MovieLens: a movie recommendation system

Rob Meekings

10/12/2020

Contents

1 Executive Summary

1.1 This report summarizes the findings of exploratory data analysis performed on the MovieLens dataset to investigate the factors that influence how users of the MovieLens website rate movies. This analysis has informed a number of extensions to the recommendation system model developed as part of the HarvardX Data Science course.

1.2 The model presented in this paper scores a RMSE of 0.8649356.

1.3 The exploratory analysis raises a number of questions for further study and opportunities to further extend and develop the model. These are presented at the end of the paper along with some research questions that could go a long way to improving our understanding in this field.

2 Introduction

2.1 This report summarizes the development and results of the MovieLens movie recommendation system, developed in R by Rob Meekings as part of the Data Science: Capstone module (HarvardX PH125.9x).

2.2 The goal of this project is to train a machine learning algorithm on a subset of the MovieLens data (the *training* set) to correctly predict *movie ratings* in a subset of the MovieLens data that is not used in making predictions (known as the *test* or *validation* set).

2.3 This report is intended to be read by someone familiar with the module and course content, as such the reader should be familiar with the R programming language and data science topics.

2.4 This report complies with the (edX Honor Code).

3 The MovieLens Data

3.1 The source data for this project comes from the MovieLens dataset, which contains records of *ratings* assigned to *movies* by *users*. The data has been pre-processed using code provided to split it into *training* and *test* sets.

Source code

3.2 This source code, developed by HarvardX as part of the course materials for this project, is reproduced here:

```
##########################################################
# Create edx set, validation set (final hold-out test set)
##########################################################

# Note: this process could take a couple of minutes

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

dl <- tempfile()
download.file("http://files.grouplens.org/datasets/movielens/ml-10m.zip", dl)

ratings <- fread(text = gsub("::", "\t", readLines(unzip(dl, "ml-10M100K/ratings.dat"))),
                 col.names = c("userId", "movieId", "rating", "timestamp"))

movies <- str_split_fixed(readLines(unzip(dl, "ml-10M100K/movies.dat")), "\\::", 3)
colnames(movies) <- c("movieId", "title", "genres")

# if using R 3.6 or earlier:
movies <- as.data.frame(movies) %>% mutate(movieId = as.numeric(levels(movieId))[movieId],
                                           title = as.character(title),
                                           genres = as.character(genres))
# if using R 4.0 or later:
#movies <- as.data.frame(movies) %>% mutate(movieId = as.numeric(movieId),
#                                           title = as.character(title),
#                                           genres = as.character(genres))


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

# Validation set will be 10% of MovieLens data
set.seed(1, sample.kind="Rounding") # if using R 3.5 or earlier, use `set.seed(1)`
test_index <- createDataPartition(y = movielens$rating, times = 1, p = 0.1, list = FALSE)
edx <- movielens[-test_index,]
temp <- movielens[test_index,]

# Make sure userId and movieId in validation set are also in edx set
validation <- temp %>%
  semi_join(edx, by = "movieId") %>%
  semi_join(edx, by = "userId")

# Add rows removed from validation set back into edx set
removed <- anti_join(temp, validation)
```

```
edx <- rbind(edx, removed)

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

3.3 This code produces two datasets, a *training* set called `edx` with 9,000,055 rows, and a *test* set with 999,999 records called `validation`, (which is 10% of the total,) that we reserve for model validation. These datasets have the same structure, they both have 6 columns: userId, movieId, rating, timestamp, title, genres .

3.4 The *summary()* function, when applied to the `edx` dataset, gives us some information, but this is rather mechanical and not very enlightening:

Table 1: Summary of the edx dataset

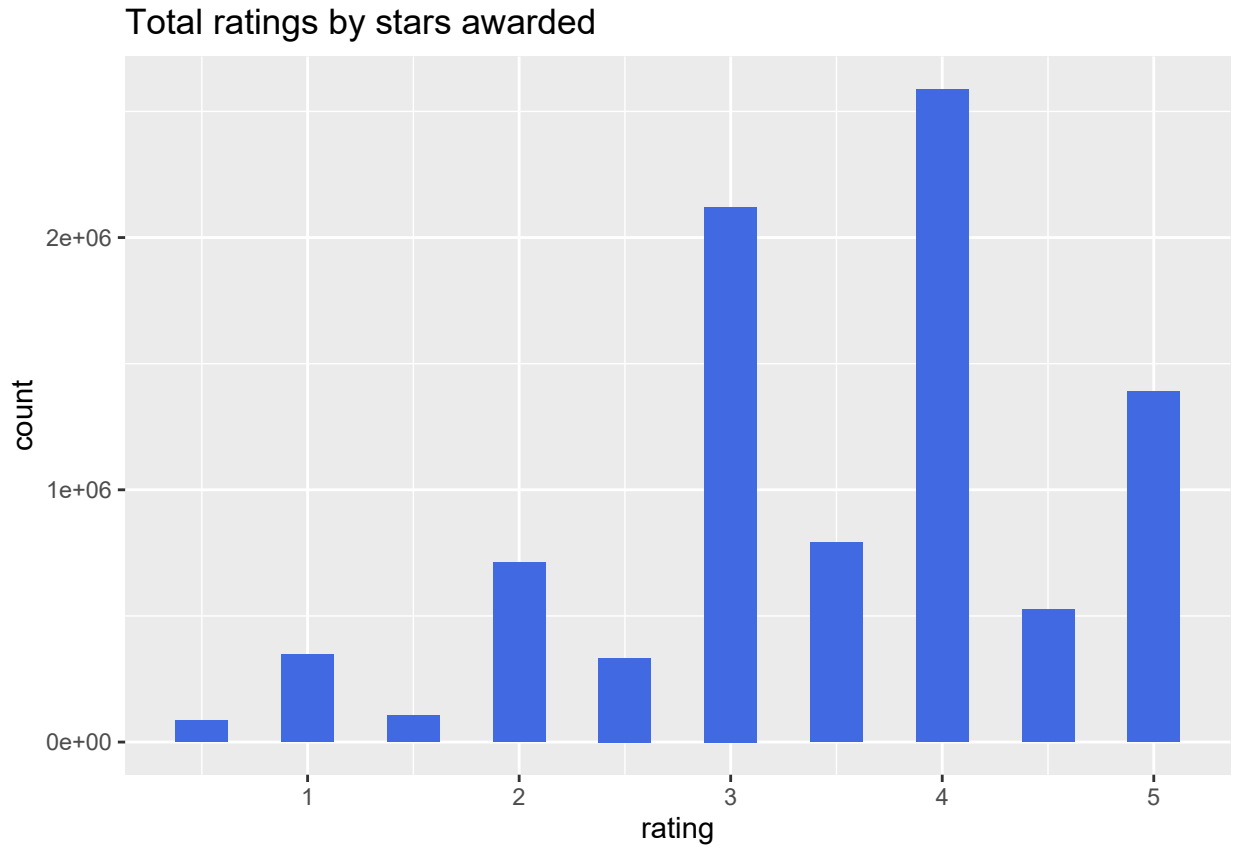
userId	movieId	rating	timestamp	title	genres
Min. : 1	Min. : 1	Min. :0.50	Min. :7.90e+08	Length:9000055	Length:9000055
1st Qu.:18124	1st Qu.: 648	1st Qu.:3.00	1st Qu.:9.47e+08	Class :character	Class :character
Median :35738	Median : 1834	Median :4.00	Median :1.04e+09	Mode :character	Mode :character
Mean :35870	Mean : 4122	Mean :3.51	Mean :1.03e+09	NA	NA
3rd Qu.:53607	3rd Qu.: 3626	3rd Qu.:4.00	3rd Qu.:1.13e+09	NA	NA
Max. :71567	Max. :65133	Max. :5.00	Max. :1.23e+09	NA	NA

4 Exploratory Data Analysis

4.1 We can better understand the data if we dig a little deeper into the distributions of *ratings*, relationships between the numbers of *movies* and *users*, and the distribution of these over time, represented in the data by *timestamps*, and *genre*. We can also look for any interactions between *titles*, *genres* and other terms to see if they hold information that might tell us something about *ratings* or have predictive power.

Ratings

4.2 *Ratings* are assigned to *movies* by *users*, it is assumed that the *user* has watched the *movie* in question and that the *ratings* are fair. The *rating* is represented by a number of stars, with a higher star *rating* denoting greater enjoyment or appreciation. The lowest *rating* that can be given is half a star, *ratings* rise by half star steps to the top *rating* of five stars. The historgram below summarizes the distribution of star *ratings*.

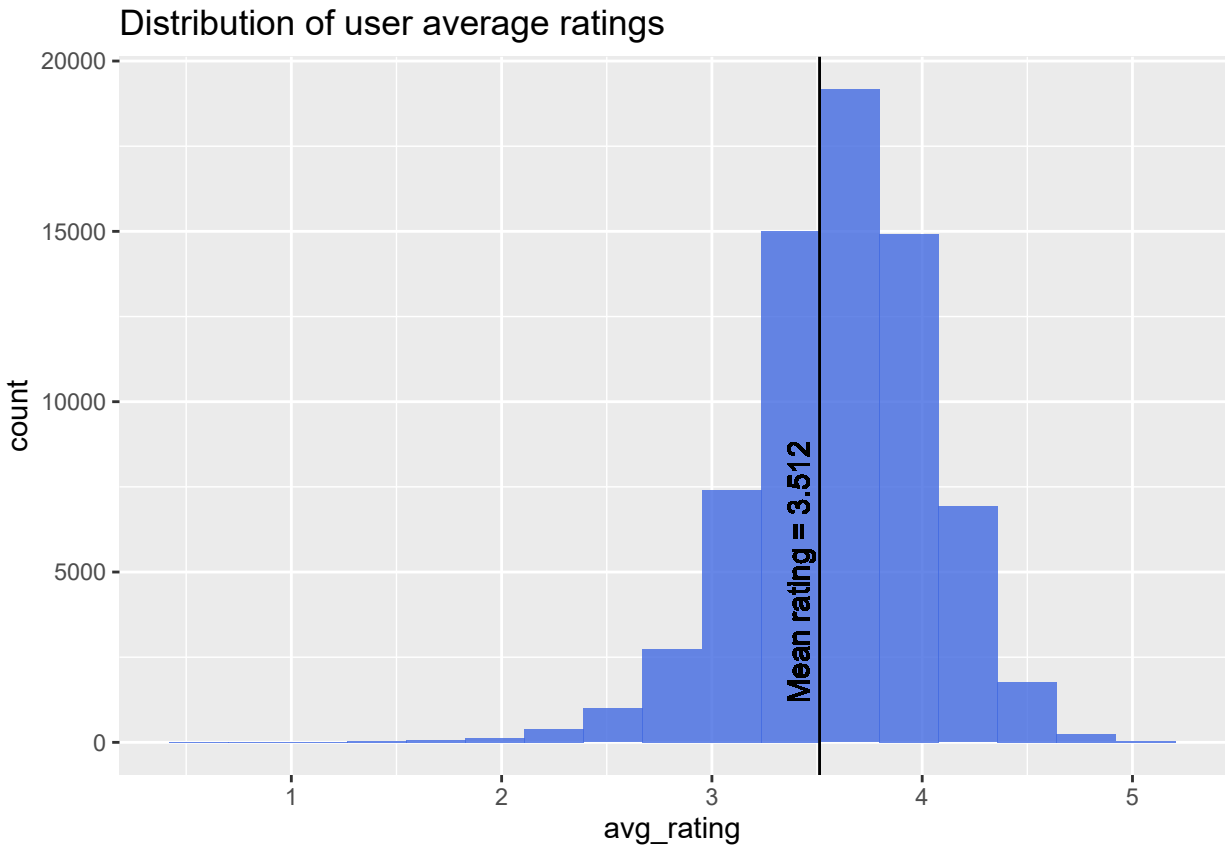


4.3 It appears that there is a positive bias in ratings, perhaps there is a tendency for *users* to rate *movies* that they like. We can quantify this by observing that *ratings* have a mean 3.512 and median of 4 .

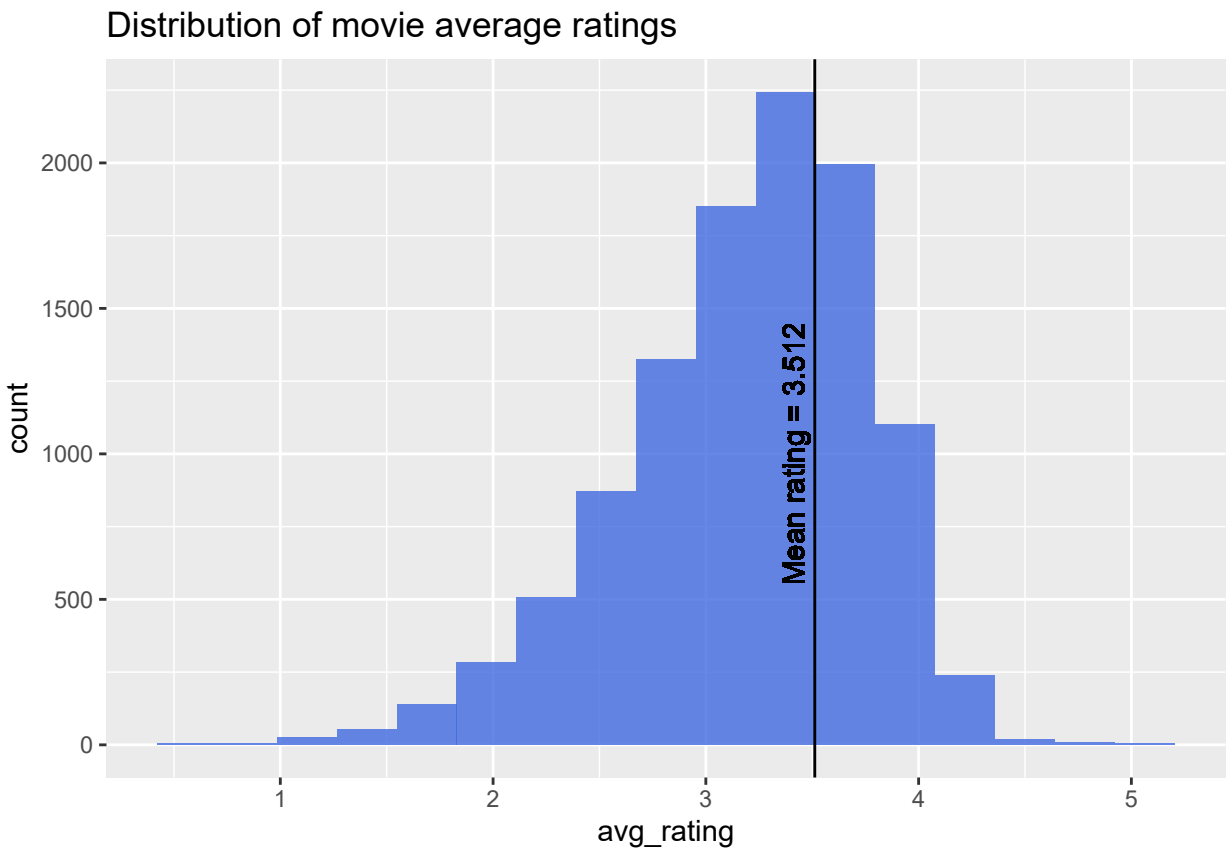
Users and movies

4.4 As a *rating* reflects the views of a single *user* about a particular *movie*, so the 9,000,055 *ratings* in the dataset reflect the views of the 69,878 *users* of 10,677 *movies*, suggesting that each *user* has rated an average of 129 *movies*, and that each *movie* has *ratings* from 843 *users*.

4.5 We can examine the distribution of ratings by *user* and by *movie* to look at whether there are *users* who are more or less positive than the average, and whether there are *movies* that tend to get better or worse *ratings* than the average. First lets look at a histogram of *user* average *ratings*, our choice of bin size in the histogram affects how smooth the emerging bell curve appears.

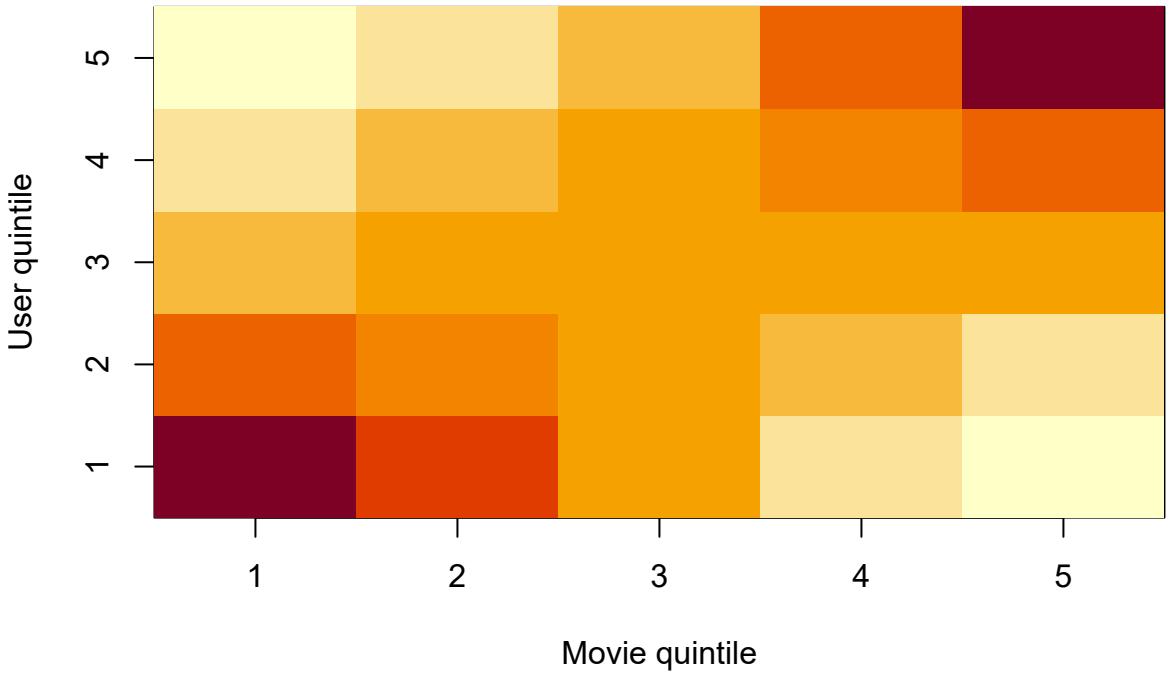


4.6 The distribution of *users'* average *ratings* are distributed around the mean of all *ratings*, but with a slight positive skew. We see a similar pattern with the distribution of average *ratings* by *movie*.



4.7 The distribution of average *rating* by *user* and by *movie* look as if they are starting to approximate normal curves, to the extent that they can censored both above and below, with this becoming more apparent in a histogram as the number of bins increases.

4.8 We should check that we're not confounding *ratings* by *users* with *movies*, and vice versa, we can approach this by stratifying the data. This would happen if good movies were only rated by positive users and less popular movies were rated by negative users: the distribution of movies' average ratings would then reflect the distribution of the positivity of the users, rather than the popularity of the movies. To do this we can stratify the movies and users by average review,



4.9 This image shows how the quintiles relate, with deeper colour indicating greater volume, there is an interaction between positivity of users and popularity of movies, because of the way we have defined our measure: it is hard to be a top quintile reviewer if you have given *movies* low scores. This gives rise to the saddle shape we see. What should reassure us is the spread in the middle: the *users* in the third quintile are almost evenly spread across all *movies* and likewise *movies* in the third quintile have reviews nearly evenly spread across the full range of *users*. If confounding was a significant factor we would expect the main diagonal to have all of the volume and very little elsewhere, which is not what we see here, and this suggests that the extent of any confounding is limited.

4.10 These numbers, especially the average number of *movies* rated per *user*, look a little high, motivating further investigation. If we organise *users* into deciles, based on the number of *ratings* they have made, it appears that some *users* are significantly more prolific in the number of *ratings* they have made, skewing the average.

4.11 This distribution is skewed towards many *users* submitting fewer *ratings*, the median number of *ratings* is 62 , and the modal class is 20 . The extreme *users*, the *superusers* with thousands of reviews, may be system accounts uploading default *ratings* as new *movies* are added to the system.

4.12 Similarly some *movies* are more frequently rated, perhaps as a result of being more widely advertised, a *block-buster* effect. We can see this by arranging *movies* into deciles by numbers of *ratings* and looking at the distribution.

4.13 Again, this distribution is skewed towards many *movies* having few *ratings*, whilst a few *blockbusters* have a very large number of ratings; the median number of *ratings* is 122 , and the modal class is 4 , with 126 , (1%) *movies* having only one review.

Table 2: Deciles of reviews per user

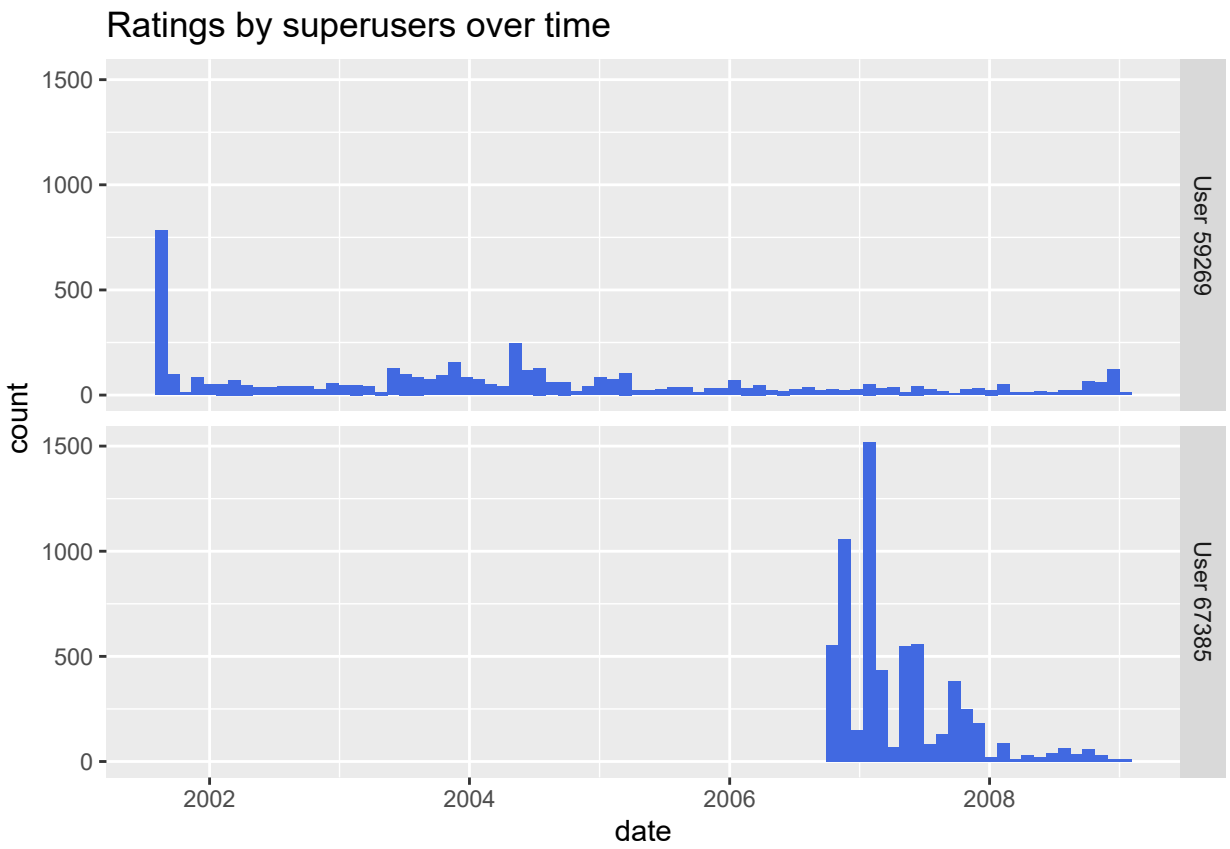
	count
0%	10
10%	22
20%	28
30%	36
40%	47
50%	62
60%	85
70%	116
80%	176
90%	301
100%	6616

Table 3: Deciles of reviews per movie

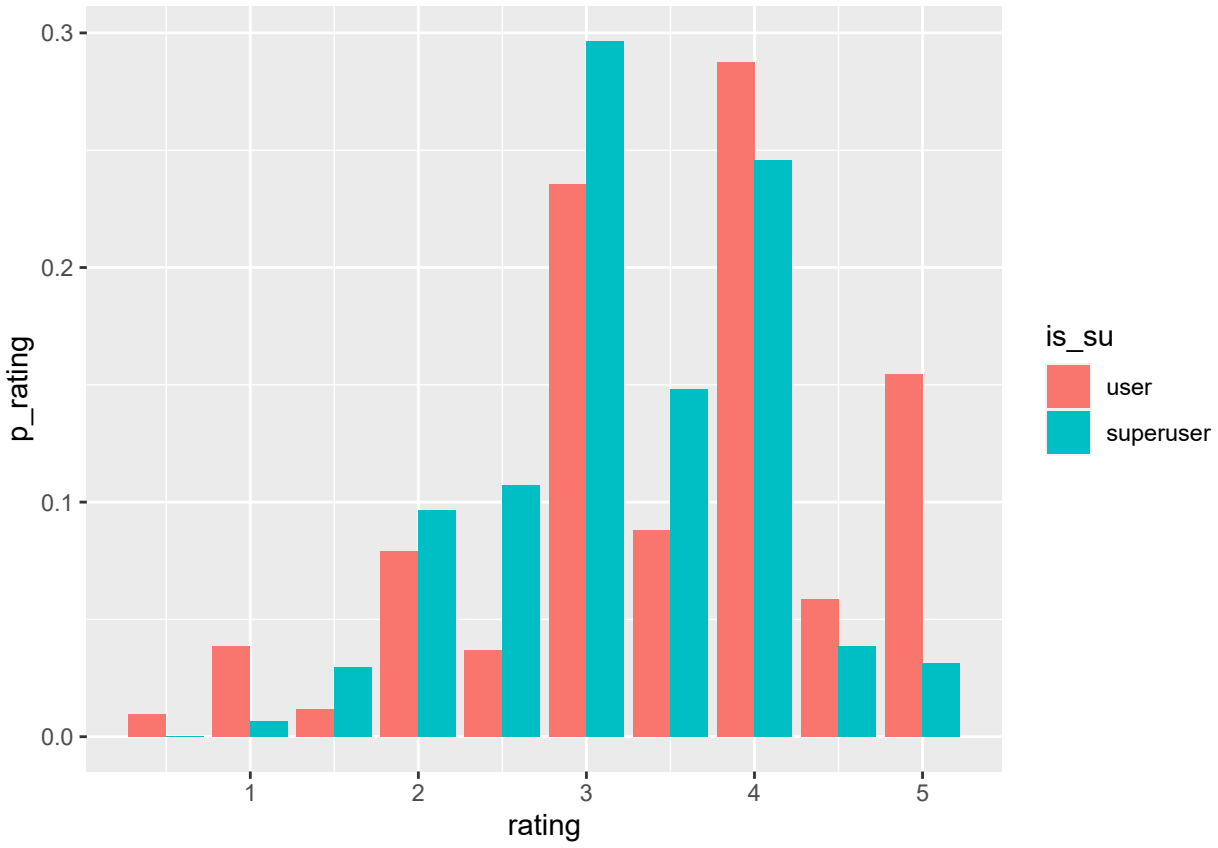
	count
0%	1
10%	10
20%	23
30%	40
40%	70
50%	122
60%	210
70%	397
80%	833
90%	2150
100%	31362

Superusers

4.14 We can identify *superusers* in the data as the *users* responsible for the greatest number of ratings, we can set the threshold for being a *superuser* where we like, but the chart below looks at *superusers* as accounts that are associated with more than 5,000 reviews.



4.15 These *superusers* are responsible for reviews of 8,463 *movies*, 79% of all *movies* in the `edx` dataset. Do *superusers* review differently to less prolific *users*? We can look at the distribution of *ratings* and compare across the two groups.



4.16 Here we see that *superusers* appear to give slightly lower *ratings* on average, and are less extreme in the *ratings* they give. This is borne out by the more analytical summary of *ratings* by *users* and *superusers*, below.

Table 4: Rating distribution by user type

Type	Average	Median	StdDev	Reviews
user	3.51	4	1.061	8987079
superuser	3.23	3	0.811	12976

4.17 This analysis quantifies the extent to which *users* rate *movies* more highly than *superusers*, ranking *movies* a quarter of a star higher, on average. The lower standard deviation for *superusers* also demonstrates the greater central tendency amongst their ratings.

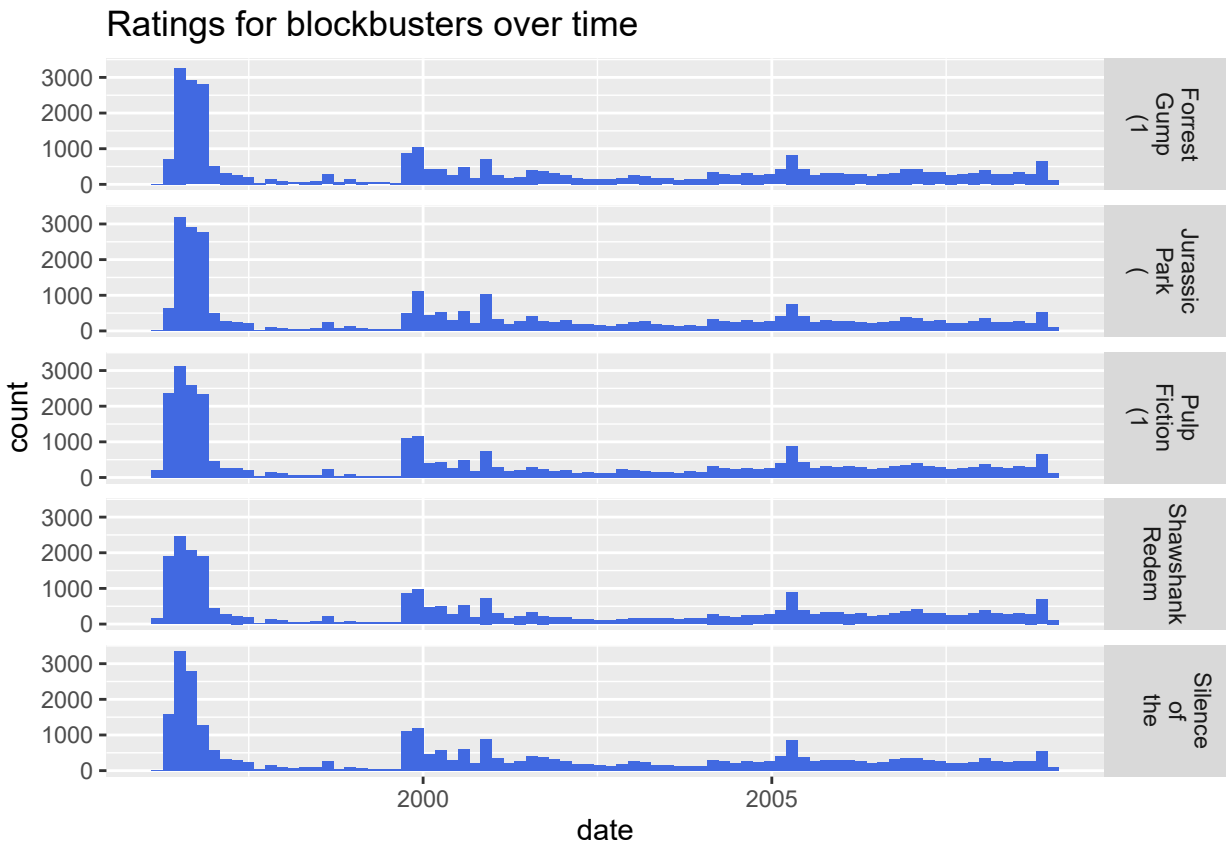
Blockbusters

4.18 We can take a similar approach to identify the *blockbusters*, *movies* that have been rated many times, to explore this further we will look at *movies* with more than 28,000 *ratings* - this cut-off is fairly arbitrary, but gives us five titles to consider, a manageable number.

Table 5: Blockbusters, movies with more than 28,000 ratings

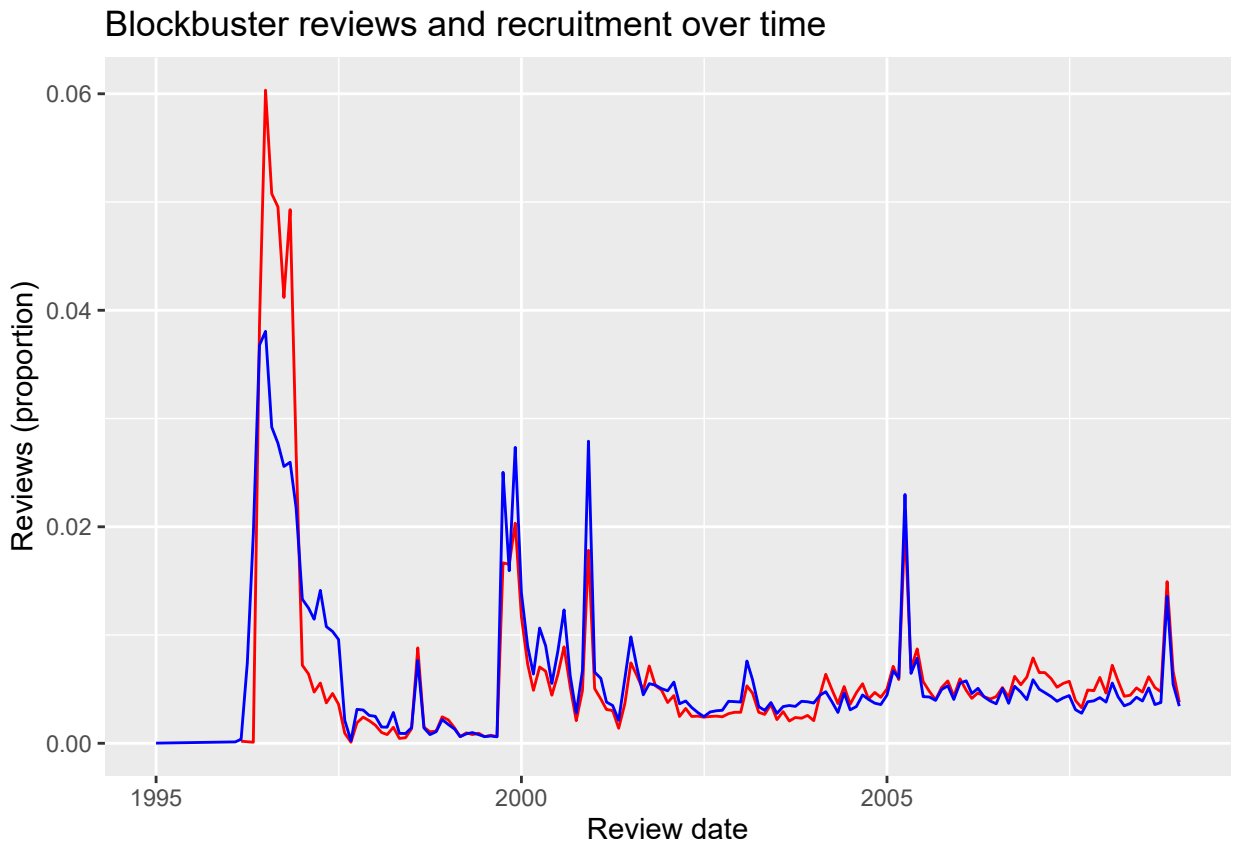
title	reviews
Pulp Fiction (1994)	31,362
Forrest Gump (1994)	31,079
Silence of the Lambs, The (1991)	30,382
Jurassic Park (1993)	29,360
Shawshank Redemption, The (1994)	28,015

4.19 This list of *blockbusters* makes some sense, these are films that are household names, but it is curious that this list is skewed towards films from quite a narrow 3-year window before the data collection period. This may be because the MovieLens data was intended for a specific use such as movie rentals, which might be heavily focused on recent films, whilst older films, such as *Star Wars* (9th in the list by ratings), or classics like *Gone with the Wind* (243rd), may have been available to watch on TV, or the demographic it was targeted at - in 2000 internet usage was much lower, below 50% in the US and 25% in the UK, and skewed towards the young and affluent. It is also interesting that *blockbusters* current at the time the data was collected, such as *Titanic* or *The Matrix* (56th and 23rd, respectively) do not feature more highly.



4.20 All of these *blockbusters* were from the early nineties, suggesting they would have been in recent memory in the early days of the MovieLens data. This recency puts a ceiling on how long ago a film was watched and therefore

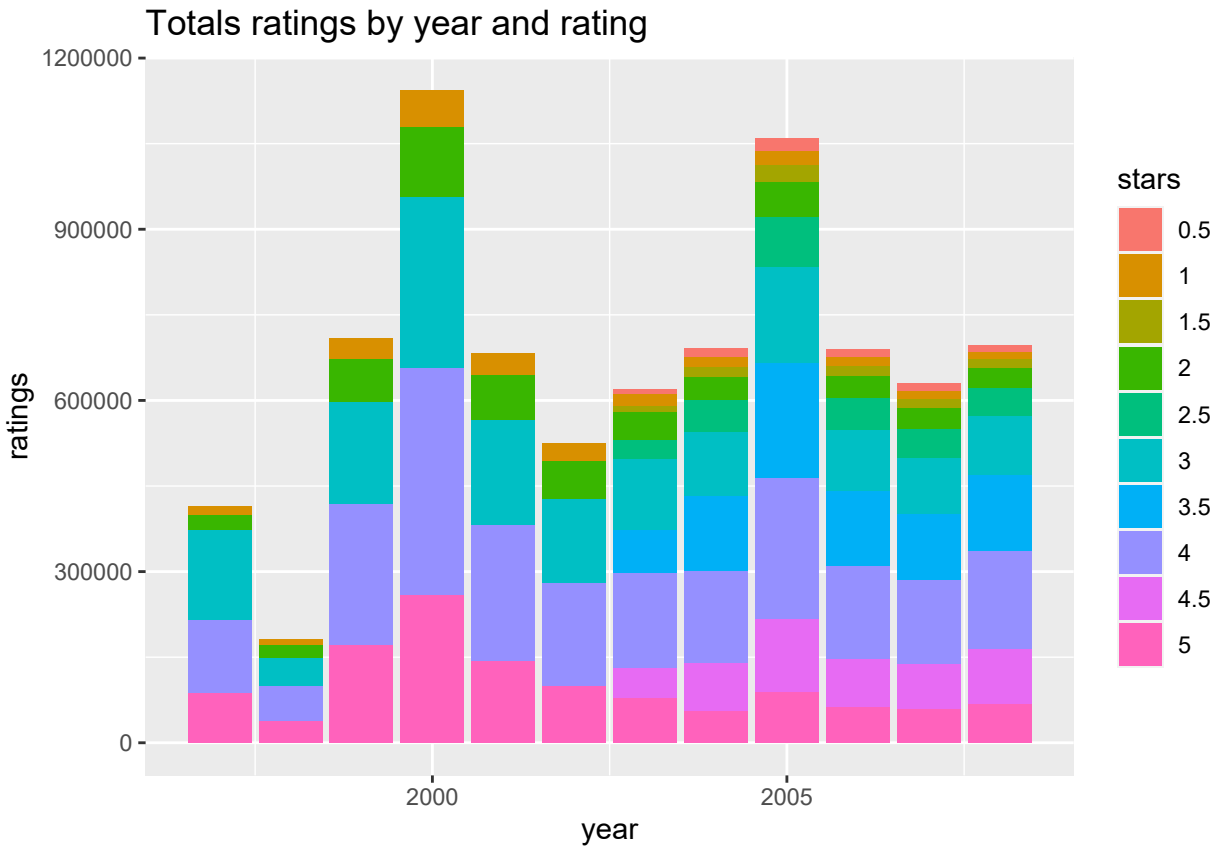
any *ratings* should better reflect the *users* view of the film, rather than their memory of their enjoyment of a half-forgotten film. What is interesting is that there appear to be very similar patterns in the volumes of reviews for these *blockbusters*. One hypothesis is that when *users* join they rate films that they recognise, the spikes would then correspond to recruitment activity. We can look for evidence of this in the data by comparing the distribution of reviews of one of our *blockbusters* (red), in this case *Forrest Gump*, to the distribution of first reviews (blue), taken as a proxy for recruitment.



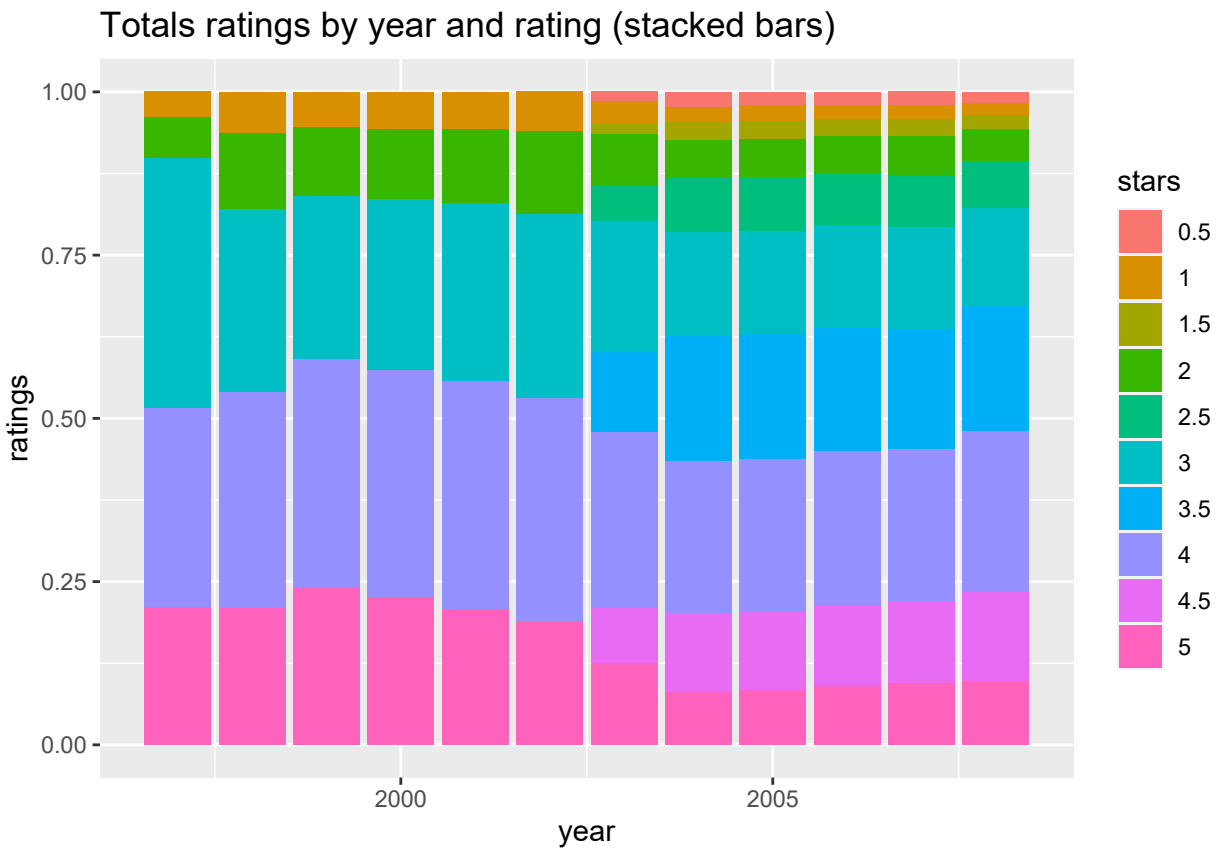
4.21 The evidence suggests that there is a recruitment effect, 6% of first reviews are of one of these five blockbusters, whilst these reviews only make up 2% of all reviews, 3.566 more likely than average. If we reduce the blockbuster threshold this becomes more marked. Further evidence is given by the fact that 10% of *users* have reviewed all five of these movies. This may have been part of an on-boarding process that asked *users* to rank familiar films.

Seasonality and the time component

4.22 In this section we will look at whether there appears to be seasonality in rating scores or the numbers of reviews. The chart below summarizes the distribution of *ratings* over the course of each year, with each line representing a year. These lines have been smoothed to look for patterns that repeat in subsequent years, such trends would represent seasonality. We start by looking at the number of reviews made in each year.



4.23 We notice that from 2001 the number of *ratings* in each year has been fairly steady, although with a bit of a dip in 2002 and a spike in 2005. Before 2001 the data is much more volatile perhaps as the *rating* platform gained users.



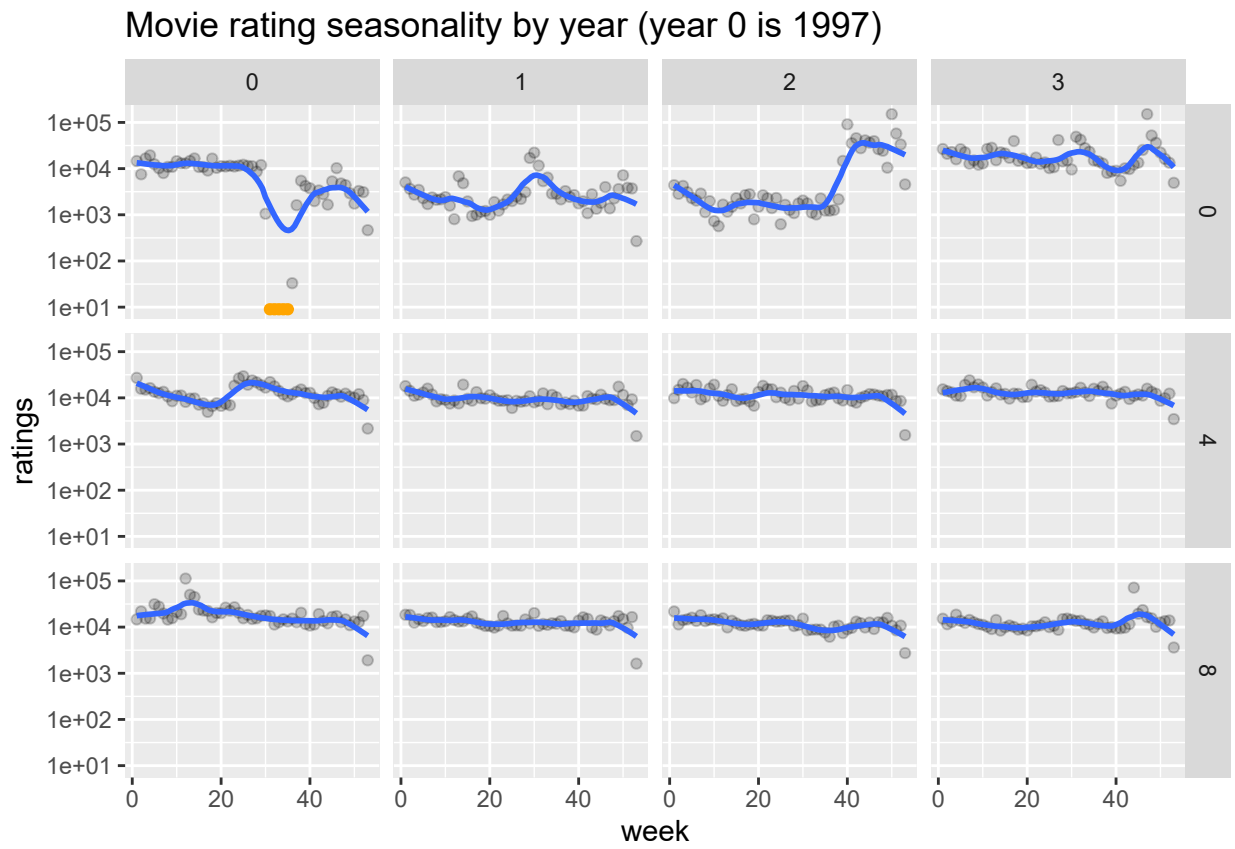
4.24 We saw earlier (in section 4.2) that half star *ratings* are less frequent than full star *ratings* in the data. This chart may go some way to explaining this. It appears that half stars were introduced mid way through the collection of the data, perhaps an attempt to boost popularity with a new feature after the decline in *ratings* seen from 2000-2002. The earliest half star review was recorded on 12 Feb 2003 .

4.25 Whilst looking at trends over time it was noticed that there were some weeks where the number of reviews was very low, further investigation identified some weeks missing from the data, perhaps when the system was taken down for maintenance.

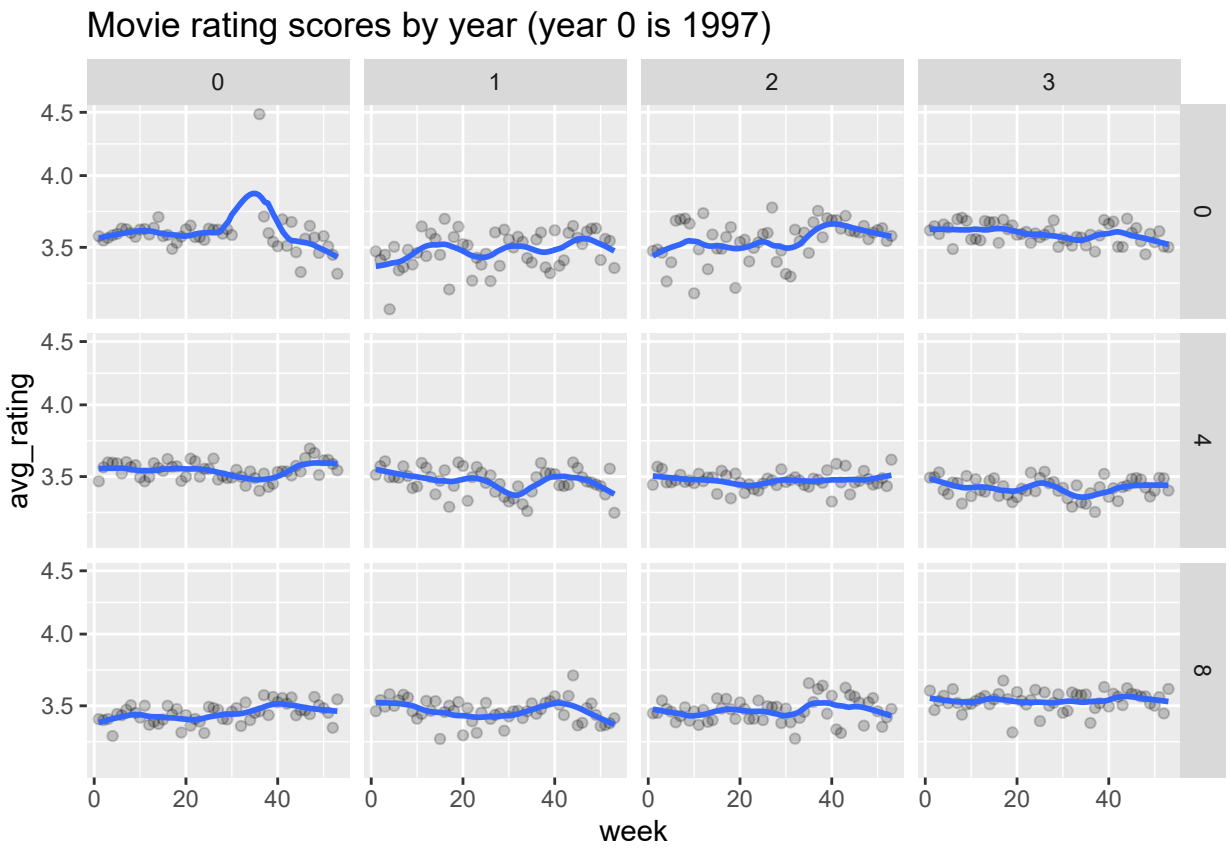
Table 6: Weeks with no ratings

year	week
1997	31
1997	32
1997	33
1997	34
1997	35

4.26 For illustrative purposes these points have been added back in to the weekly data and are highlighted in orange, in the charts below. These panelled charts (in which year 0 is 1997) allow the seasonal trends by year to be compared and also the changes between years.



4.27 The charts show a step change in the numbers of *ratings* in the last third of 1999, (top row, third chartlet,) when weekly *ratings* jump from fewer than one thousand per week to weekly figures in the tens of thousands. This correlates with the jump in blockbuster *ratings* in late 1999 that we saw earlier (4.13). Further, the volatility in the data in the early years seems to drop out from 2002 (second chartlet, second row) onwards. Note that week 53, the last observation in each chart is lower than other weeks because the last week of the year does not have a full seven days.



4.28 The lack of a clear repeating pattern across these lines suggests that there is not a clear seasonality in numbers of *movie ratings* posted each week, but that there are trends in the number of *movies* rated in a year. A similar chart, looking at the average *rating* by week also shows no clear trend; in this chart the lines for 2004 and 2005, for example, appear to be mirror opposites.

Half-stars

4.29 We digress briefly to consider the impact of the introduction of half scores on average ratings. Earlier (4.11) we looked at how *ratings* by *users* and *superusers* differed, and found that *users* gave a wider range of *ratings* and were generally more positive. We can repeat that analysis on partitions of the data before and after half scores were introduced.

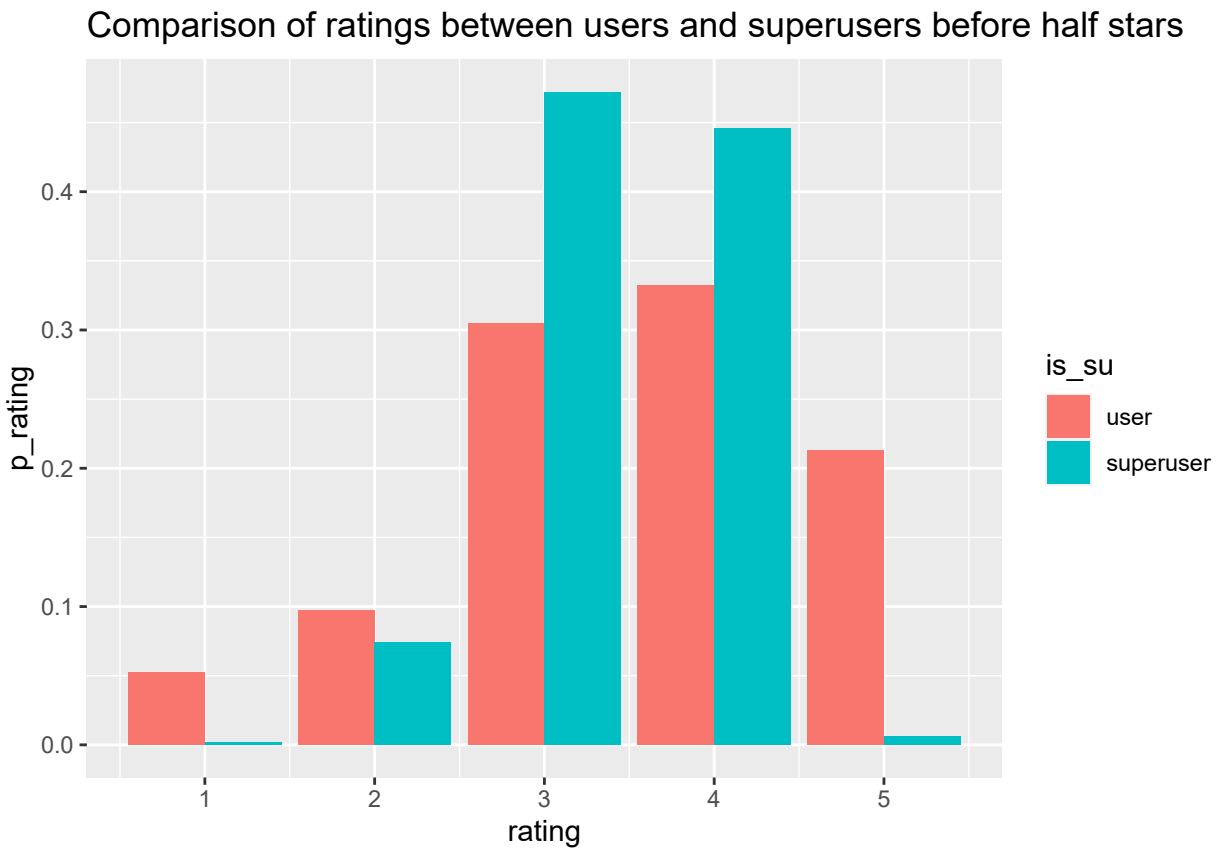


Table 7: Mean ratings by user type before half stars

user	superuser
3.56	3.38

4.30 The *superusers*, before the introduction of half stars, were very conservatively positive in their ratings, with nearly 92% of *ratings* being of three or four stars. Because *superusers* only very rarely found *movies* worthy of five stars their average *rating* is lower than that of less prolific users.

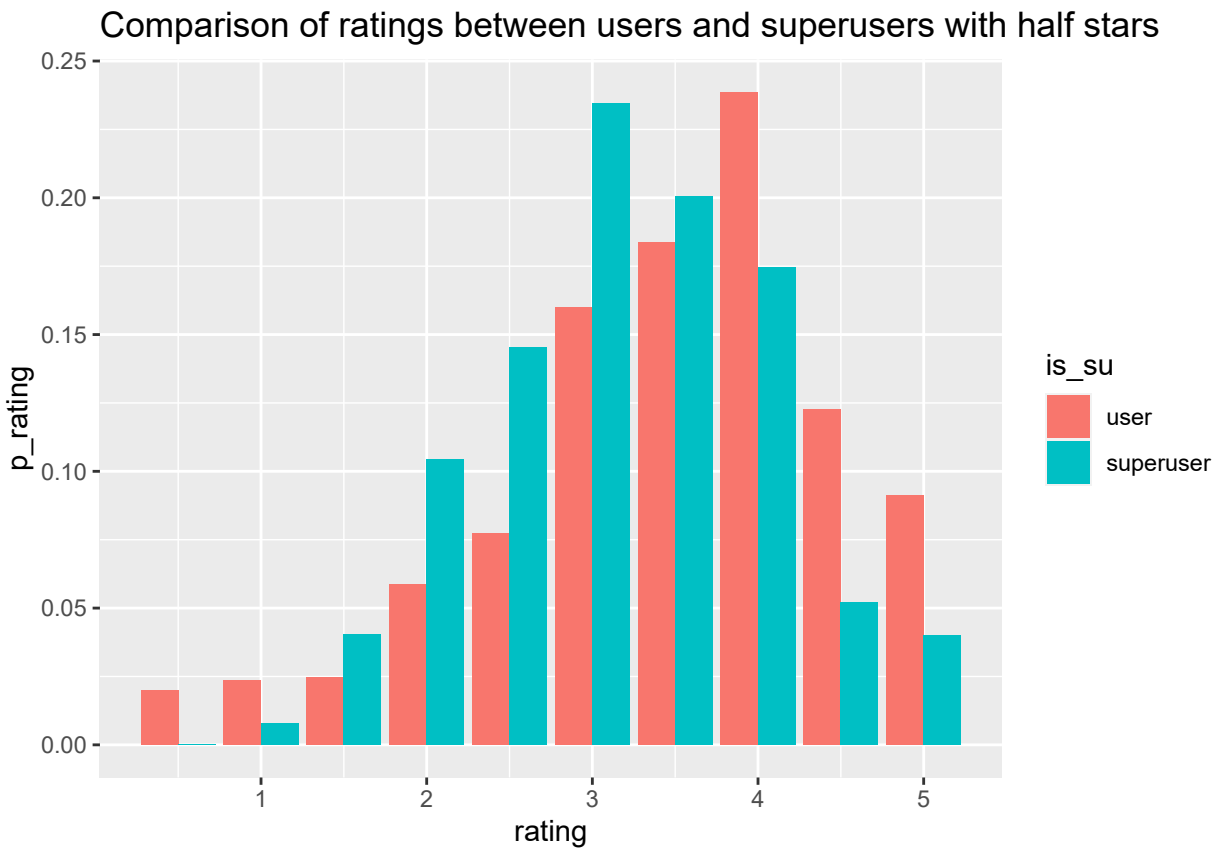


Table 8: Mean ratings by user type with half stars

user	superuser
3.47	3.18

4.31 With the introduction of half stars, *superuser ratings* became more nuanced and the average *rating* dropped by 0.2 points, this trend was also observed amongst general users, although the drop in *ratings* was less pronounced. There is still the risk, here, that the choice of *movies* reviewed, accounts for some of the difference. By restricting the reviews we analyze to the *movies* rated by the *superusers* we can account for this, and discover the impact on the mean *ratings* below.

Table 9: Mean ratings of common movies by user type before half stars

user	superuser
3.56	3.38

4.32 Mean *ratings* by *superusers* and *users* before the introduction of half stars with reviews restricted to the *movies* that *superusers* rated, above, and when half stars were available, below.

4.33 We see that controlling for *movie* choice does not make a significant difference, and because of this we have not reproduced the corresponding charts.

Table 10: Mean ratings of common movies by user type with half stars

user	superuser
3.48	3.18

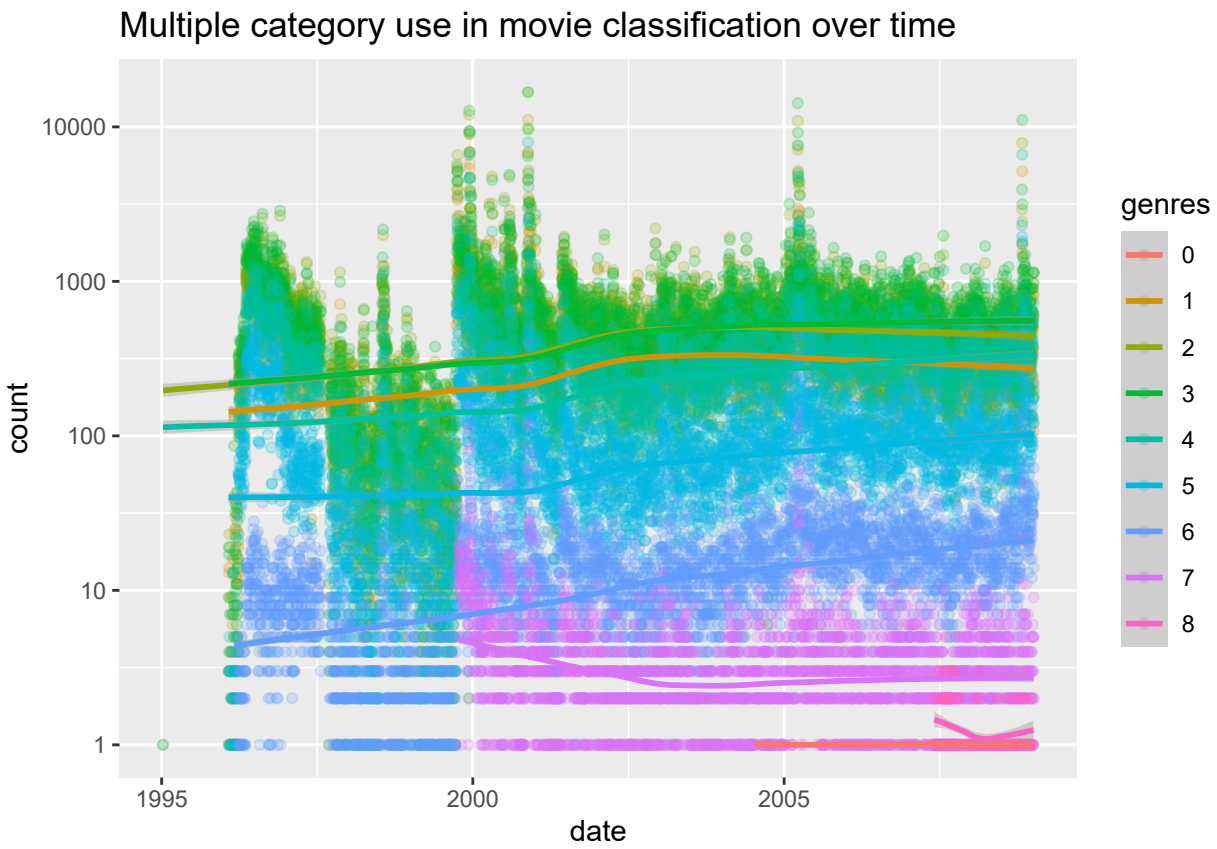
Genres

4.34 The reviews are classified by *genre*, this lists one or more categories that describes the type of *movie*, so we might expect a romantic comedy to be categorized as "Comedy|Romantic". From the constructor code for the `edx` dataset it appears that *genre* is set per *movie*, rather than varying per *user review*.

Table 11: Reviews by category

category	reviews
Action	31,612,380
Fantasy	3,910,127
Crime	3,540,930
Adventure	2,560,545
War	2,325,899
Animation	1,908,892
Sci.Fi	1,712,100
Thriller	1,341,183
Documentary	1,327,715
Film.Noir	925,637
Comedy	737,994
IMAX	691,485
Romance	568,332
Western	511,147
Children	467,168
Mystery	433,080
Horror	118,541
Drama	93,066
Musical	8,181

4.35 Note that because a *movie* can be assigned to more than one category the total sum of all *movies* across all categories is greater than the total number of movies: *movies* have been counted against each of the categories they are assigned to. Now we look at how the use of categories varies over time.



4.36 We can see that the anomalously low volumes in 1998, discussed in section 3.x, are visible here, we also notice that the number of categories in use has stayed broadly level over time, although the first seven-part catgeory was added in 1999, and the first eight-part category was added in 2007. It is natural to ask what part categories play, do they allow *users* to be more selective and does the number of categories therefore improve rankings? Are all categories equal and how do rankings of combinations correspond to rankings of individual categories?

4.37 If we are going to use genre-based categories in our prediction model we want to look at how discriminative the category is, the extent to which movies that are members of a category score differently to non-members. For a category to have a significant impact on our modelling we need it to have a large membership.

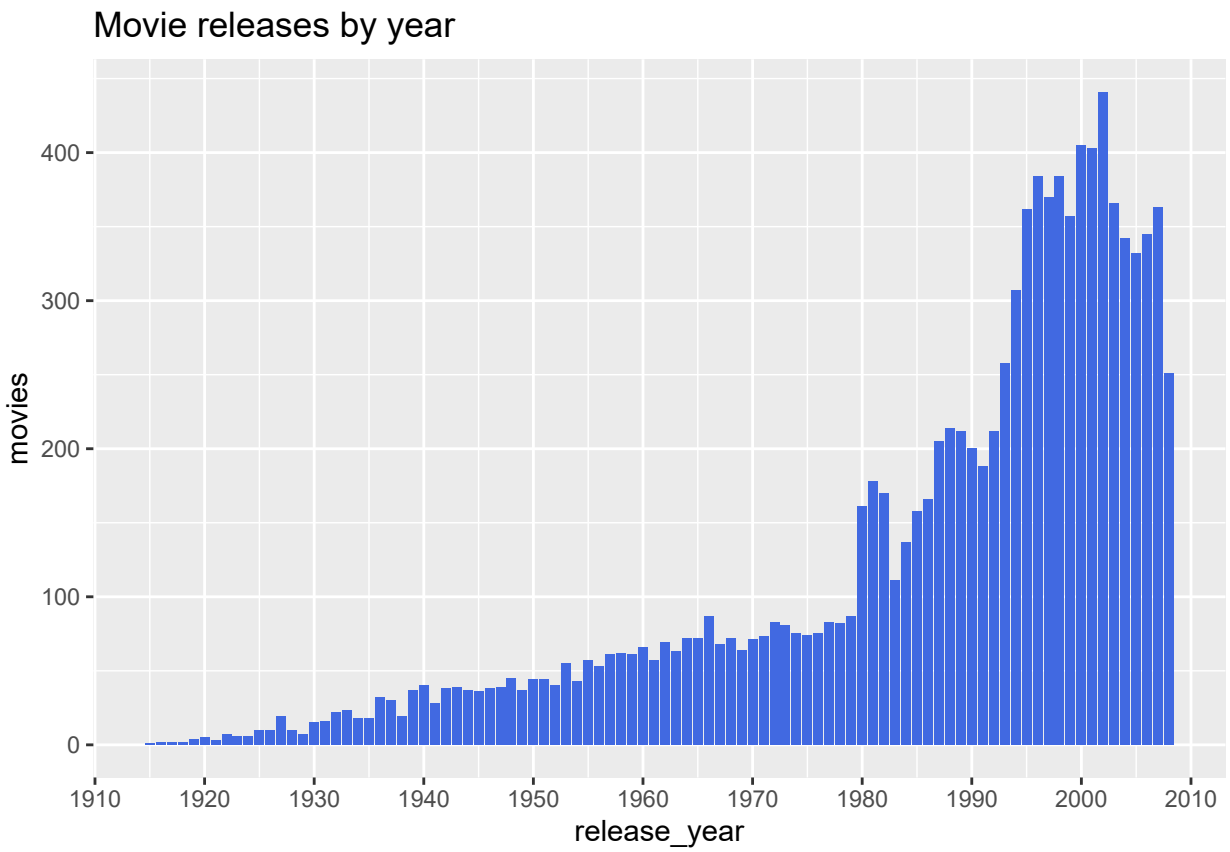
4.38 From this list it looks as if *Film Noir* should be the most predictive category, but notice that it is comparatively small. Of the larger categories *Drama* appears to be the most impactful. We haven't explored interactions in categories, for example do rom coms (Romantic Comedies) score more like Romance + Comedy.

Table 12: Category reviews

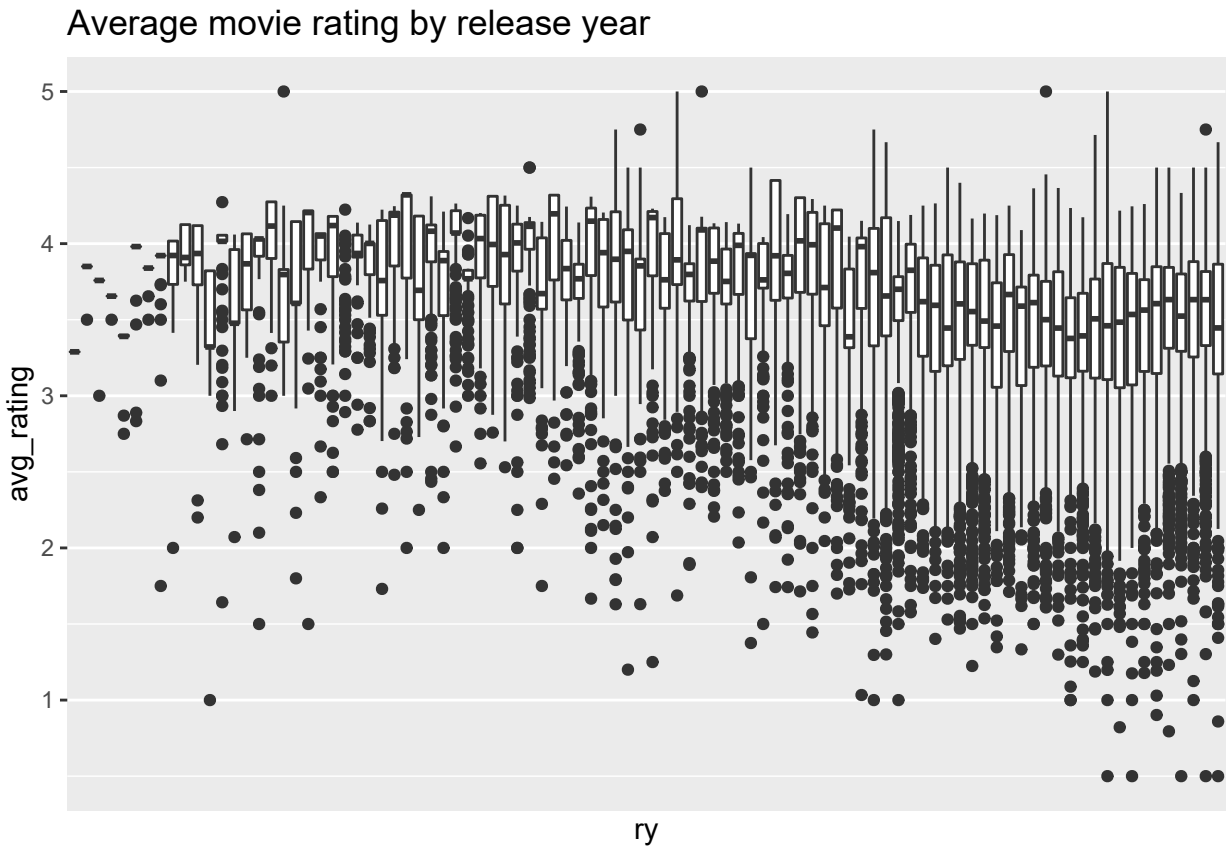
	cats	num	pos	neg	diff
10	Film.Noir	118541	4.01	3.51	0.506
18	War	511147	3.78	3.50	0.285
8	Drama	3910127	3.67	3.39	0.284
7	Documentary	93066	3.78	3.51	0.274
11	Horror	691485	3.27	3.53	-0.263
12	IMAX	8181	3.77	3.51	0.255
6	Crime	1327715	3.67	3.49	0.180
14	Mystery	568332	3.68	3.50	0.176
16	Sci.Fi	1341183	3.40	3.53	-0.137
1	Action	2560545	3.42	3.55	-0.127
5	Comedy	3540930	3.44	3.56	-0.125
4	Children	737994	3.42	3.52	-0.102
3	Animation	467168	3.60	3.51	0.093
13	Musical	433080	3.56	3.51	0.053
15	Romance	1712100	3.55	3.50	0.051
19	Western	189394	3.56	3.51	0.044
2	Adventure	1908892	3.49	3.52	-0.024
9	Fantasy	925637	3.50	3.51	-0.012
17	Thriller	2325899	3.51	3.51	-0.006

Titles

4.39 Movie titles are stored in the data as text, and the title text includes the release year, in parentheses, at the end. Our first step in processing the titles is therefore to remove the year; we will store it as a variable and investigate whether it has predictive power.



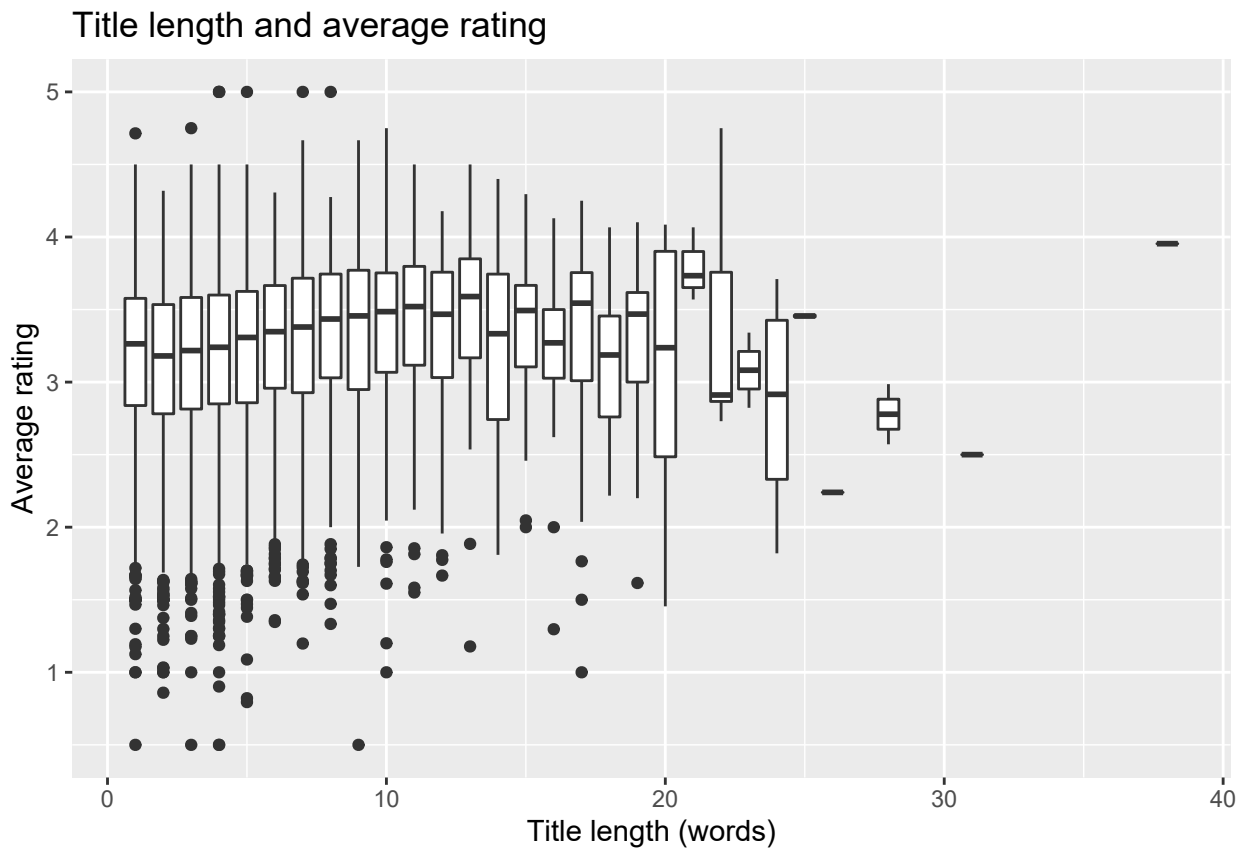
4.40 The number of releases rises gradually through the first decades of cinema before accelerating from the 1980s to the mid 1990s when numbers of releases levelled off. There is a marked drop in 2008, this may be due to the financial crisis of that year, or a censoring of releases that were near contemporaries of the compilation of the dataset.



4.41 The average *rating* by year is volatile, but, with a few exceptions, for most of the period has hovered around four stars. Films released in more recent years seem to attract more criticism and the average *ratings* trending towards 3.5 stars and the lowest *ratings* are lower.

4.42 Having extracted release year we are ready to start processing the titles. The hope in doing this is that titles influence viewers by setting expectations: that, in some sense, an effective title should be reflected by better than average reviews. Our hypothesis is that a title whose sentiment is aligned to its genre will fare relatively better. To investigate this we start by using similar string splitting techniques to those used for genres in section 4.34, above.

4.43 Reviewing the list of titles, having split the words, the longest appears to be *Every Man for Himself and God Against All a.k.a. The Enigma of Kaspar Hauser a.k.a. The Mystery of Kaspar Hauser: Jeder für sich und Gott gegen alle)*. This suggests the unanticipated presence of foreign language films in the sample, and makes us ask if we can we classify these effectively, either by language or binarily as English/non-English. We could potentially derive a number of variables from the title such as length and an analysis of n-grams, (groups of words within a title that do, or do not, often fit together,) whilst this might be valuable, this is beyond the scope of the present work. The chart below illustrates the idea: *movies* with two-word titles are not rated very highly, adding words to the title seems to improve *ratings* until titles become unwieldly beyond 13 words, above this the pattern is less obvious due to the scarcity of the data.


Title length and average rating
Average rating
5
4
3
2
1
0
10
20
30
40
Title length (words)

Sentiment analysis

4.44 We start our sentiment analysis by converting the words to lower case and excluding stop words. We will try using the Afinn lexicon which should give each word a score between -5 and 5, we will average these. This is a very simple choice of aggregate sentiment metric, under this meaure the most postively and negatively scored titles are listed below.

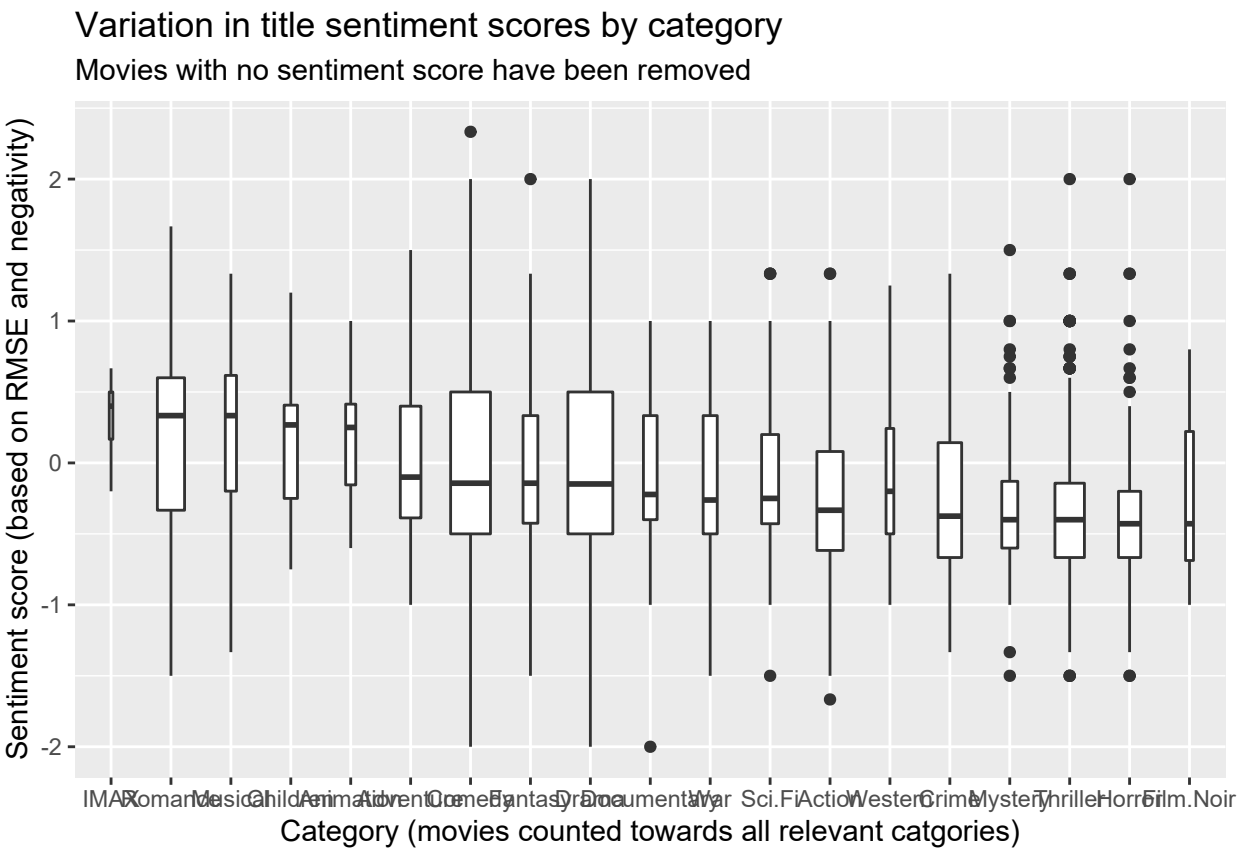
Table 13: High positive sentiment titles

release_year	title	sentiment	genres
2001	Lovely & Amazing	2.33	Comedy
1989	Worth Winning	2.00	Comedy
1994	Fun	2.00	Drama
2004	Miracle	2.00	Drama
2004	Godsend	2.00	Drama|Fantasy|Horror|Thriller
1949	Love Happy	2.00	Comedy

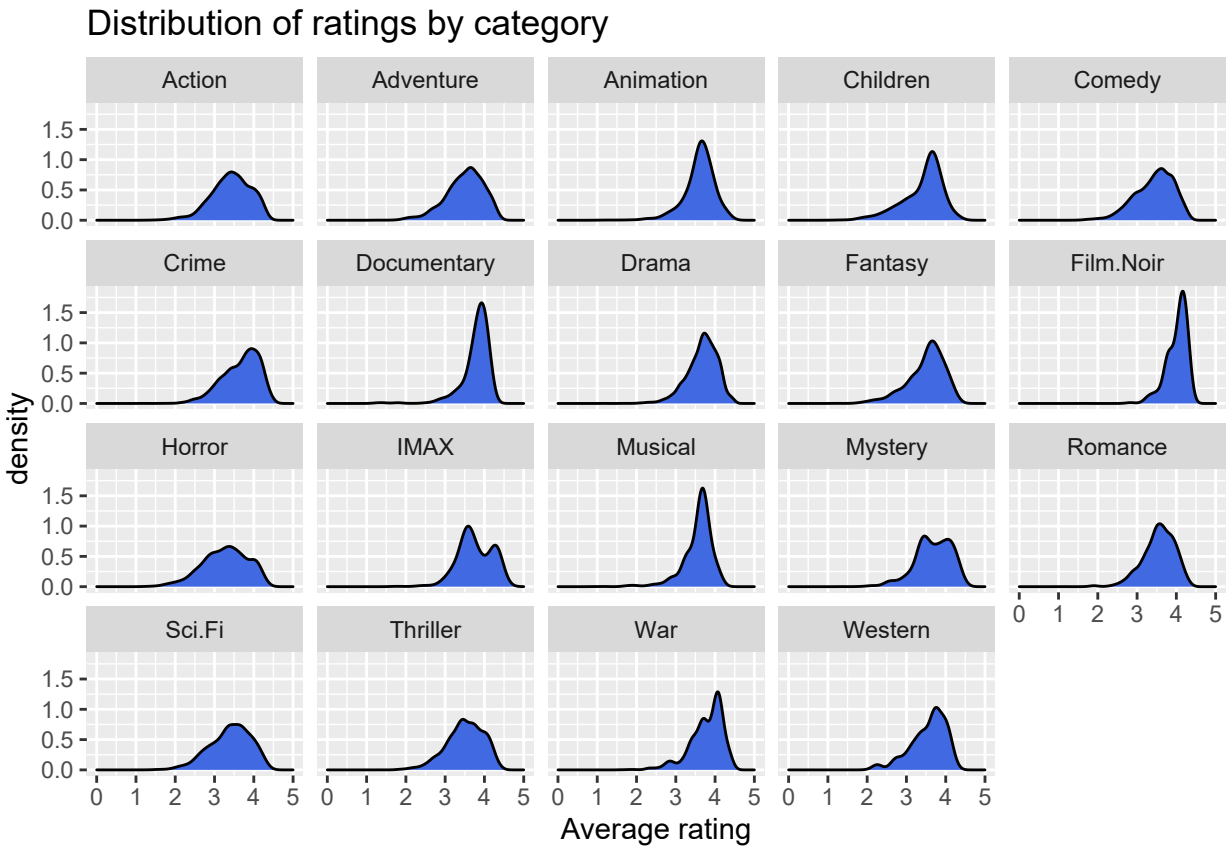
Table 14: High negative sentiment titles

release_year	title	sentiment	genres
1997	Liar Liar	-2.00	Comedy
1999	Dick	-2.00	Comedy
1991	Whore	-2.00	Drama
2005	Fuck	-2.00	Comedy|Documentary
1989	Disorganized Crime	-1.67	Action|Comedy
1996	Criminals	-1.50	Documentary

4.45 Now we have can look at relationships between sentiment scores and categories, we start by transforming our *movie* list from a row per *movie* with indicator variables by category, to a row by *movie* and category. We can use the union() function to deduplicate categories and movies, and then look at how sentiment scores vary across categories. These categories feature a *movie* if it has the category name in the genres list, this means an individual *movie* may be featured more than once in these counts.

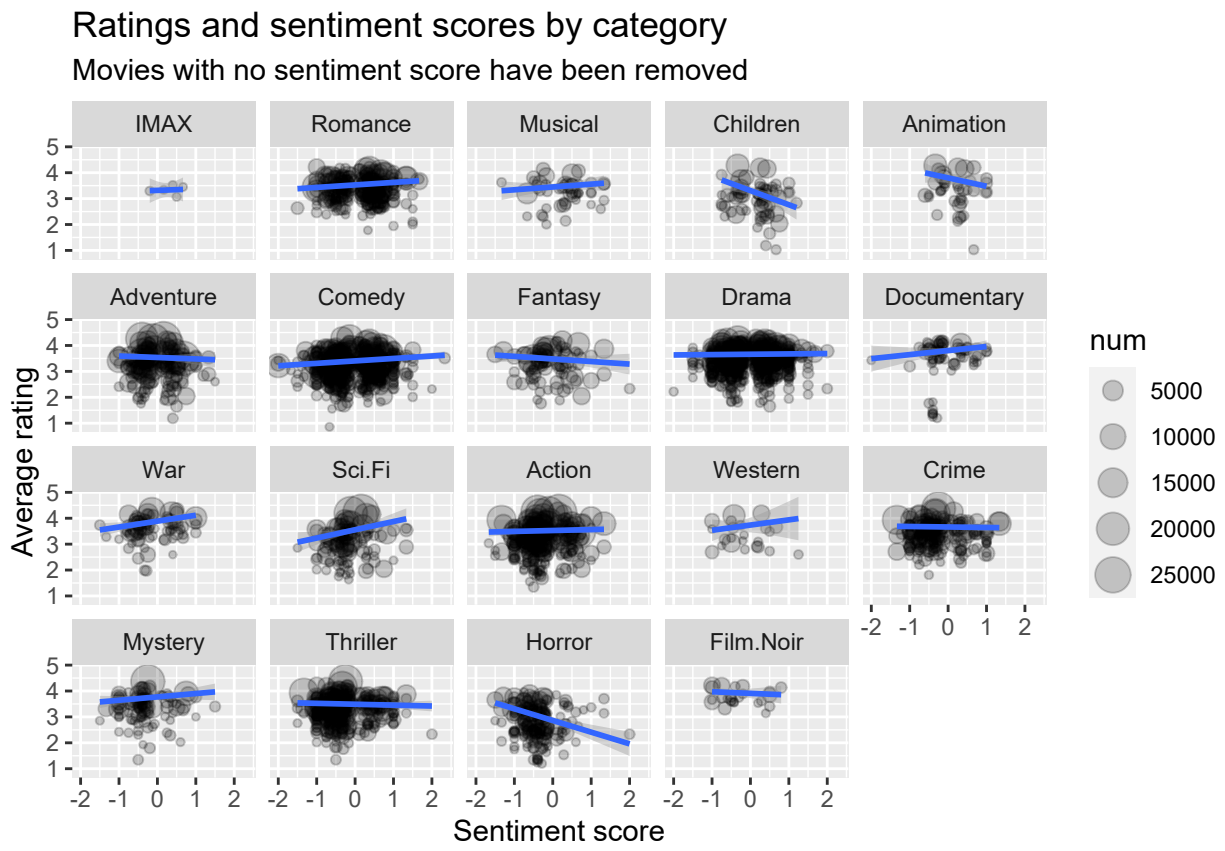


4.46 This distribution suggests a relationship between titles and categories as we might expect, the categories with the most positive sentiment scores are *Romance*, *Children*, *Musical* and *Animation*; whilst the most negative are *Horror*, *Mystery*, *Thriller*, and *Film Noir*. Do these relationships translate into patterns in *ratings* by category? Firstly lets look at how *ratings* are distributed by category.



4.47 These distributions are similar-but-different, reminiscent perhaps of ID shots of dolphin fins, and suggest that there may be differences in distribution of *ratings* across different categories. This would make sense since viewers may use these categories to select films they are more pre-disposed to enjoy, and this may, in part, explain why average *ratings* are above average: the reviews are average *ratings* across *movies* that viewers have watched, rather than average *ratings* across all movies. Notice that films in the categories *iMax*, *Mystery* and *War* appear to have two humps.

4.48 The chart below takes this a step further by looking at whether there are patterns in *ratings* and sentiment by category.



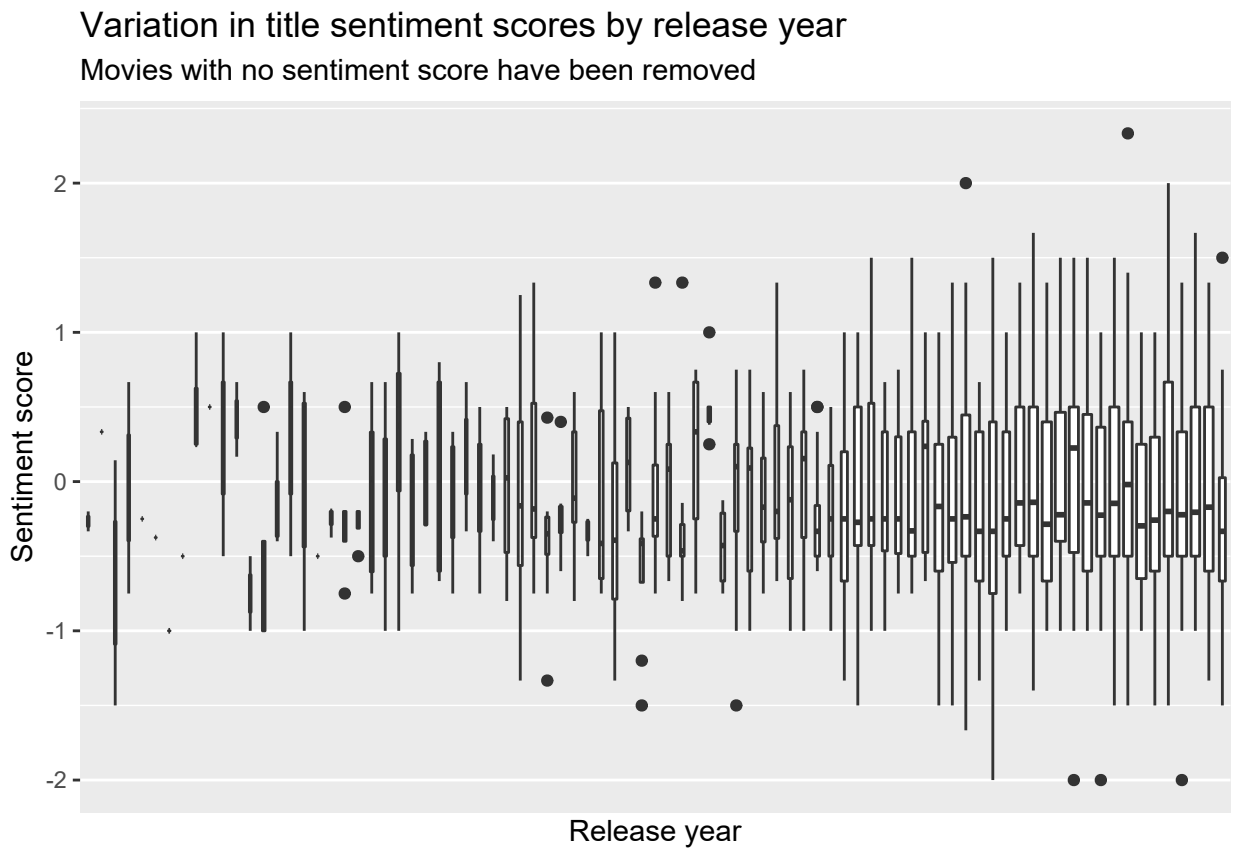
4.49 Here we notice negative relationships between sentiment and *rating* for *Children*, *Horror* and *Film Noir*, and positive correlations for *Sci-Fi*, *War*, *Mystery* and *Westerns*. There does appear to be some influence here from outliers, especially in smaller categories such as Horror, where the *movie Godsend* has a high sentiment score and low average rating; this film also appears as an outlier in the *Thriller* and *Fantasy* categories, it was classed as *Drama|Fantasy|Horror|Thriller*. The distributions for categories *Thriller*, *Sci-Fi* and *Horror* appear to be strongly weighted towards more negative sentiment titles, having most of their points to the left in the chartlets above.

4.50 Can we get sentiment scores for categories: do titles aligned with the category sentiment score more highly, for example? If we apply sentiment scoring we find that only a few categories get non-zero sentiments, and that the categories don't tally with the trends we saw earlier (see 4.49, for example), and that some categories, such as *Horror*, don't get sentiment scores when we might expect them to.

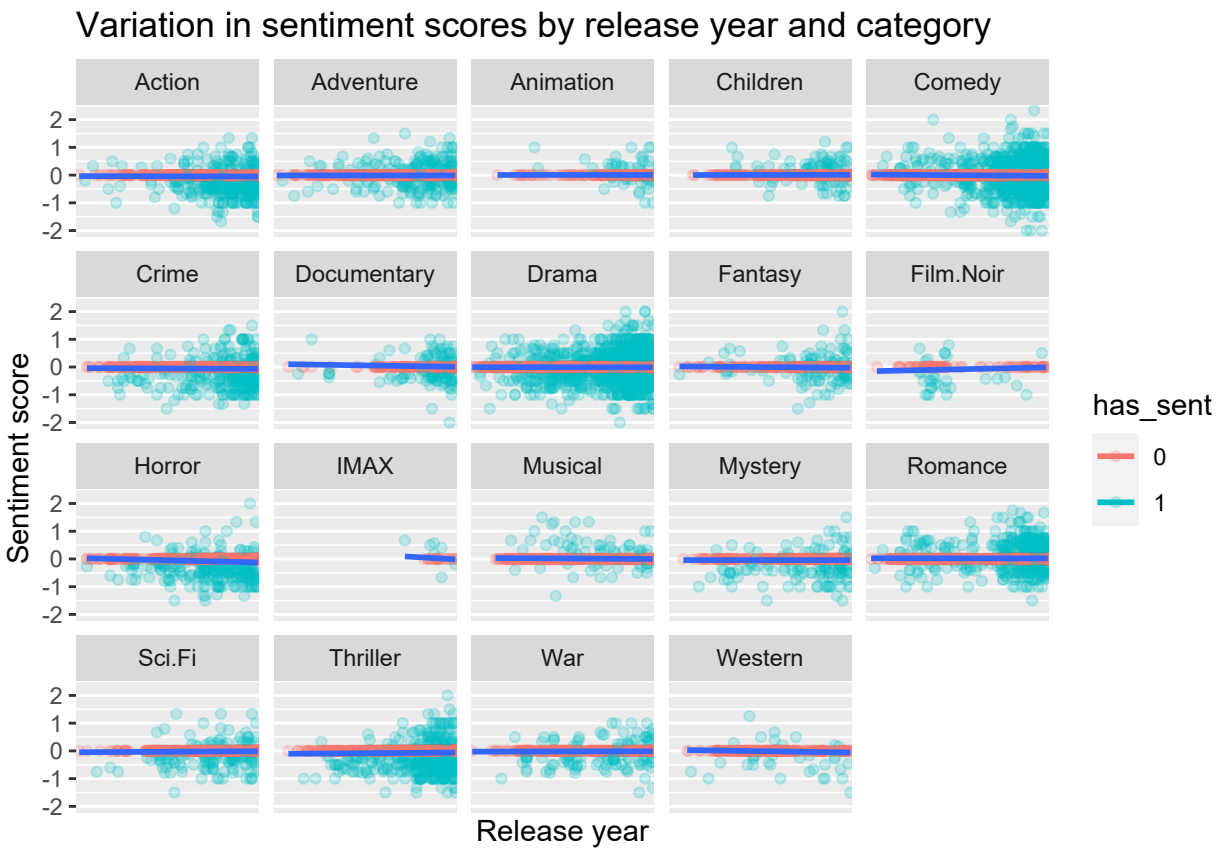
Table 15: Sentiment of categories

	word	value
1	adventure	2
4	romance	2
2	comedy	1
5	war	-2
3	crime	-3

4.51 The chart below illustrates how sentiment scores vary by release year, there is a tendency for titles that can be scored to generate negative sentiments, with a few years in which positive sentiment titles were more common.



4.52 There is no obvious trend in sentiment over time, it doesn't appear that *movie* titles are getting any more extreme. The risk here is that the movie mix may be confounding any trend. We can explore this by splitting the data into categories and repeating the analysis.



4.53 Having split the data by category we don't see any marked trends, this is in large part due to the weight of *movies* in the middle ground, with sentiment scores of zero, which exert an inertial braking effect. If we were to exclude these, and we'd be hard pushed to justify this, we might see *Crime* titles appear to be getting less negative, or the opposite for *Musicals*.

Summary of findings

4.54 From our exploratory data analysis we have found some relationships that might be nice to know, and some that should be useful in modeling ratings. We have seen

- the mean rating is 3.51 (section 4.2)
- half-star ratings were introduced on 12 Feb 2003 (4.29)
- there is variability in between user rating of movies (4.5)
- there is variability in between movie ratings (4.6)
- super-users appear to give lower scores than users (4.14) and be less extreme in scoring.
- blockbusters appear to have been rated as part of the recruitment of new members (4.20)
- different genre categories have different distributions of ratings (4.38)
- there is some relationship between sentiment scores and ratings by category (4.46)

5 Modeling methods

Modeling approach

5.1 We start by following the approach taken in the lecture on recommendation systems and fit a linear model based on the user and movie effects, we have validated the motivation for this approach in our analysis of *users* (4.5) and *movies* (4.6) and we have reassured ourselves that there is no significant confounding of these variables.

Model assessment

5.2 We will assess models based on the square root of the mean of the squared errors (RMSE), using the function provided in the lecture notes:

```
# define a function to calculate rmse between observed and predicted data
RMSE <- function(true_ratings, predicted_ratings){
    sqrt(mean((true_ratings - predicted_ratings)^2))
}
```

A model will be better if it has a smaller RMSE.

Model design

5.3 We have seen that a number of variables, both native and derived, in the data appear to influence ratings, so we will extend the linear movie recommendation model discussed in the course. This had the form:

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

where b_u is a user-specific effect, and b_i is a movie effect and $\epsilon_{u,i}$ is a Normal error term.

5.4 We extend this model to capture the predictive power of some of the other variables we have or can derive. In practice we test the predictive power of the model as we go along, here we will summarize the outcome of this process.

5.5 How can we capture trends in review ratings over time? In section 4.29 we saw that the introduction of half-star scores led to a step change in rating scores and so we select this as a variable to add to our model. This has a time component, so we will not use the timestamp, although this may have predictive value.

5.6 The movie titles include the release year and we saw, in 4.39, that this had an impact on ratings so we include this as a factor. We have included the character representation of the year as a factor, rather than as a numeric, since we want to capture the predictive power of the year without considering any linearity of relationship between release year and rating.

5.7 We saw, in section 4.38, that a number of categories had an impact on ratings, of these we add variables to the model for categories *Film Noir*, *Action*, *Crime*, *Comedy*, *Drama*. We could have added more category variables but there is a diminishing return in adding more variables to the model. These categories have been chosen because of their relative size and the predictive power they appear to have.

Validation hold-out

5.8 The source data has been segmented into training and validation segments, see 3.3 for details. We will develop and train the model using the full training dataset, holding out the validation dataset, which will only be used for validation. This separation also applies to the exploratory data analysis, which has only been performed on the training data.

5.9 Before testing, the validation data has been augmented with derived variables for half scores, based on the time stamp, and release year based on the title.

5.10 The goodness of fit of the model has been evaluated based on the RMSE returned when applying the model to the validation dataset.

Model improvements

5.11 Improvements could potentially be made to the model by adding further variables, for example for further categories, the review year, or based on sentiment analysis. Interaction terms between variables would most likely be of benefit, for example between categories, and the time variables. 5.12 We have only considered a linear model, this has the benefit of being simple to understand and fit, especially for a large dataset, but this shouldn't be a limiting factor and we could consider fitting a more sophisticated model such as a random forest, given enough computing power.

6 Results

Modelling results

6.1 The model discussed in the course produced an RMSE of 0.8653488, by extending this linear model we see very little improvement in the RMSE when we add a variable for the availability of half scores, however after adding indicator variables for selected categories we get and RMSE of 0.8652808, and achieve an RMSE score of 0.8649356 when release year is factored in.

Model performance

6.3 The number of variables fitted increases the time required to fit the model and produce predictions. Using a simple model fitting function like lm() function would be likely to be much slower than using slightly "clunkier" iterative code to fit to successive residuals. Increasing the number of variables and interaction terms would further slow down the model. Using modelling techniques that are not tuned for large datasets might produce very poor performance.

6.4 A typical recent run on a home PC (64 bit Windows 10 on an Intel i5 7400 @ 3.00 GHz with 16GB RAM) with the model took 1 minute, 17 seconds a few minutes, this is a fraction of the time taken to run the full modelling script, which is slower because it includes downloading and processing the source data, rather than to the actual model fitting.

7 Conclusion

Summary

7.1 We have looked in some depth at the MovieLens data and this exploratory data analysis has allowed us to see some ways to extend and improve the recommendation model. The final model we present here is a linear model with variables for category, the existence of half scores, and the release year. This is a still a relatively small and simple model and there is scope to extend this model in terms of complexity to get a better fitting model.

Limitations

7.2 The model works very well at predicting the rating scores for known movies by known users, this is valuable but may limit its use when the user and/or the movie is unkwown. Suppose we wanted to use the data to predict the likely rating for a new movie, we could do this by omitting the movie parameter b_i from the equation, if we try this we get a much worse fit, with an RMSE of 0.991. Similarly if we omit the user parameter, b_u, we get an RMSE of 0.944.

7.3 Once we had reviews for new movies we could refit the model based on this new data. This is potentially time consuming as we would need to reload and rerun the modelling steps. It would be easier if we could update the parameters without the need to rerun the entire model. Further it would be nice if we could establish some certainty over some parameters, so that we could rely on the effect of a category being relatively stable over time - there is the potential that the sign of a parameter could chnge in refitting because there are no constraints in the model.

Further work

7.3 As pointed to in the limitations section there could be value in developing a predictive model that was tailored to handling unseen data. This would likely want to consider trends in ratings over time and focus more attention on categories and interactions between categories.

7.4 The sophistication in modelling users could also be improved from the current model that effectively assigns each user a positive or negative score, to assigning the user a score in different categories, for example, interactions between user and category.

7.5 There is potentially value in exploring the sequencing of reviews, especially if this reflects the order in which users view movies: one can imagine that the order in which one watches a series might influence the user's enjoyment, being able to follow a story arc, for example.

7.6 The MovieLens data scratches the surface of what could be analysed in relation to films. There is an enormous amount of data that could be associated with users and movies that we don't have:

- could gender play a role in movie rating, would this vary across categories?
- does the star quality of some actors translate into better reviews?
- does nationality or language influence ratings for some users?
- do people enjoy films more when they watch them alone?
- how do MovieLens ratings compare to other platforms?
- do some directors produce better or worse movies?
- do bigger budgets translate into better reviews?
- does popcorn improve ratings, salt or sweet?
- do longer films get better reviews?