Evaluation of Movie Recommendation System with RMSE for 10M dataset

Sang Kim

```
################################################################
# Create data set, validation set
################################################################


if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages ---------------------------------------------------------------------------------------------------
---------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------------------------------
----------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: caret
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
# 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 <- read.table(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")
movies <- as.data.frame(movies) %>% mutate(movieId = as.numeric(levels(movieId))[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)
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)
```

```
## Joining, by = c("userId", "movieId", "rating", "timestamp", "title", "genres")
```

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

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




library(tidyverse)
library(dslabs)


edx %>% as_tibble()
```

```
## # A tibble: 9,000,055 x 6
##    userId movieId rating timestamp title                genres
##  *  <int>   <dbl>  <dbl>     <int> <chr>                <chr>
##  1      1     122      5 838985046 Boomerang (1992)    Comedy|Romance
##  2      1     185      5 838983525 Net, The (1995)     Action|Crime|Thrill~
##  3      1     292      5 838983421 Outbreak (1995)     Action|Drama|Sci-Fi~
##  4      1     316      5 838983392 Stargate (1994)     Action|Adventure|Sc~
##  5      1     329      5 838983392 Star Trek: Generat~ Action|Adventure|Dr~
##  6      1     355      5 838984474 Flintstones, The (~ Children|Comedy|Fan~
##  7      1     356      5 838983653 Forrest Gump (1994) Comedy|Drama|Romanc~
##  8      1     362      5 838984885 Jungle Book, The (~ Adventure|Children|~
##  9      1     364      5 838983707 Lion King, The (19~ Adventure|Animation~
## 10      1     370      5 838984596 Naked Gun 33 1/3: ~ Action|Comedy
## # ... with 9,000,045 more rows
```

```
edx %>%
  summarize(n_users = n_distinct(userId),
            n_movies = n_distinct(movieId))
```

```
##   n_users n_movies
## 1   69878    10677
```

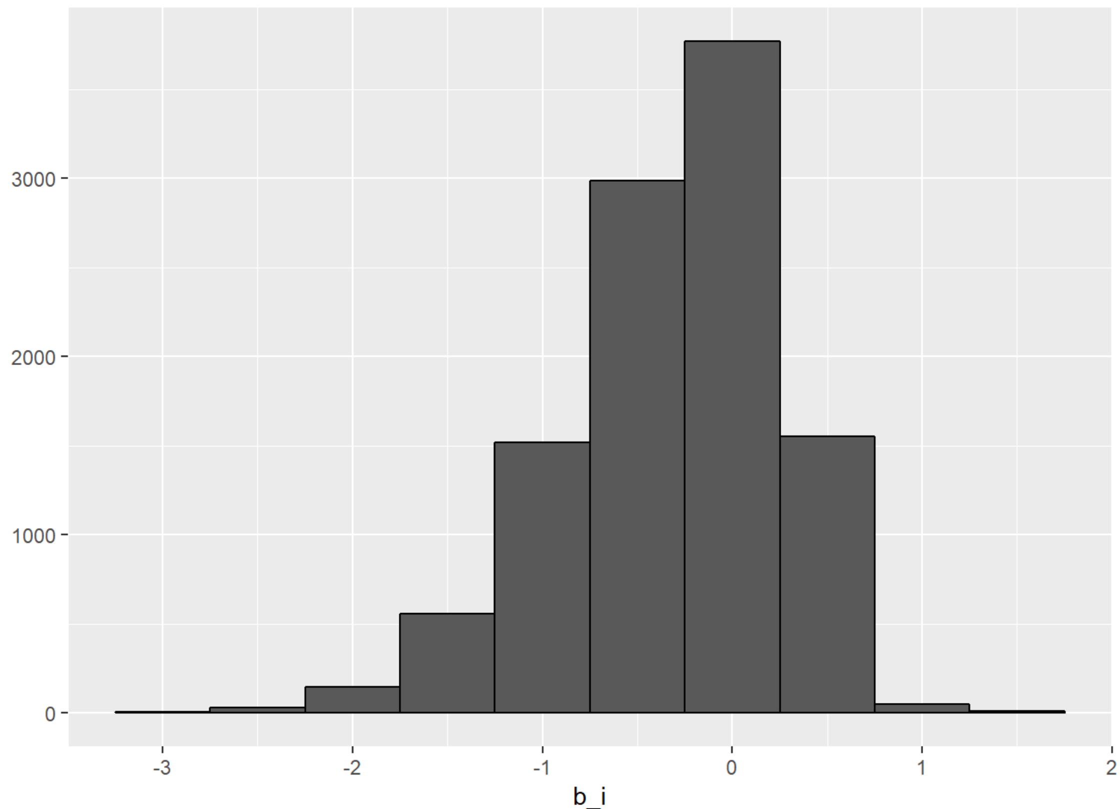
```
library(caret)
set.seed(755)
test_index <- createDataPartition(y = edx$rating, times = 1, p = 0.2, list = FALSE)
train_set <- edx[-test_index,]
test_set <- edx[test_index,]


test_set <- test_set %>%
  semi_join(train_set, by = "movieId") %>%
  semi_join(train_set, by = "userId")

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


mu_hat <- mean(train_set$rating)
mu_hat
```

```
## [1] 3.512527
```

```
naive_rmse <- RMSE(test_set$rating, mu_hat)
naive_rmse
```

```
## [1] 1.060561
```

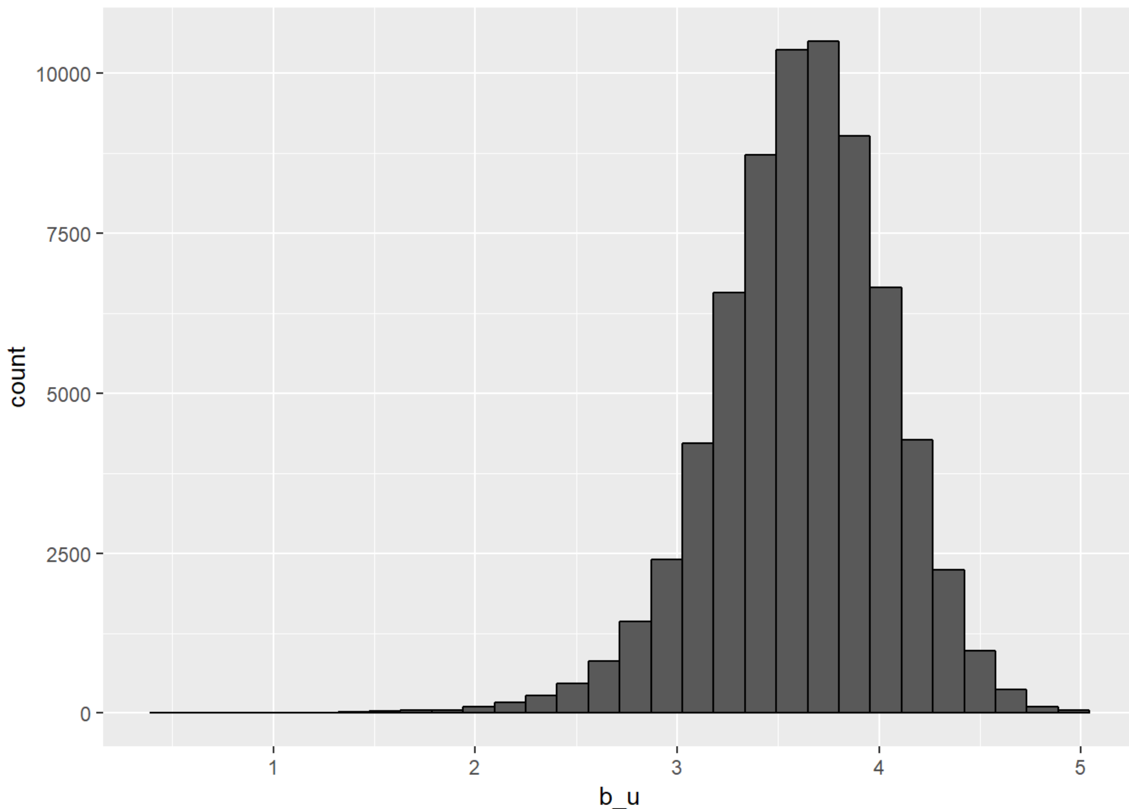
```
rmse_results <- data_frame(method = "Just the average", RMSE = naive_rmse)


mu <- mean(train_set$rating)
movie_avgs <- train_set %>%
  group_by(movieId) %>%
  summarize(b_i = mean(rating - mu))

movie_avgs %>% qplot(b_i, geom ="histogram", bins = 10, data = ., color = I("black"))
```




```
predicted_ratings <- mu + test_set %>%
  left_join(movie_avgs, by='movieId') %>%
  pull(b_i)

model_1_rmse <- RMSE(predicted_ratings, test_set$rating)
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Movie Effect Model",
                                     RMSE = model_1_rmse))
rmse_results
```

```
## # A tibble: 2 x 2
##   method              RMSE
##   <chr>              <dbl>
## 1 Just the average   1.06
## 2 Movie Effect Model 0.944
```

```
train_set %>%
  group_by(userId) %>%
  summarize(b_u = mean(rating)) %>%
  filter(n()>=100) %>%
  ggplot(aes(b_u)) +
  geom_histogram(bins = 30, color = "black")
```




```
user_avgs <- train_set %>%
  left_join(movie_avgs, by='movieId') %>%
  group_by(userId) %>%
  summarize(b_u = mean(rating - mu - b_i))


predicted_ratings <- test_set %>%
  left_join(movie_avgs, by='movieId') %>%
  left_join(user_avgs, by='userId') %>%
  mutate(pred = mu + b_i + b_u) %>%
  pull(pred)


model_2_rmse <- RMSE(predicted_ratings, test_set$rating)
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Movie + User Effects Model",
                                     RMSE = model_2_rmse))
rmse_results
```

```
## # A tibble: 3 x 2
##   method                      RMSE
##   <chr>                      <dbl>
## 1 Just the average           1.06
## 2 Movie Effect Model         0.944
## 3 Movie + User Effects Model 0.867
```

```
test_set %>%
  left_join(movie_avgs, by='movieId') %>%
  mutate(residual = rating - (mu + b_i)) %>%
  arrange(desc(abs(residual))) %>%
  select(title,  residual) %>% slice(1:10)
```

```
##                                                                      title
## 1                                            Pok<U+CC55>mon Heroes (2003)
## 2  Samurai Rebellion (J<U+CC99>i-uchi: Hairy<U+CC99> tsuma shimatsu) (1967)
## 3                                        Shawshank Redemption, The (1994)
## 4                                        Shawshank Redemption, The (1994)
## 5                                        Shawshank Redemption, The (1994)
## 6                                        Shawshank Redemption, The (1994)
## 7                                        Shawshank Redemption, The (1994)
## 8                                                   Godfather, The (1972)
## 9                                                   Godfather, The (1972)
## 10                                                  Godfather, The (1972)
##    residual
## 1   4.00000
## 2  -4.00000
## 3  -3.95308
## 4  -3.95308
## 5  -3.95308
## 6  -3.95308
## 7  -3.95308
## 8  -3.91806
## 9  -3.91806
## 10 -3.91806
```

```
movie_titles <- edx %>%
  select(movieId, title) %>%
  distinct()

movie_avgs %>% left_join(movie_titles, by="movieId") %>%
  arrange(desc(b_i)) %>%
  select(title, b_i) %>%
  slice(1:10)
```

```
## # A tibble: 10 x 2
##    title                                                                  b_i
##    <chr>                                                                <dbl>
##  1 Hellhounds on My Trail (1999)                                         1.49
##  2 Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to ta~  1.49
##  3 Satan's Tango (S<U+CC3C>t<U+CC3C>ntang<U+CC98>) (1994)                                 1.49
##  4 Fighting Elegy (Kenka erejii) (1966)                                  1.49
##  5 Sun Alley (Sonnenallee) (1999)                                        1.49
##  6 Along Came Jones (1945)                                               1.49
##  7 Angus, Thongs and Perfect Snogging (2008)                             1.49
##  8 Bullfighter and the Lady (1951)                                       1.49
##  9 Blue Light, The (Das Blaue Licht) (1932)                              1.49
## 10 Constantine's Sword (2007)                                            1.49
```

```
movie_avgs %>% left_join(movie_titles, by="movieId") %>%
  arrange(b_i) %>%
  select(title, b_i) %>%
  slice(1:10)
```

```
## # A tibble: 10 x 2
##    title                                     b_i
##    <chr>                                   <dbl>
##  1 Besotted (2001)                         -3.01
##  2 Grief (1993)                            -3.01
##  3 Altered (2006)                          -3.01
##  4 Accused (Anklaget) (2005)               -3.01
##  5 Confessions of a Superhero (2007)       -3.01
##  6 War of the Worlds 2: The Next Wave (2008) -3.01
##  7 Karla (2006)                            -2.76
##  8 SuperBabies: Baby Geniuses 2 (2004)     -2.75
##  9 Disaster Movie (2008)                   -2.70
## 10 From Justin to Kelly (2003)             -2.60
```

```
train_set %>% count(movieId) %>%
  left_join(movie_avgs) %>%
  left_join(movie_titles, by="movieId") %>%
  arrange(desc(b_i)) %>%
  select(title, b_i, n) %>%
  slice(1:10)
```

```
## Joining, by = "movieId"
```

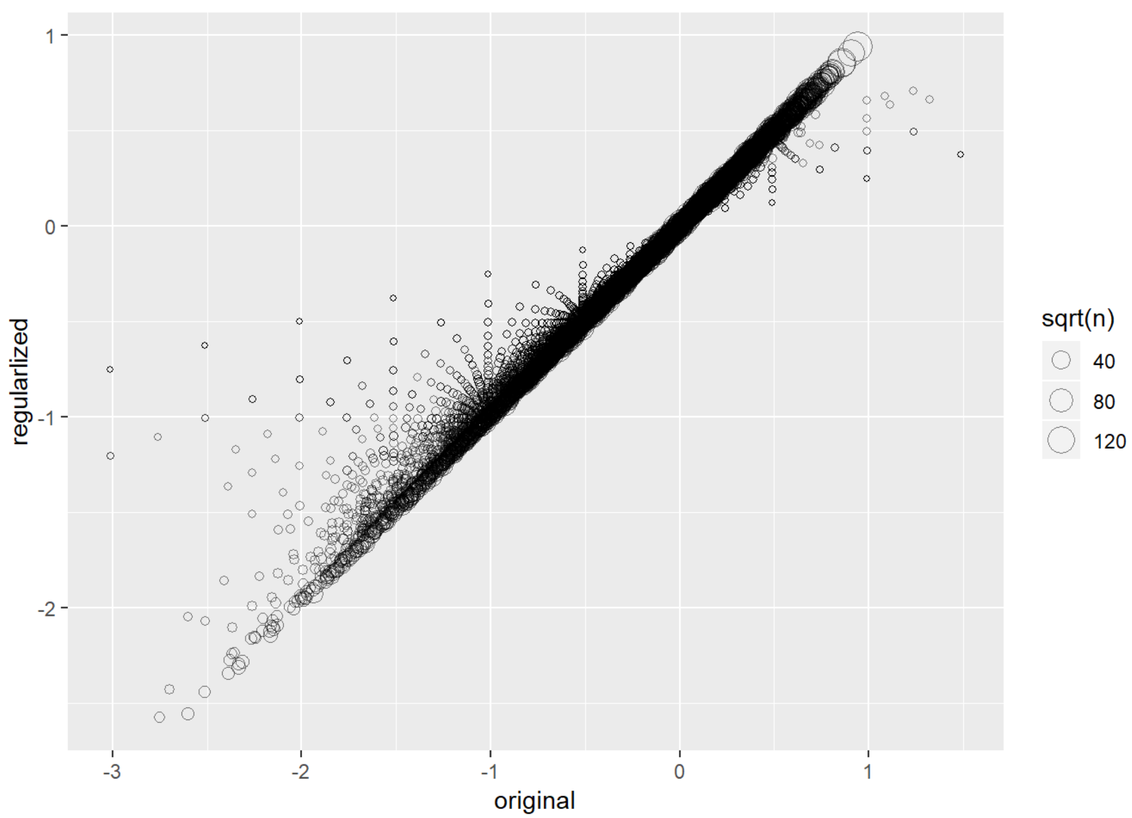
```
## # A tibble: 10 x 3
##    title                                                      b_i     n
##    <chr>                                                    <dbl> <int>
##  1 Hellhounds on My Trail (1999)                             1.49     1
##  2 Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko~  1.49     1
##  3 Satan's Tango (S<U+CC3C>t<U+CC3C>ntang<U+CC98>) (1994)                  1.49     1
##  4 Fighting Elegy (Kenka erejii) (1966)                      1.49     1
##  5 Sun Alley (Sonnenallee) (1999)                            1.49     1
##  6 Along Came Jones (1945)                                   1.49     1
##  7 Angus, Thongs and Perfect Snogging (2008)                 1.49     1
##  8 Bullfighter and the Lady (1951)                           1.49     1
##  9 Blue Light, The (Das Blaue Licht) (1932)                  1.49     1
## 10 Constantine's Sword (2007)                                1.49     1
```

```
train_set %>% count(movieId) %>%
  left_join(movie_avgs) %>%
  left_join(movie_titles, by="movieId") %>%
  arrange(b_i) %>%
  select(title, b_i, n) %>%
  slice(1:10)
```

```
## Joining, by = "movieId"
```

```
## # A tibble: 10 x 3
##    title                                     b_i     n
##    <chr>                                   <dbl> <int>
##  1 Besotted (2001)                         -3.01     2
##  2 Grief (1993)                            -3.01     1
##  3 Altered (2006)                          -3.01     1
##  4 Accused (Anklaget) (2005)               -3.01     1
##  5 Confessions of a Superhero (2007)       -3.01     1
##  6 War of the Worlds 2: The Next Wave (2008) -3.01   2
##  7 Karla (2006)                            -2.76     2
##  8 SuperBabies: Baby Geniuses 2 (2004)     -2.75    44
##  9 Disaster Movie (2008)                   -2.70    27
## 10 From Justin to Kelly (2003)             -2.60   159
```

```
lambda <- 3
mu <- mean(train_set$rating)
movie_reg_avgs <- train_set %>%
  group_by(movieId) %>%
  summarize(b_i = sum(rating - mu)/(n()+lambda), n_i = n())

data_frame(original = movie_avgs$b_i,
           regularlized = movie_reg_avgs$b_i,
           n = movie_reg_avgs$n_i) %>%
  ggplot(aes(original, regularlized, size=sqrt(n))) +
  geom_point(shape=1, alpha=0.5)
```




```
train_set %>%
  count(movieId) %>%
  left_join(movie_reg_avgs, by = "movieId") %>%
  left_join(movie_titles, by = "movieId") %>%
  arrange(desc(b_i)) %>%
  select(title, b_i, n) %>%
  slice(1:10)
```

```
## # A tibble: 10 x 3
##    title                                         b_i     n
##    <chr>                                       <dbl> <int>
##  1 Shawshank Redemption, The (1994)            0.940 22432
##  2 Godfather, The (1972)                       0.905 14230
##  3 Schindler's List (1993)                     0.857 18486
##  4 Usual Suspects, The (1995)                  0.851 17330
##  5 Rear Window (1954)                          0.808  6334
##  6 Casablanca (1942)                           0.805  9025
##  7 Sunset Blvd. (a.k.a. Sunset Boulevard) (1950) 0.796  2281
##  8 Double Indemnity (1944)                     0.794  1744
##  9 Godfather: Part II, The (1974)              0.793  9521
## 10 Seven Samurai (Shichinin no samurai) (1954)   0.792  4144
```

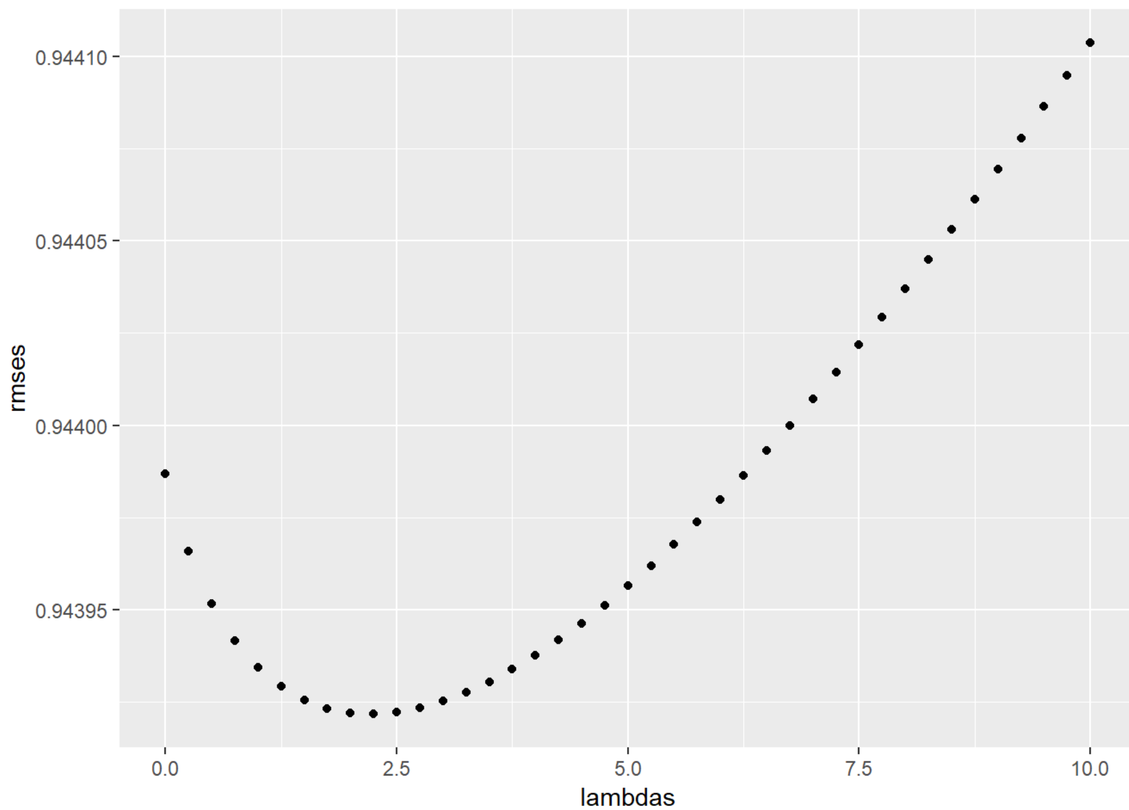
```
train_set %>%
  count(movieId) %>%
  left_join(movie_reg_avgs, by = "movieId") %>%
  left_join(movie_titles, by="movieId") %>%
  arrange(b_i) %>%
  select(title, b_i, n) %>%
  slice(1:10)
```

```
## # A tibble: 10 x 3
##    title                                                b_i     n
##    <chr>                                              <dbl> <int>
##  1 SuperBabies: Baby Geniuses 2 (2004)                -2.58    44
##  2 From Justin to Kelly (2003)                        -2.56   159
##  3 Pok<U+CC55>mon Heroes (2003)                             -2.44   106
##  4 Disaster Movie (2008)                              -2.43    27
##  5 Pokemon 4 Ever (a.k.a. Pok<U+CC55>mon 4: The Movie) (2002) -2.34   155
##  6 Glitter (2001)                                     -2.31   274
##  7 Barney's Great Adventure (1998)                    -2.30   170
##  8 Gigli (2003)                                       -2.29   268
##  9 Yu-Gi-Oh! (2004)                                   -2.28    65
## 10 Faces of Death: Fact or Fiction? (1999)            -2.24    52
```

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

model_3_rmse <- RMSE(predicted_ratings, test_set$rating)
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Regularized Movie Effect Model",
                                     RMSE = model_3_rmse))
rmse_results
```

```
## # A tibble: 4 x 2
##   method                          RMSE
##   <chr>                          <dbl>
## 1 Just the average               1.06
## 2 Movie Effect Model             0.944
## 3 Movie + User Effects Model     0.867
## 4 Regularized Movie Effect Model 0.944
```

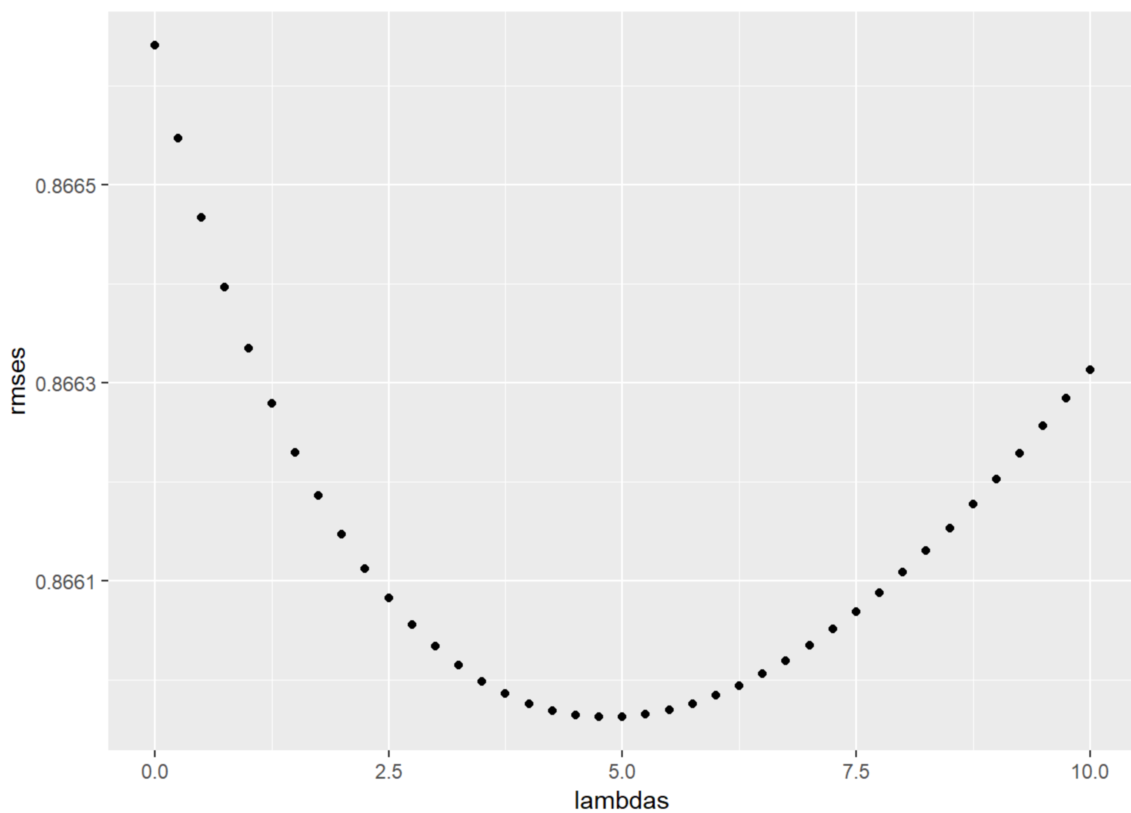
```r
lambdas <- seq(0, 10, 0.25)

mu <- mean(train_set$rating)
just_the_sum <- train_set %>%
  group_by(movieId) %>%
  summarize(s = sum(rating - mu), n_i = n())

rmses <- sapply(lambdas, function(l){
  predicted_ratings <- test_set %>%
    left_join(just_the_sum, by='movieId') %>%
    mutate(b_i = s/(n_i+l)) %>%
    mutate(pred = mu + b_i) %>%
    pull(pred)
  return(RMSE(predicted_ratings, test_set$rating))
})
qplot(lambdas, rmses)
```




```r
lambdas[which.min(rmses)]
```

```
## [1] 2.25
```

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

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

  mu <- mean(train_set$rating)

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

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

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

  return(RMSE(predicted_ratings, test_set$rating))
})

qplot(lambdas, rmses)
```




```
lambda <- lambdas[which.min(rmses)]
lambda
```

```
## [1] 4.75
```

```
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Regularized Movie + User Effect Model",
                                     RMSE = min(rmses)))
rmse_results %>% knitr::kable()
```

method	RMSE
Just the average	1.0605613
Movie Effect Model	0.9439868
Movie + User Effects Model	0.8666408
Regularized Movie Effect Model	0.9439252
Regularized Movie + User Effect Model	0.8659626