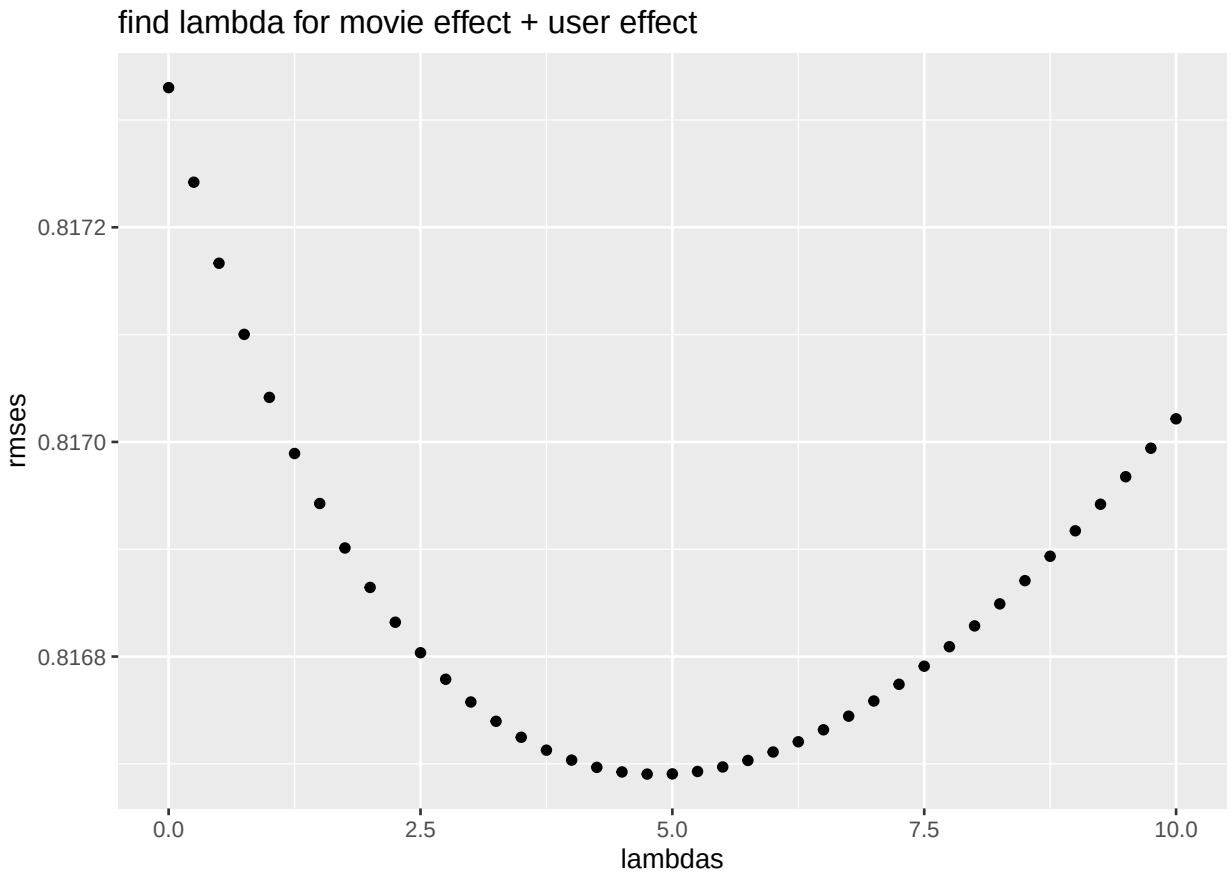


As described in the course text, a regularized model takes into account that a movie with few reviews but a very high rating will have a misleading and over- sized impact on the sample average variable estimates. The bi estimates for such obscure movies will have a larger variance (they are 'noisy') leading to an increase in the RMSE of the model. To remove the influence of these records, they are treated with a penalty (as represented by lambda) which reduces the value of a small rating count b(i) towards zero, such that it has little impact on the average sample beta estimate and reduces the error term of the model, improving the RMSE. The equation to obtain the regularized b(i) is:

b_hat_i( )=(1/ +i)* (Y_u,i - u_hat), where the summation goes from u=1 to number of ratings for movie i.

To obtain an estimate of lambda for the regularized equation, a program is run to find the lambda which minimizes the RMSE.

The grid below finds the lambda for the two largest contributors to the model: b_i, the beta for the movie effect and b_u, the beta for users. As can be seen, a lambda of approximately 5.0 will result in the best RMSE.



In the BellKor Solution paper, a separate lambda was estimated for each variable. I did not attempt to find a separate lambda for each variable, as the lambda of 5 estimated for the combined movie effect and the user effect appeared to be quite suitable.

The RMSE for all equations is shown below. From what we see so far, the model including movie effect+bin effect + user effect+frequency of rating + genres effect shows the best RMSE; standardizing the variables with a lambda equal to 5 brings the RMSE to the lowest value of 0.816268.

Table 4: RMSE results, all models

method	RMSE
naive model	1.0002168
include movie effect	0.8902753
include bin and movie effect	0.8876198
include movie effect + bin effect + user effect	0.8175100
include movie effect + bin effect + user effect+freq effect	0.8172486
include movie effect + bin effect + user effect+freq effect+genre effect	0.8169254
include movie effect, regularize with lambda = 5	0.8902466
include movie effect + bin effect, regularize with lambda = 5	0.8875589
include movie effect + bin effect + user effect, regularize with lambda = 5	0.8168391
include movie effect + user effect + bin effect+freq effect, regularize with lambda = 5	0.8165781
include movie effect + user effect + bin effect+freq effect+genres effect, regularize with lambda = 5	0.8162675
validation set	0.8155724

Conclusions

The models tested yielded the greatest effect from the relationships between ratings and the user,u and the movie, i. All other factors entered into my model were improvements, but in the linear model, they only contributed small increments toward understanding the tastes and preferences of users. Factor analysis or other techniques might be better techniques for nonlinear variables such as genres or for changes in taste over time. Also noted is that to create efficient code, it is important to remember that the database is huge and that pulling out only the variables of interest for a particular process will save on processing speeds and RAM. Also learned was that despite the allure of complex models with lots of detailed estimations, the RAM available for processing will limit what is possible.

To improve processing speed for some data manipulations and visual presentations, I created an index for the genres. It would be nice to have that index created formally, so that everyone processing the data base has easy access and understanding of the index created. The most important lesson learned was the value of scaling the variables, as the best improvements to RMSE were obtained with the scale transformations. Standardizing the beta estimates also yielded some very good improvements to the values obtained.