PH125.9x Capstone Part 1 - MovieLens Project

Jonathan Shiell

Files also at https://github.com/JonathanShiell/PH125.9x-Capstone-1

Introduction

An Introduction to the Dataset

For the purposes of this project, the dataset is divided into a training set `edx` and a test set `validation`. All values of `userId` and `movieId` in the test set `validation` are contained in the training set `edx`. These were prepared using code supplied by the edx PH125.9x course website.

The dataset being used is the MovieLens 10M dataset, provided by GroupLens . It features a total of approximately ten million unique ratings, each of which are considered to be an observation.

Variable Name	Description
`userId`	Unique, anonymised user identifier (as integer)
`movieId`	Unique movie identifier (as integer)
`rating`	Rating score given, from 0.5 to 5.0 in increments of 0.5
`timestamp`	Timestamp at the time that the rating was given (as integer)
`title`	Movie Title including year in brackets at end.
`genres`	Genres of Movie, separated by `|` for multiple genres

Observations are stored row-wise, in accordance with the 'tidy' principles proposed by Wickham (2014). Let us consider the first six items of the training set `edx`:

userId	movieId	rating	timestamp	title	genres
1	122	5	838985046	Boomerang (1992)	Comedy|Romance
1	185	5	838983525	Net, The (1995)	Action|Crime|Thriller
1	292	5	838983421	Outbreak (1995)	Action|Drama|Sci-Fi|Th...
1	316	5	838983392	Stargate (1994)	Action|Adventure|Sci-Fi
1	329	5	838983392	Star Trek: Generations (1994)	Action|Adventure|Drama...
1	355	5	838984474	Flintstones, The (1994)	Children|Comedy|Fantasy

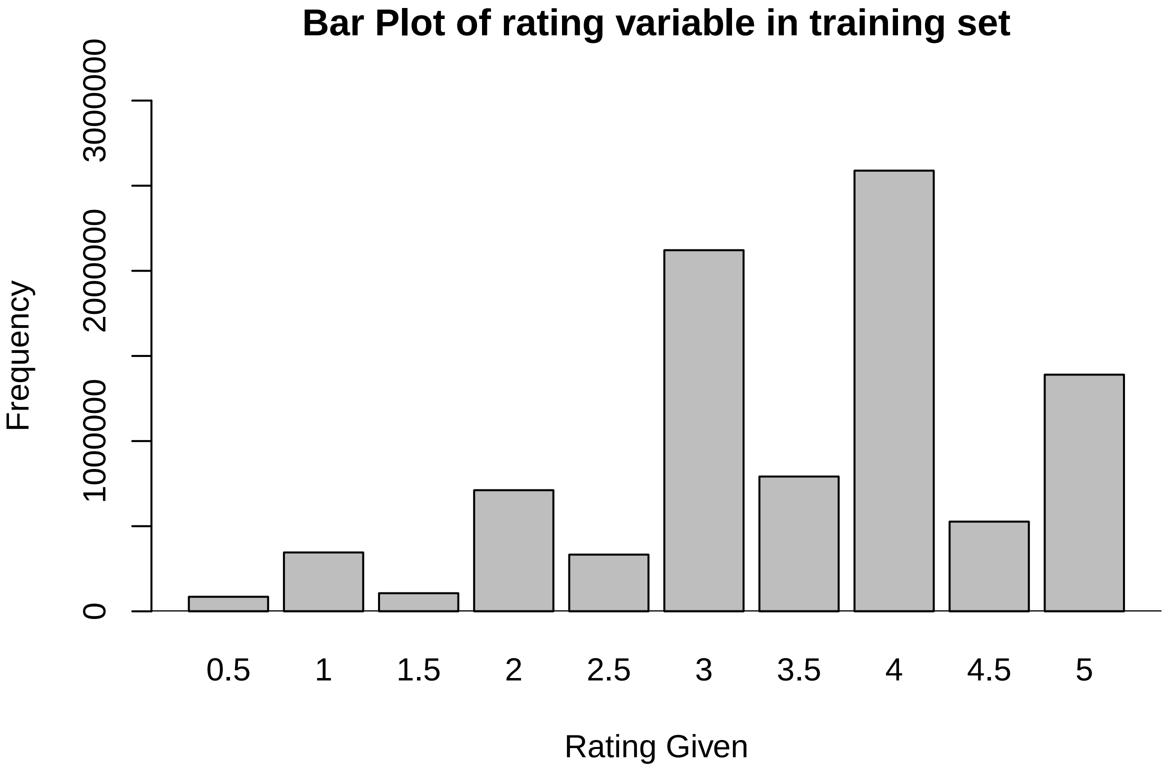
The dataset consists of a total of 10000054 items, with 9000055 items in the training set `edx` and 999999 items in the test set `validation`. There are a total of 10000054 items in the combined dataset.

Dependent Variable (Rating, Training Set only)

The dependent variable `rating` in the training set has a minimum of 0.5, a maximum of 5, a mean of 3.5124652 and a standard deviation of 1.0603314. The variable `mu` (population mean μ of the training set) will be retained for use later. The frequencies of each rating score given are as follows:

Rating	Frequency of Rating
0.5	85374
1	345679
1.5	106426
2	711422
2.5	333010
3	2121240
3.5	791624
4	2588430
4.5	526736
5	1390114

This may also be plotted on a bar chart:



Independent Variables (Training Set only)

Number of unique movieId values	10677
Number of unique userId values	69878
Number of unique timestamps	6519590
Number of unique titles	10676
Number of unique genre combinations	797

The independent variables, including integers, are factors that are not ranked, and therefore are free to be assigned any value. In particular, it may be observed that there are 10677 unique movies and 69878 unique users. This means that there are 746087406 possible combinations. Given that there are 9000055 total items in the training set, there are approximately 83 times more unique movie/user pairs than in the training set.

We may derive additional information from the `genre` vector. Genres are separated by the pipe character `|`; there are 20 total unique genres described. In the GroupLens published description, they are described as: Action, Adventure, Animation, Children, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War and Western. In addition, the elements `IMAX` and `(no genres listed)` are present.

Introduction to Methods Used

Estimation of Movie and User Bias

Bias is considered to be the preference shown on average across, for a particular movie (described as b_i) and by users after movie bias has been taken into account (b_u). The simplest method of determining movie bias is using the formula:

$$b_i = \frac{\sum_i (y_i - \mu)}{n_i}$$

where y is the individual rating given to a movie by an individual user, Y is the vector of all items of y that may be indexed by user and/or movie identification number, μ is the population mean over the training set and n_i is the number of ratings received by that movie. This is applied movie-wise, and may be simplified for computation purposes by calculating the grouped mean of $(Y_i - \mu)$ for a particular movie if regularisation is not required. This is based on the following model

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

where $\epsilon_{u,i}$ is the error function, i.e. the difference between an observed value of the dependent variable and the relevant predicted value that is explained by the known independent values by application to the model of interest.

A similar formula may be used to determine user bias, if required, as a second step:

$$b_u = \frac{\sum_{u,i} (y_u - (\mu + b_i))}{n_{user}}$$

where Y_u is the rating given by a particular user to a specific movie, b_u is the user bias, n_{user} is the number of ratings per user and all other symbols are as per the previous equation. This is applied user-wise, using different values of b_i based upon each movie as appropriate, and may be simplified as the mean of $Y_u - (\mu + b_i)$ for a particular user if regularisation is not required. This is based on the following model:

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

Biases are applied in the following, regression-like manner, for movie bias effect only and for movie and user bias effect respectively, in order to predict the rating for a specified movie. These are applied to the test set, by rating event, in order to obtain predicted values as follows:

$$\hat{y}_i = \mu + b_i$$

$$\hat{y}_{u,i} = \mu + b_i + b_u$$

Measurement of Model Error

Performance will be measured by the use of root mean standard error (RMSE). This is determined as the square root of the mean of the square of the difference between an observed value and that predicted for the training set, i.e.

$$RMSE = \sqrt{\frac{1}{N} \sum (y - \hat{y})^2}$$

where y represents an observed value in the test set, $\hat{y}$ represents the corresponding expected value, and N represents the number of items being considered. This may be simplified to the square root of the mean of $(Y - \hat{Y})^2$ for computation purposes, where Y and $\hat{Y}$ are the vectors of y and the aligned corresponding values of $\hat{y}$ respectively.

This may be computed in R using function such as:

```
RMSE <- function(true_ratings, predicted_ratings){
  sqrt(mean((true_ratings - predicted_ratings)^2))
}
```

This uses R's in-built vector arithmetic capabilities. *Source: Irizarry(2019)*

A lower value of RMSE represents a dataset that, on average, is more accurate. This method places a greater emphasis on outliers than on other measures of total error, and therefore penalises larger variations between observed and expected values to a greater extent than smaller variations.

Error minimisation for bias effects

Error in a model is defined by the difference between an observed value and that which would be predicted by a model. For any given value y and a prediction $\hat{y}$, the error ϵ for a single observation is described as follows:

$$\epsilon = y - \hat{y}$$

The error function is often adapted for optimisation purposes; for instance, on this occasion I the RMSE method above returns the square root of the sum of ϵ^2 for all observations against their predicted values in the training set.

The main independent variables, `userId` and `movieId`, are treated as categorical factors, even though they are integers. A regularisation technique is applied, by using values of λ (lambda). The regularisation formula (described as penalisation in Irizarry(2019)) for movie bias b_i at a given value of λ is as follows:

$$b_i(\lambda) = \frac{1}{\lambda + n} \sum (Y_i - \mu)$$

This is computed by grouping by each movie using the `group_by()` command. It is not amenable to simplification by the computation of means.

This value is chosen by iterating through various values of lambda and obtaining the lowest RMSE. This is therefore a 'least squares' approach. A similar method is used for user bias b_u at a given λ :

$$b_u(\lambda) = \frac{1}{\lambda + n_{user}} \sum (Y_i - (\mu + b_i(\lambda)))$$

This is computed by, having previously calculated all required values of b_i, mapping to the relevant whole training dataset by movie and grouping by each user using the `group_by()` command. It is computed user-wise, using different values of b_u as appropriate. It is not amenable to simplification by the computation of means. When determining the value of λ for such a calculation, it is generally more efficient to use the same value of λ for both movie and user regularisation.

Determination of Predicted values

Predicted values are determined as follows, for movie bias only and movie and user bias respectively:

$$\hat{Y} = \mu + b_i$$

$$\hat{Y} = \mu + b_i + b_u$$

Where the values for user bias are either with or without regularisation. These values of $\hat{Y}$ may be used in order to evaluate the model using the RMSE function with Y as the observed rating and $\hat{Y}$ being the predicted rating using the relevant bias factors (b_i only or b_i and b_u as appropriate). The values of error ϵ may be determined by a method based on the differences between Y and $\hat{Y}$ as described above.

Aims of the Project

- To determine how much of the variation is determined by the user and movie bias effects.
- To develop a model that may be used as is to make recommendations based on the highest ten regularised movie biases.

The second objective would be fulfilled by the use of a table of the highest ten regularised movie biases, with additional information provided in the object-related model in order to provide for the possibility of filtering in order to customise the output to match the preferences of a particular user.

Methods

Packages used

- `tidyverse` (including `dplyr` and `ggplot2` functions)
- `knitr` was used to prepare tables (including calls to the `kable` function).
- A call to `dslabs::ds_theme_set()` was used in order to standardise the theme of `ggplot2` plots.

Code is adapted from Irizarry(2019).

Preparation etc. of Data

The data was reviewed by use of the `head()` function as in the introduction above. The data was observed to be in a format one row per observation and is therefore consistent with the concept of tidy data. Additional transformations were not applied at the exploratory stage, however the number of ratings given by each user and received by each movie were determined in the summary stages.

Analysis of Data

The dependent variable chosen was the rating. The measure of performance chosen was that of RMSE using the formula described in the introduction. The initial independent factors chosen were `movieId` and `userId` in unmodified form. For non-regularised values, the data was grouped by the relevant factor using the `group_by()` function with the relevant factor and any associated information to be retained. and a mean of differences from `mu` (population mean μ of the training set, computed using the code `mu <- mean(edx$rating)`) and previous predicted factors, determined using the `summary()` function. RMSE as described in the introduction was used as the measure of performance. The sections of code used are reproduced in the results section below along with their output.

Inital Development Comparison of Models

A model using `mu` as the prediction for every value in the training set was made so as to compare the performance of other models. Any model whose RMSE exceeds this value is liable to be considered unsatisfactory. The RMSE from this model was stored as `simple_average`.

Two non-regularised models were made, the first based on `movieId` only, and the second based on `movieId` and `userId`. RMSE was determined for each model and stored for later comparison. Where other information was retained, the `ungroup()` function was also used in order to prevent difficulty when making further use of the objects. These are referred to as the 'basic' models, and are stored as `basic_movie_only` and `basic_movie_user` respectively.

Application of Regularisation and further Comparison

Regularisation methods were applied as described in the introduction. The first regularisation was for movie effects only, and the value of lambda that yields the lowest predicted RMSE from the training set was determined by iteration and stored as `lambda_movie`. A second model involved determining by iteration the value of lambda that minimises RMSE for both movie and user biases, this value of lambda was stored as `lambda_movie_user`. These models were reproduced at the relevant value of λ , and the RMSES from these models stored as `reg_movie_only` and `reg_movie_user` respectively.

These models were repeated using the values of lambda determined above, and used for formal determination of RMSE and additional tests of model performance. Additional tests involved the comparison of biases determined as above, plotting density charts of movie and user biases. The final model was determined by the overall lowest RMSE as the primary criterion, and verified by the use of other measures recorded in further analysis below.

Further analysis of Model Performance

In addition, tables of the ‘highest ten’ and ‘lowest ten’ ratings were derived for movie biases, both the non-regularised model and also any values of lambda (λ) accepted as optimising RMSE for a given set of bias effects. These include the number of ratings per movie, and were used to provide an additional assessment of model performance, on the basis that a movie should not appear with relatively few ratings. Local truncation was applied to the final tables using calls to `str_trunc`; this was performed table-wise immediately before display in order to leave the underlying objects intact.

Model Finalisation

The performance of the models was reviewed, in terms of the following outputs:

- Mimimum RMSE by tuning criteria
- Usefulness of predictions (‘highest ten’ movies)
- Consideration of other factors (regression plots between ratings per movie and movie bias and ‘lowest ten’ movies) for each model.

An additional script was prepared so as to reproduce the predicted ratings and RMSE of the final model.

Methods of Plotting

Plots were made of various input and output variables. Histograms of numbers of ratings per movie and ratings per user, and regression plots between number of ratings received and movie bias were made using the `ggplot2` package. Other plots, including a barplot, lambda vs RMSE scatterplots and density plots were made using base R plotting commands. Additional alterations were made so all plots as to optimise the legibility of the plots.

For the regression plots, the locally estimated scatterplot smoothing (LOESS) method was used throughout.

Results

Computation of Basic (non-Regularised) Biases

Let us obtain the basic (non-regularised) movie biases; this will also provide counts of ratings received by each movie (`n_movie`) . These are stored as columns in the objects `movie_avgs` and `user_avgs` respectively. These form the basis of the basic (i.e. non-regularised) movie bias effect model and movie and user bias effect model.

```r
#Obtain a basic (non-regularised) version of the movie bias effects.
movie_avgs <- edx %>%
  group_by(movieId,title,genres) %>%
  summarize(b_i_basic = mean(rating - mu),n_movie = n()) %>%
  ungroup()
```

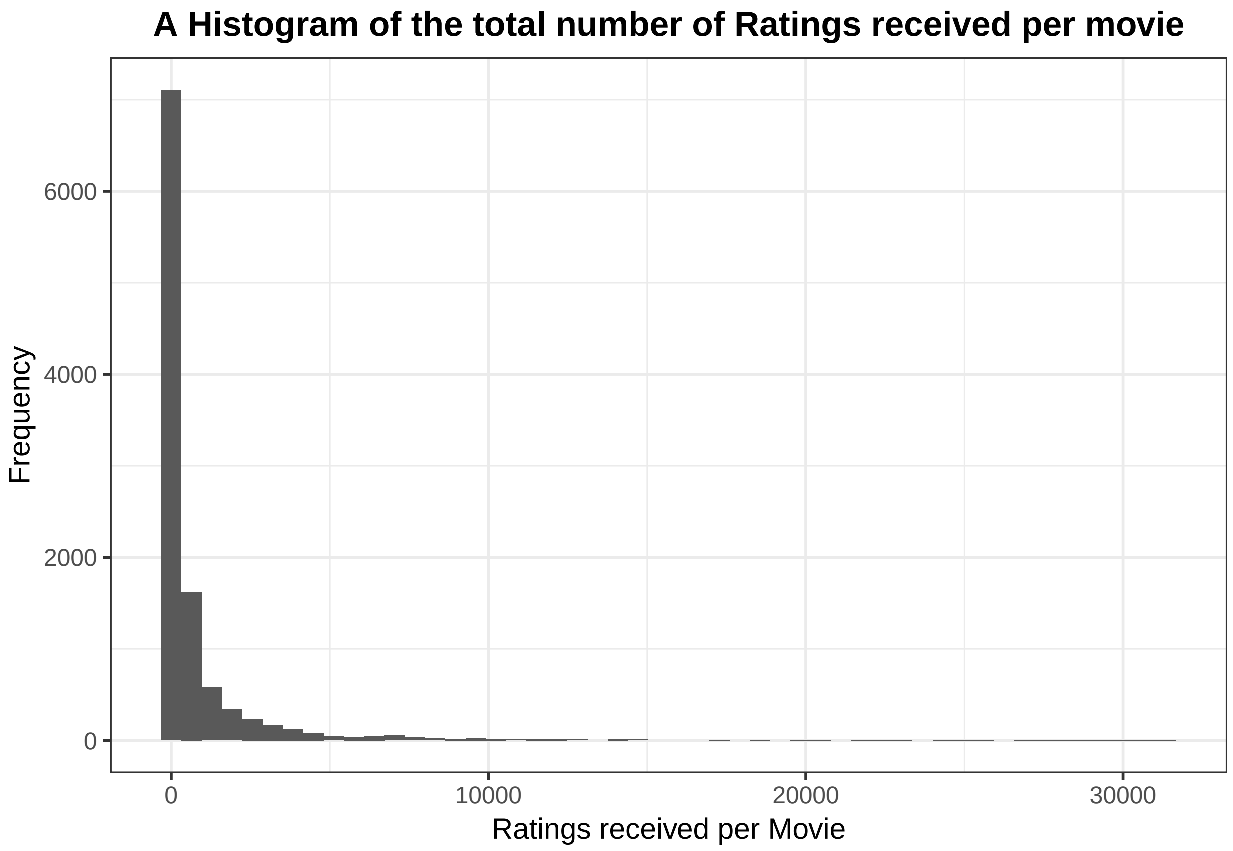
Let us now repeat this for basic user biases, and also obtain counts of the number of ratings given by each unique user(`n_user`)

```r
#Let us repeat this for user averages (after basic movie effects)
user_avgs <- edx %>%
  left_join(movie_avgs, by='movieId') %>%
  group_by(userId) %>%
  summarize(b_u_basic = mean(rating - mu - b_i_basic),n_user = n())
```

Additional Analysis of Variables based on Above Computations

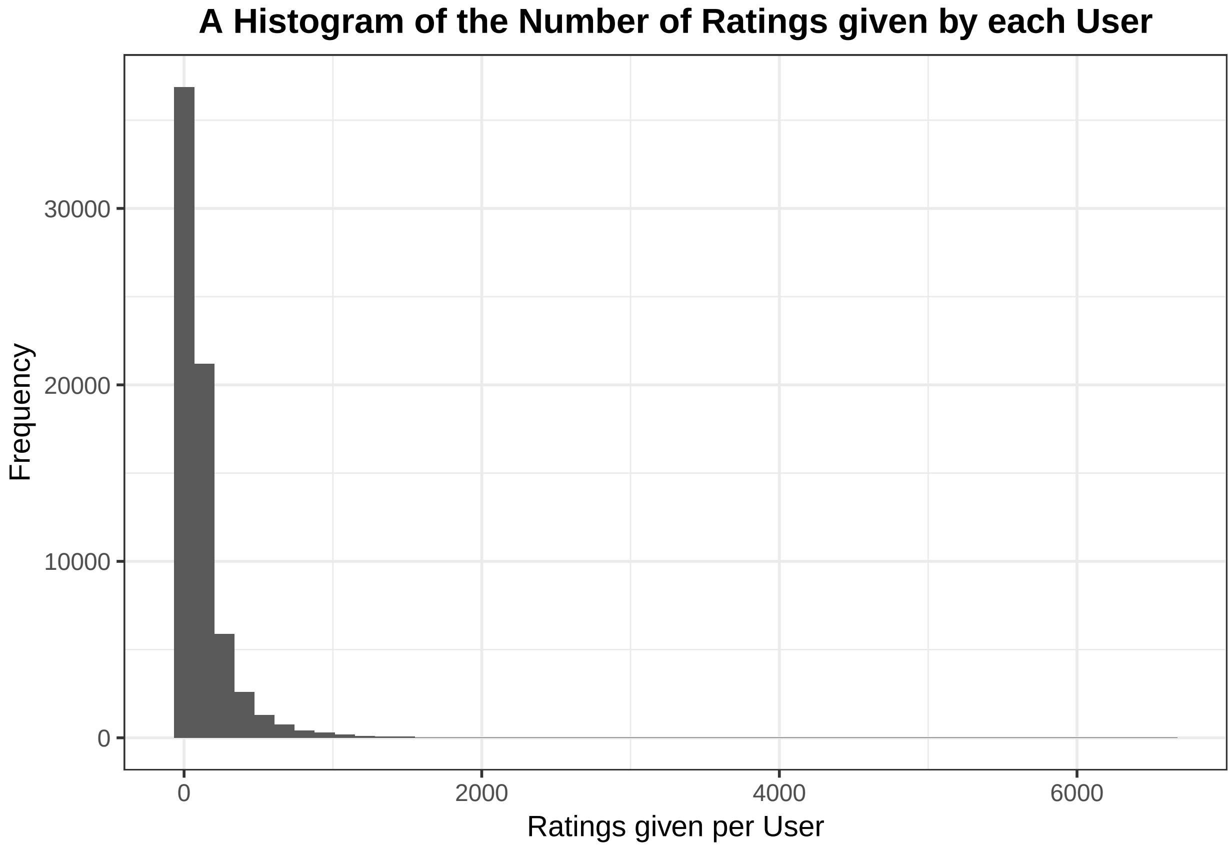
Let us consider the contents of `n_movie`, a vector that describes the number of ratings that a particular movie has received.

n_movie
Min. : 1.0
1st Qu.: 30.0
Median : 122.0
Mean : 842.9
3rd Qu.: 565.0
Max. :31362.0



Let us consider the contents `n_user`, the vector of number of movies rated by each user.

n__user
Min. : 10.0
1st Qu.: 32.0
Median : 62.0
Mean : 128.8
3rd Qu.: 141.0
Max. :6616.0



Basic (non-Regularised) Effects Models

We will use the the RMSE (Root Mean Squared Error) function from Irizarry (2019) to determine the accuracy of the model; lower values represent less total errors and therefore a better model. The simplest model is where every prediction is the training set population mean $\hat{y} = \bar{\mu}$, which is relatively poor but gives a baseline against which to compare other models. Only models with a lower RMSE (and therefore less total error) will be accepted. We will use the the RMSE (Root Mean Squared Error) function from Irizarry (2019) as described in the Introduction above:

```
#A simple average
simple_average <- RMSE(validation$rating, mu)
simple_average
```

```
## [1] 1.061202
```

This compares every value in `validation$rating` to `mu`, the population mean rating (μ of `edx$rating`) of the training set. It provides a high output of 1.0612018, but will be used as a comparision for other models.

Computation of basic effect bias models

Let us compute the basic models for both movie bias effect only and for movie and user bias effects.

```
#Movie effects only
predicted_ratings_movie_basic <-validation %>%
  left_join(movie_avgs, by='movieId') %>%
  mutate (pred = b_i_basic + mu) %>% pull(pred)

basic_movie_only <- RMSE(validation$rating, predicted_ratings_movie_basic)
basic_movie_only
```

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

This provides predicted values giving a RMSE of 0.9439087 for movie bias effect only.

```
#Movie and user effects
user_avgs <- edx %>%
  left_join(movie_avgs, by='movieId') %>%
  group_by(userId) %>%
  summarize(b_u_basic = mean(rating - mu - b_i_basic),n_user = n())

predicted_ratings_movie_user_basic <-
  validation %>%
  left_join(movie_avgs, by='movieId') %>%
  left_join(user_avgs, by='userId') %>%
  mutate(pred = mu + b_i_basic + b_u_basic) %>%
  pull(pred)

basic_movie_user <- RMSE(validation$rating, predicted_ratings_movie_user_basic)
basic_movie_user
```

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

This provides predicted values giving a RMSE of 0.8653488 for movie and user bias effects.

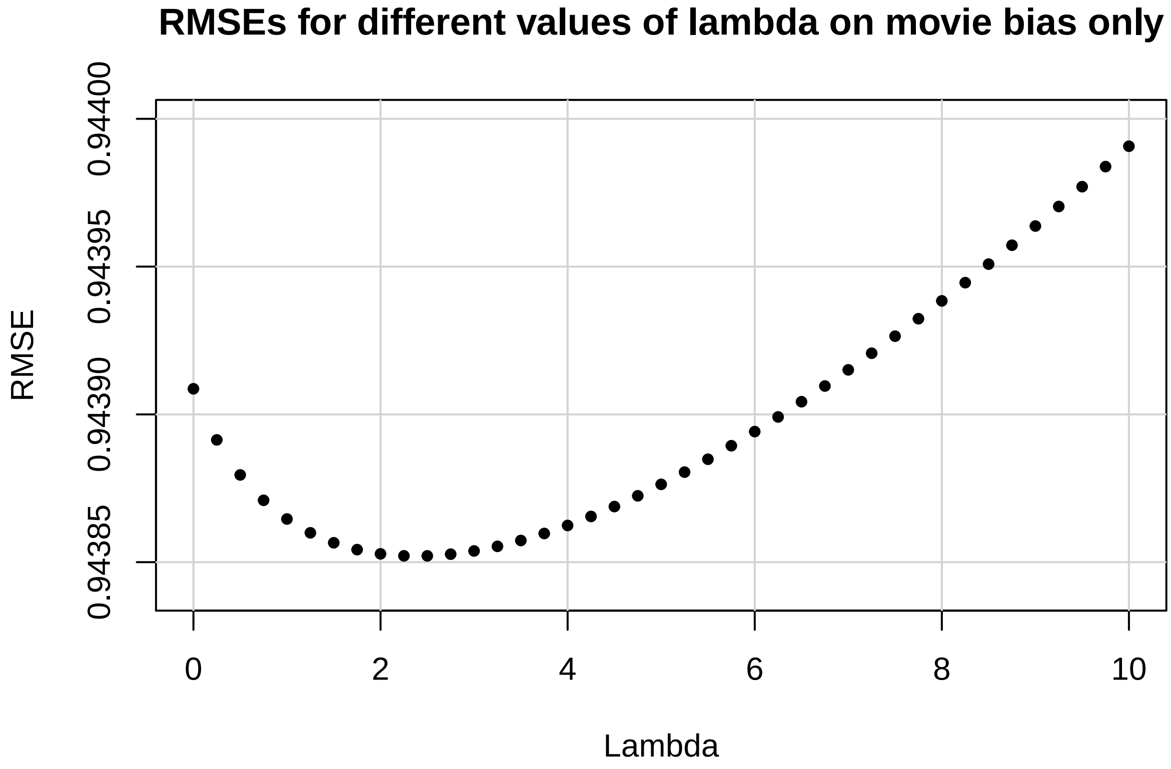
Regularised Models

Computation for Movie Bias Effect only

Let us consider the effect of various levels of lambda on the RMSE obtained, firstly on movie effects only.

```
lambdas <- seq(0, 10, 0.25)

rmses_movie_only <- sapply(lambdas, function(l){

  b_i <- edx %>%
    group_by(movieId) %>%
    summarize(b_i = sum(rating - mu)/(n()+l))

  predicted_ratings <-
    validation %>%
    left_join(b_i, by = "movieId") %>%
    mutate(pred = mu + b_i) %>%
    pull(pred)

  return(RMSE(validation$rating, predicted_ratings))
})
```

Let us compare the values of λ to the RMSE obtained at each value for movie bias effects only.



Let us obtain the relevant value of lambda and use this to produce an object `b_i_only`, which may then be used to confirm the effect of the optimised biasaes against the validation set and also for further analysis of the model performance.

```r
lambda_movie_only <- lambdas[which.min(rmses_movie_only)]

b_i_only <- edx %>%
  group_by(movieId,title,genres) %>%
  summarize(b_i_reg_movie = sum(rating - mu)/(n()+lambda_movie_only),n_movie = n()) %>%
  ungroup()

predicted_ratings_movie_reg <-
  validation %>%
  left_join(b_i_only, by = "movieId") %>%
  mutate(pred = mu + b_i_reg_movie) %>%
  pull(pred)

reg_movie_only <- RMSE(validation$rating, predicted_ratings_movie_reg)
reg_movie_only
```

```
## [1] 0.9438521
```

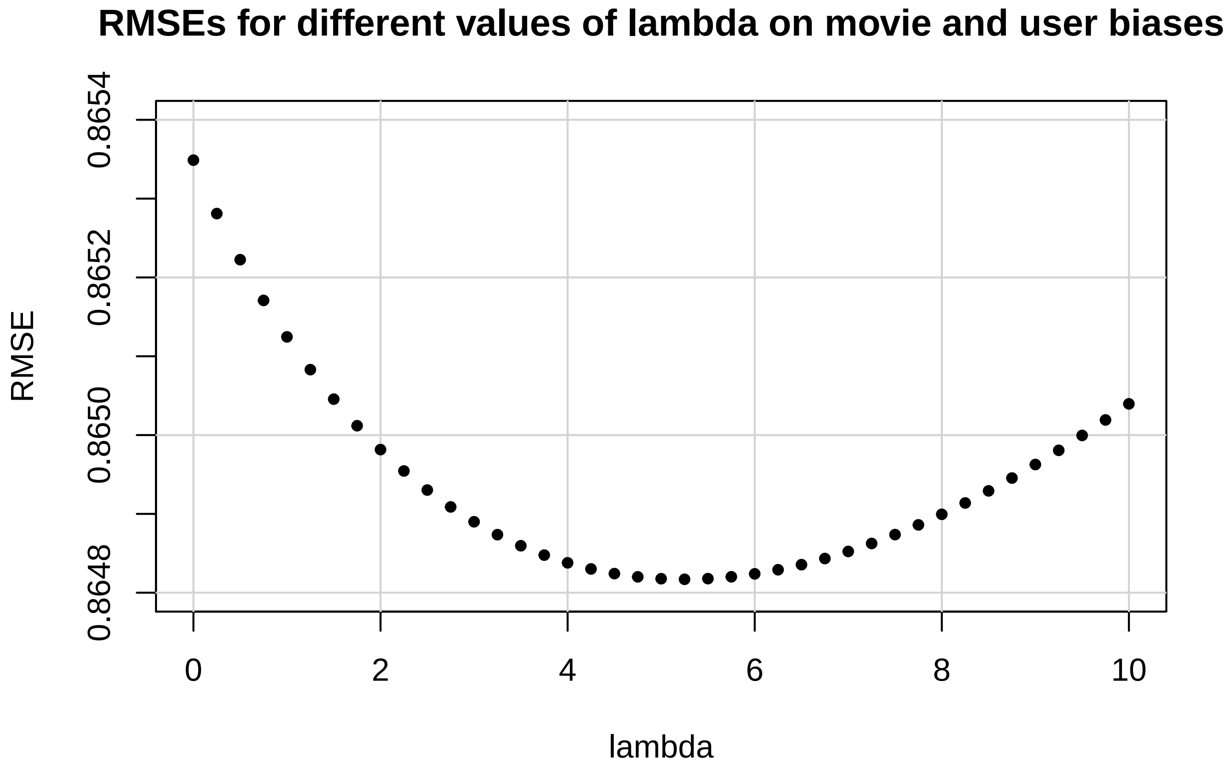
This provides predicted values giving an optimum RMSE of 0.9438521 at a regularisation parameter of $\lambda = 2.5$.

Computation for Movie and User Bias Effects

Let us now repeat this for both movie and user effects, in that order.

```
rmses_movie_User <- sapply(lambdas, function(l){

  b_i <- edx %>%
    group_by(movieId) %>%
    summarize(b_i = sum(rating - mu)/(n()+l))

  b_u <- edx %>%
    left_join(b_i, by="movieId") %>%
    group_by(userId) %>%
    summarize(b_u = sum(rating - b_i - mu)/(n()+l))

  predicted_ratings <-
    validation %>%
    left_join(b_i, by = "movieId") %>%
    left_join(b_u, by = "userId") %>%
    mutate(pred = mu + b_i + b_u) %>%
    pull(pred)

  return(RMSE(validation$rating,predicted_ratings))
})
```

Let us now compare the values of λ to the RMSE obtained at each value for movie then user bias effects.



Let us obtain the relevant value of lambda and use this to produce the objects `b_i` and `b_u`, which may then be used to confirm the effect of the optimised biasaes against the validation set and also for further analysis of the model performance.

```
lambda_movie_user <- lambdas[which.min(rmses_movie_User)]

b_i <- edx %>%
  group_by(movieId,title,genres) %>%
  summarize(b_i_reg = sum(rating - mu)/(n()+lambda_movie_user),n_movie = n()) %>%
  ungroup()

b_u <- edx %>%
  left_join(b_i, by="movieId") %>%
  group_by(userId) %>%
  summarize(b_u_reg = sum(rating - b_i_reg - mu)/(n()+lambda_movie_user))

predicted_ratings_movie_user_reg <-
  validation %>%
  left_join(b_i, by = "movieId") %>%
  left_join(b_u, by = "userId") %>%
  mutate(pred = mu + b_i_reg + b_u_reg) %>%
  pull(pred)

reg_movie_user <- RMSE(validation$rating,predicted_ratings_movie_user_reg)

reg_movie_user
```

```
## [1] 0.864817
```

This provides predicted values giving an optimum RMSE of 0.864817 at a regularisation parameter of $\lambda = 5.25$

Comparison of Basic and Regularised Models:

Let us compare the basic models (i.e. without regularisation) to the regularised models (using a value of λ so as to minimise RMSE:

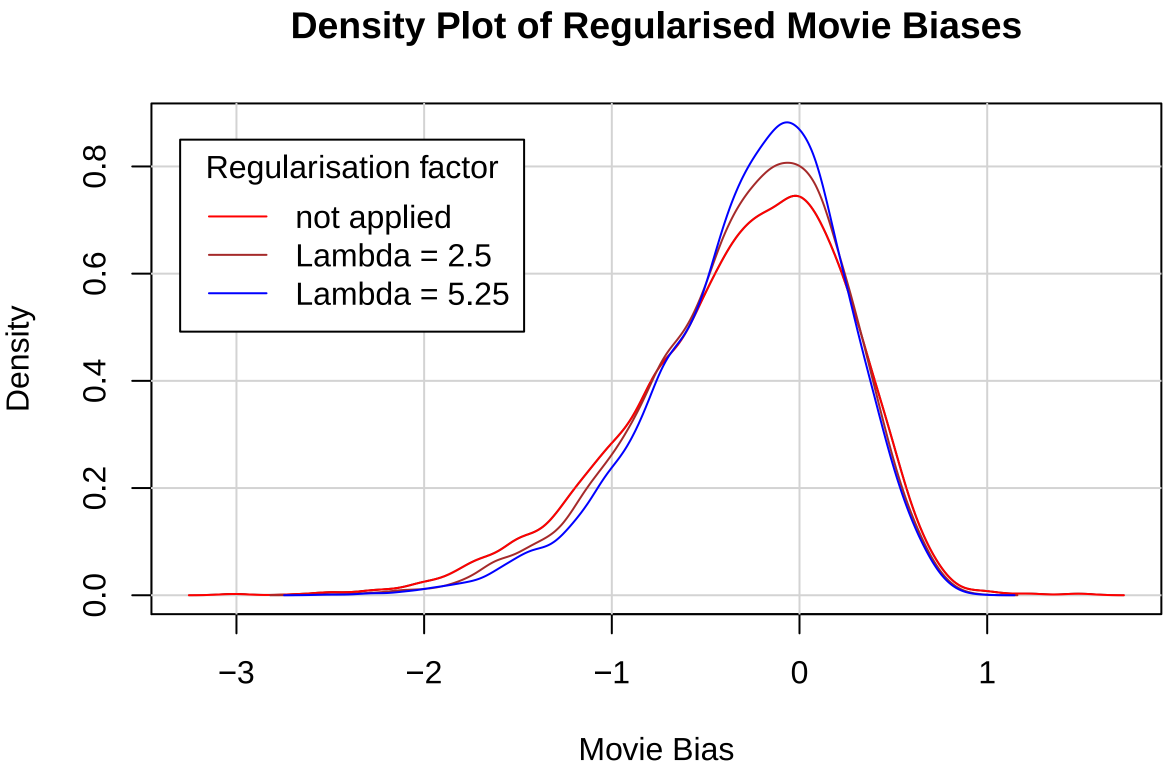
	Basic	Regularised
Average only	1.0612018	1.0612018
Movie bias effect only	0.9439087	0.9438521
Movie and user bias effects	0.8653488	0.8648170

The lowest RMSE is observed in the regularised version of the movie and user effects bias models.

Comparison of the Effects of Regularisation

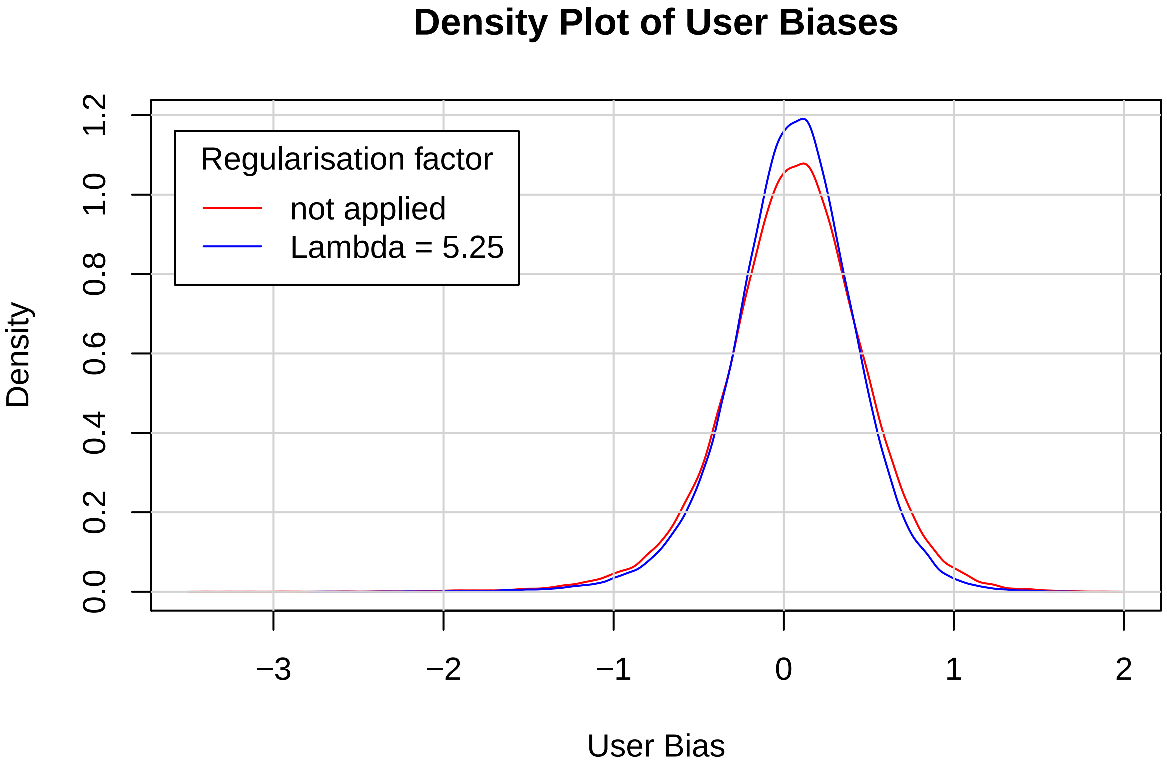
Let us consider how the different levels of regularisation have affected the distribution of movie and user biases; the model optimised for RMSE on movie bias only has a $\lambda = 2.5$ and the model optimised for RMSE on movie and user biases computed in that order has a $\lambda = 2.5$. Let us first consider the relevant descriptive statistics and the shape of the density function for movie bias effects (b_i).

b_i_basic	b_i_reg_movie	b_i_reg
Min. :-3.01247	Min. :-2.60168	Min. :-2.54336
1st Qu.:-0.66820	1st Qu.:-0.60708	1st Qu.:-0.56167
Median :-0.24461	Median :-0.22136	Median :-0.20320
Mean :-0.32073	Mean :-0.28797	Mean :-0.26728
3rd Qu.: 0.09691	3rd Qu.: 0.09101	3rd Qu.: 0.08255
Max. : 1.48753	Max. : 0.94258	Max. : 0.94249



Let us now consider the effect of regulation on the descriptive statistics and the shape of density function of user bias effects (b_u):

b_u_basic	b_u_reg
Min. :-3.39056	Min. :-2.68971
1st Qu.:-0.17948	1st Qu.:-0.16436
Median : 0.07288	Median : 0.06518
Mean : 0.06134	Mean : 0.05345
3rd Qu.: 0.32107	3rd Qu.: 0.28884
Max. : 1.89056	Max. : 1.66683



Ten highest-rated and ten lowest-rated derived from all models

Basic Models (i.e. without regularisation)

Let us consider the highest ten movie biases, without regularisation.

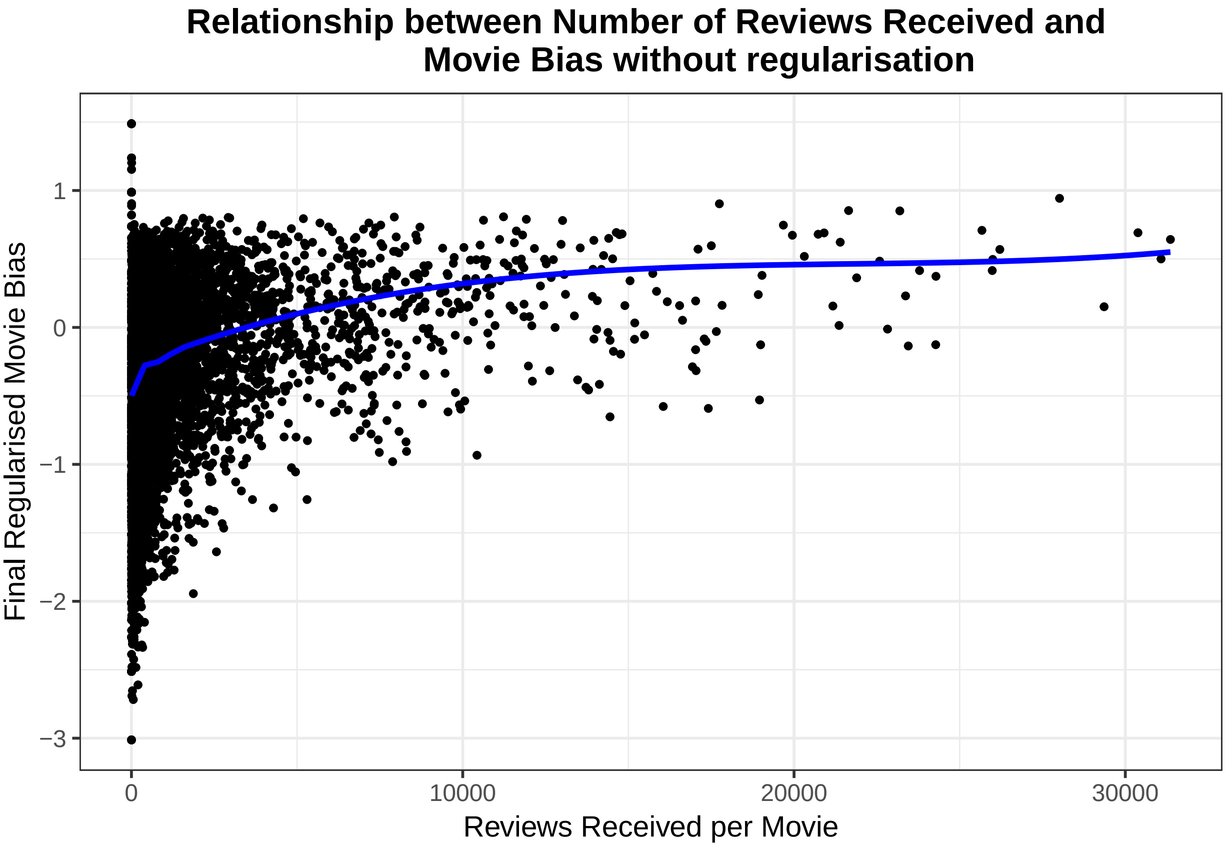
title	genres	b_i_basic	n_movie
Hellhounds on My Trail (1999)	Documentary	1.487535	1
Satan's Tango (SÃ¡tÃ¡ntangÃ³) (1994)	Drama	1.487535	2
Shadows of Forgotten Ancestors (1964)	Drama|Romance	1.487535	1
Fighting Elegy (Kenka erejii) (1966)	Action|Comedy	1.487535	1
Sun Alley (Sonnenallee) (1999)	Comedy|Romance	1.487535	1
Blue Light, The (Das Blaue Licht) ...	Drama|Fantasy|Mystery	1.487535	1
Who's Singin' Over There? (a.k.a. ...	Comedy	1.237535	4
Human Condition II, The (Ningen no...	Drama|War	1.237535	4
Human Condition III, The (Ningen n...	Drama|War	1.237535	4
Constantine's Sword (2007)	Documentary	1.237535	2

Let us consider `n_movie`, the number of users who have reviewed a particular movie. These appear to be very low numbers, comapred to the descriptive statistics above. In particular, the 5th percentile of `n_movie` is 4. This means that all ten movies have the same or fewer reviews than the 5th percentile. They are therefore all among the least-rated movies. Indeed, there are only a total of 21 for the movies with the highest ten average ratings (without regularisation).

title	genres	b_i_basic	n_movie
Besotted (2001)	Drama	-3.012465	2
Hi-Line, The (1999)	Drama	-3.012465	1
Accused (Anklaget) (2005)	Drama	-3.012465	1
Confessions of a Superhero (2007)	Documentary	-3.012465	1
War of the Worlds 2: The Next Wave...	Action	-3.012465	2
SuperBabies: Baby Geniuses 2 (2004)	Comedy	-2.717822	56
Hip Hop Witch, Da (2000)	Comedy|Horror|Thriller	-2.691037	14
Disaster Movie (2008)	Comedy	-2.653090	32
From Justin to Kelly (2003)	Musical|Romance	-2.610455	199
Criminals (1996)	Documentary	-2.512465	2

All ten of the highest-rated (on average) movies are therefore in the lowest 5th percentile of ratings received, as are 6 of the ten lowest-rated (on average) without regularisation. These movies have received ‘$sum(lowest_{t}en_{b}asic$n_movie)‘\$ ratings between them.

Let us consider the relationship between `n_movie` and the movie bias `b_i_basic` as computed without regularisation.



Regression line (blue) calculated by LOESS method

Model with Regularisation optimised for RMSE on movie bias effect only

title	genres	b_i_reg_movie	n_movie
Shawshank Redemption, The (1994)	Drama	0.9425819	28015
Godfather, The (1972)	Crime|Drama	0.9027736	17747
More (1998)	Animation|IMAX|Sci-Fi	0.8855520	7
Usual Suspects, The (1995)	Crime|Mystery|Thriller	0.8532899	21648
Schindler's List (1993)	Drama|War	0.8509364	23193
Casablanca (1942)	Drama|Romance	0.8077788	11232
Rear Window (1954)	Mystery|Thriller	0.8059324	7935
Sunset Blvd. (a.k.a. Sunset Boulev...	Drama|Film-Noir|Romance	0.8027275	2922
Third Man, The (1949)	Film-Noir|Mystery|Thri...	0.7982878	2967
Double Indemnity (1944)	Crime|Drama|Film-Noir	0.7974264	2154

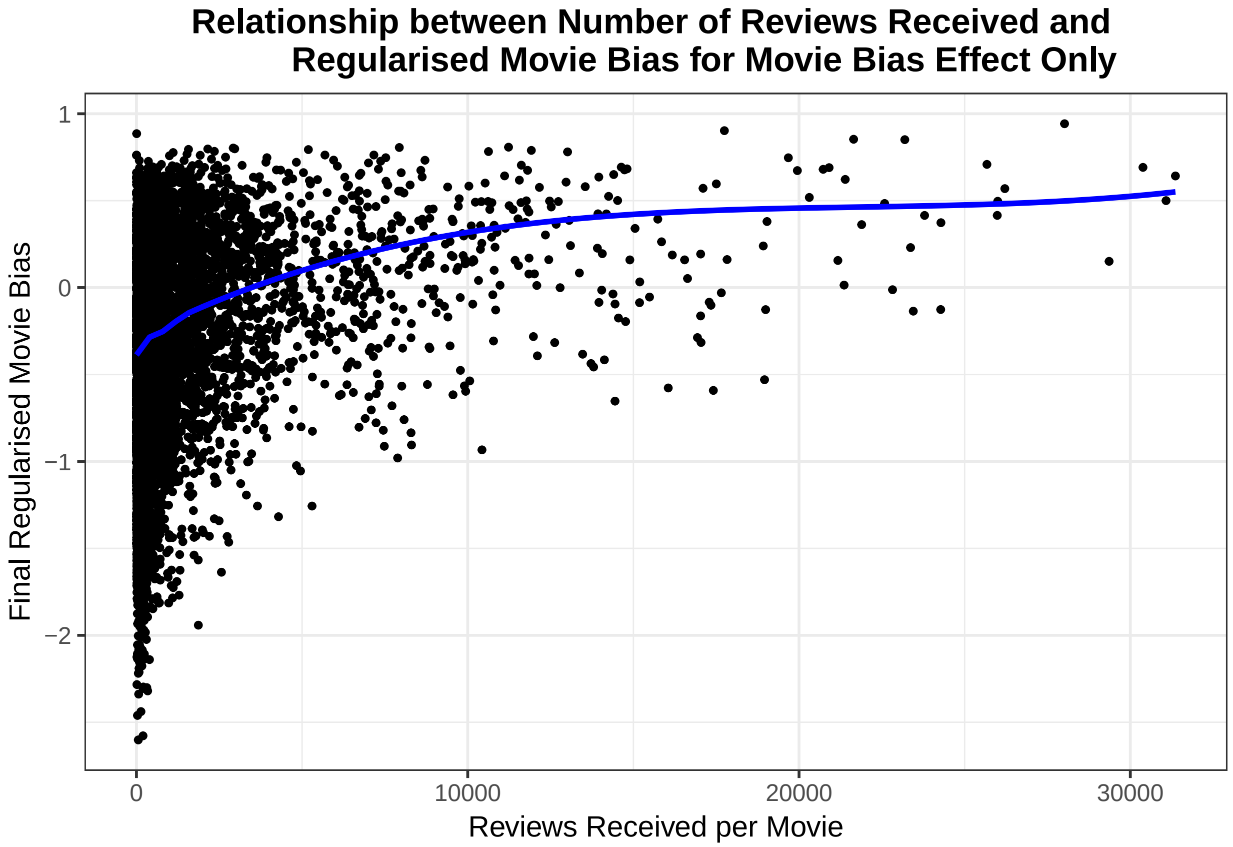
These have much higher numbers than without regularisation; indeed, it is helpful to obtain the 95th and 99th percentiles of `n_movie` for the overall training set. These are 4025.6 and 1.15211×10^4 respectively.

It may be obsereved that 6 of these movies are above the 95th percentile, and of these 4 also exceed the 99th percentile. The sum of ratings received by these ten movies is 117820.

title	genres	b_i_reg_movie	n_movie
SuperBabies: Baby Geniuses 2 (2004)	Comedy	-2.601676	56
From Justin to Kelly (2003)	Musical|Romance	-2.578067	199
Disaster Movie (2008)	Comedy	-2.460837	32
PokÃ©mon Heroes (2003)	Animation|Children	-2.438765	137
Carnosaur 3: Primal Species (1996)	Horror|Sci-Fi	-2.338264	68
Glitter (2001)	Drama|Musical|Romance	-2.319841	339
Pokemon 4 Ever (a.k.a. PokÃ©mon 4:...	Adventure|Animation|Ch...	-2.305711	202
Gigli (2003)	Action|Crime|Drama	-2.300797	313
Barney's Great Adventure (1998)	Adventure|Children	-2.297353	208
Hip Hop Witch, Da (2000)	Comedy|Horror|Thriller	-2.283304	14

We can therefore see that regularisation has improved the highest and lowest ten movies by rating to those that have been viewed by a wider section of users. It may be observed that all these fall between the 5th and 95th percentiles of 4 and 4026 respectively. It is also observed that there are a total of 1568 ratings in this table, which is greater than without regularisation.

Let us consider the relationship between `n_movie` and the computed, regularised movie bias `b_i_reg` as optimised for movie bias effect only.



Regression line (blue) calculated by LOESS method

Model with Regularisation optimised for RMSE on movie and user bias effects

title	genres	b_i_reg	n_movie
Shawshank Redemption, The (1994)	Drama	0.9424894	28015
Godfather, The (1972)	Crime|Drama	0.9026338	17747
Usual Suspects, The (1995)	Crime|Mystery|Thriller	0.8531815	21648
Schindler's List (1993)	Drama|War	0.8508355	23193
Casablanca (1942)	Drama|Romance	0.8075811	11232
Rear Window (1954)	Mystery|Thriller	0.8056533	7935
Sunset Blvd. (a.k.a. Sunset Boulev…	Drama|Film-Noir|Romance	0.8019734	2922
Third Man, The (1949)	Film-Noir|Mystery|Thri…	0.7975492	2967
Double Indemnity (1944)	Crime|Drama|Film-Noir	0.7964108	2154
Paths of Glory (1957)	Drama|War	0.7936033	1571

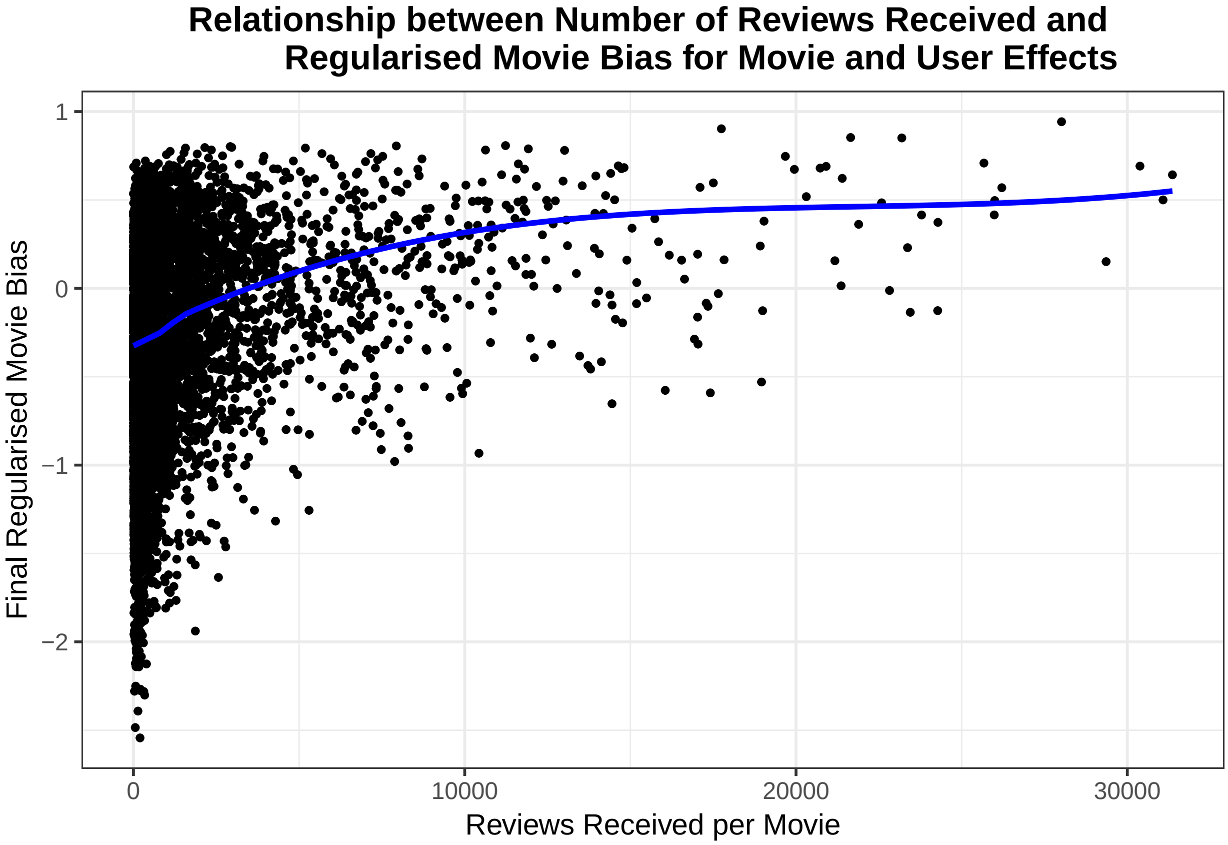
It may be observed that there are 119384 total ratings given to the highest ten movies. Of these, 6 have more than the 95th percentile of ratings and 4 have more than the 99th percentile of ratings.

title	genres	b_i_reg	n_movie
From Justin to Kelly (2003)	Musical|Romance	-2.543356	199
SuperBabies: Baby Geniuses 2 (2004)	Comedy	-2.484866	56
PokÃ©mon Heroes (2003)	Animation|Children	-2.391618	137
Glitter (2001)	Drama|Musical|Romance	-2.301309	339
Gigli (2003)	Action|Crime|Drama	-2.280916	313
Disaster Movie (2008)	Comedy	-2.279165	32
Pokemon 4 Ever (a.k.a. PokÃ©mon 4:…	Adventure|Animation|Ch…	-2.275117	202
Barney's Great Adventure (1998)	Adventure|Children	-2.267727	208
Carnosaur 3: Primal Species (1996)	Horror|Sci-Fi	-2.250480	68
Son of the Mask (2005)	Action|Adventure|Comed…	-2.141303	165

As for the earlier model with regularisation optimised for movie bias effects only, it may be observed that all these fall between the 5th and 95th percentiles of 4 and 4025.6 respectively. It is also observed that there are a total of 1719 ratings in this table, which is greater than with regularisation for movie effects only.

We see a similar effect produced when optimising for RMSE based on both user and movie bias effects; however we do not notice any movies with fewer than 30 individual ratings (the first quartile) received in either category.

Let us consider the relationship between `n_movie` and the computed, regularised movie bias `b_i_reg` as optimised for movie and user bias effects.



Regression line (blue) calculated by LOESS method

Confirmation of Output Model

The model that is selected is that of regularised movie and user bias effects, with a regularistaion parameter of $\lambda = 5.25$. This yields a RMSE of 0.864817. The descriptive statistics of the predicted values of the final model are as follows:

Predicted Rating
Min. :-0.4065
1st Qu.: 3.1392
Median : 3.5626
Mean : 3.5095
3rd Qu.: 3.9359
Max. : 5.9909

In addition, there are 86 predictions below the minimum possible value of 0.5 and 1533 predictions above the maximum possible value of 5.

Discussion and Conclusions

It is observed that the inclusion of both movie and user biases provides a better performance (measured by a lower RMSE), both with and without regularisation. It is also observed that regulariaation provides further, albeit small, improvement in performance as measured by a reduction in RMSE. We can observe from the density plots above that this also reduces the proportion of outlying movie biases, with a more central distribution of biases for both movie and user effects. This reduces the distorting effect of a relatively small number of high or low ratings. This is reflected in the less extreme minimum and maximum values of biases as greater levels of regularisation are applied.

This may also be observed by considering the `n_movie` values observed in the top ten and lowest ten provided; there is a general increase in the value of `n_movie` as higher levels of regularisation are applied. This is more pronounced for the highest ten movies; the un-regularised top 10 has a total of 21 movies (a mean of just over 2 ratings per movie) whereas the final model (regularised by lambda for optimum RSE) yields an total n_movie of 119,384. This represents an increase of over a thousand-fold. However, there are more total ratings per movie in the non-regularised lowest ten movies, and the effect of regularisation is such that the total `n_movie` value increases less than tenfold with regularisation applied to both movie and user bias effects. This is most obvious when viewing the regression plots of number of ratings received per movie against movie bias both before and after regularisation; there are a few points protruding for non-regularised biases at very low levels of reviews per movie for non-regularised movie biases, but these do not appear to protrude on the regression plot between the number of ratings received and regularised movie biases.

It is also noted that this is higher for the top ten than the lowest ten; this also suggests that movies that are enjoyed by a large number of users may still receive a higher rating. It is also notable that movies that are repeatedly unpopular will still receive a lower rating; however it is necessary for repeated adverse ratings to be given for this to occur. This is more consistent with good movies receiving consistently better ratings and also a greater number of ratings. This suggests that the movies in the highest ten chart based on regularised movie biases are so highly rated because they are considered by the people giving the ratings to be good movies.

References and Bibliography

Irizarry, R. 2019. *Introduction to Data Science* as published at https://rafalab.github.io/dsbook/ and linked pages on the same website. Also sample code from https://github.com/rafalab/dsbook was consulted.

Wickham, H., 2014. Tidy data. *Journal of Statistical Software*, 59(10), pp.1-23.

Data Sources

MovieLens 10M Dataset from GroupLens per https://grouplens.org/datasets/movielens/10m/