

Evaluation of Movie Recommendation System with RMSE for 100K dataset

Sang Kim

```
library(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()
```

```
library(dslabs)  
data("movielens")  
  
dim(movielens)
```

```
## [1] 100004      7
```

```
movielens %>% as_tibble()
```

```
## # A tibble: 100,004 x 7  
##   movieId title          year genres          userId rating timestamp  
##   <int> <chr>          <int> <fct>          <int> <dbl> <int>  
## 1      31 Dangerous Minds    1995 Drama          1     2.5  1.26e9  
## 2     1029 Dumbo            1941 Animation|Chil~  1     3    1.26e9  
## 3     1061 Sleepers         1996 Thriller       1     3    1.26e9  
## 4     1129 Escape from New Y~ 1981 Action|Adventu~  1     2    1.26e9  
## 5     1172 Cinema Paradiso (~ 1989 Drama          1     4    1.26e9  
## 6     1263 Deer Hunter, The   1978 Drama|War      1     2    1.26e9  
## 7     1287 Ben-Hur           1959 Action|Adventu~  1     2    1.26e9  
## 8     1293 Gandhi            1982 Drama          1     2    1.26e9  
## 9     1339 Dracula (Bram Sto~ 1992 Fantasy|Horror~  1    3.5  1.26e9  
## 10    1343 Cape Fear         1991 Thriller        1     2    1.26e9  
## # ... with 99,994 more rows
```

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

```
##   n_users n_movies  
## 1     671   9066
```

```
library(caret)
```

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

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

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

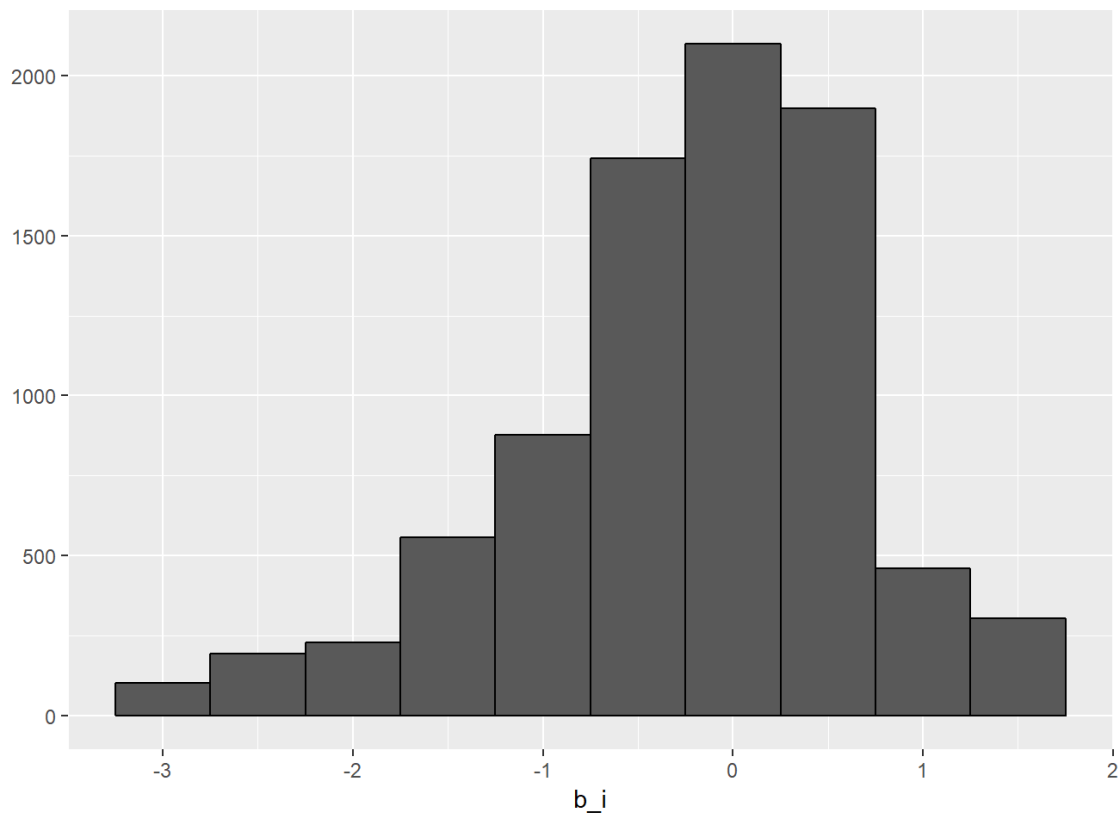
```
set.seed(755)  
test_index <- createDataPartition(y = movielens$rating, times = 1, p = 0.2, list = FALSE)  
train_set <- movielens[-test_index,]  
test_set <- movielens[test_index,]  
  
test_set <- test_set %>%  
  semi_join(train_set, by = "movieId") %>%  
  semi_join(train_set, by = "userId")  
  
RMSE <- function(true_ratings, predicted_ratings){  
  sqrt(mean((true_ratings - predicted_ratings)^2))  
}  
  
mu_hat <- mean(train_set$rating)  
mu_hat
```

```
## [1] 3.542793
```

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

```
## [1] 1.04822
```

```
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"))
```



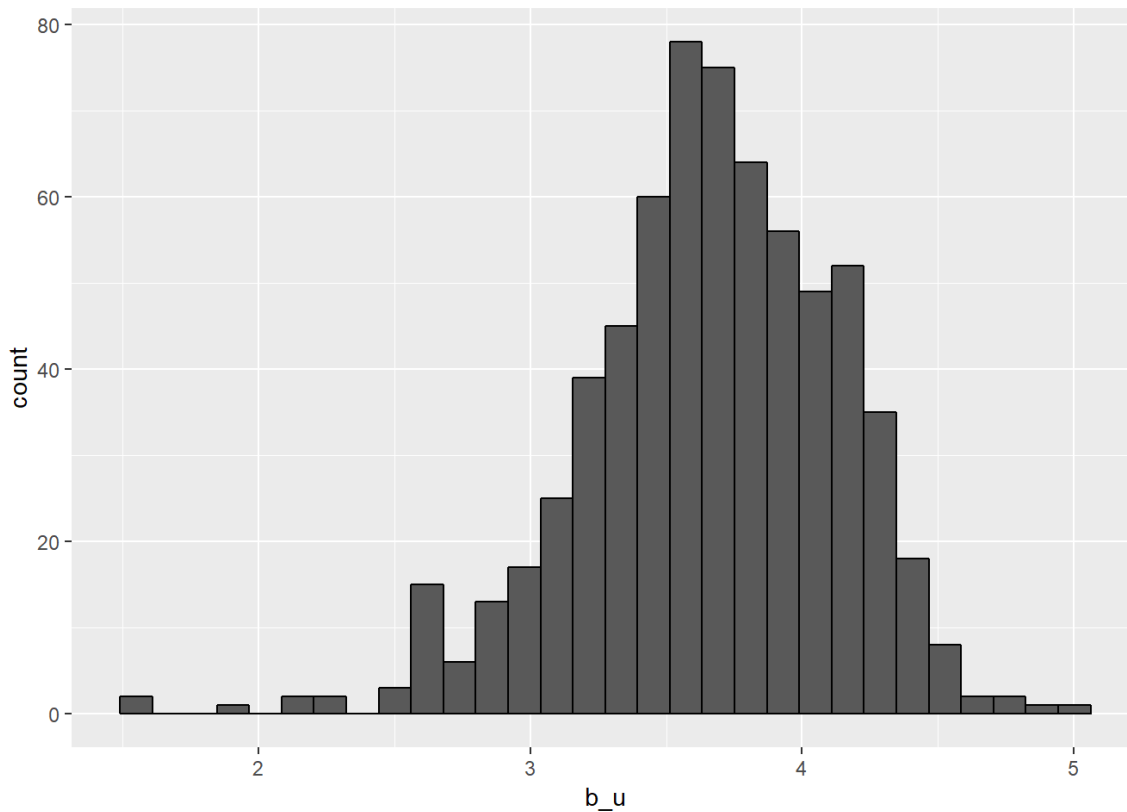
```
predicted_ratings <- mu + test_set %>%
  left_join(movie_avgs, by='movieId') %>%
  pull(b_i)

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 1.05
## 2 Movie Effect Model 0.986
```

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



```
#lm(rating ~ as.factor(movieId) + as.factor(userId))

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    1.05
## 2 Movie Effect Model  0.986
## 3 Movie + User Effects Model 0.908
```

```
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  residual
## 1      Day of the Beast, The (Día de la Bestia, El) 4.500000
## 2                                Horror Express -4.000000
## 3                                No Holds Barred  4.000000
## 4 Dear Zachary: A Letter to a Son About His Father -4.000000
## 5                                Faust -4.000000
## 6                                Hear My Song -4.000000
## 7                                Confessions of a Shopaholic -4.000000
## 8      Twilight Saga: Breaking Dawn - Part 1, The -4.000000
## 9                                Taxi Driver -3.806931
## 10                               Taxi Driver -3.806931
```

```
movie_titles <- movielens %>%
  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 Lamerica                          1.46
## 2 Love & Human Remains              1.46
## 3 Enfer, L'                        1.46
## 4 Picture Bride (Bijo photo)       1.46
## 5 Red Firecracker, Green Firecracker (Pao Da Shuang Deng) 1.46
## 6 Faces                            1.46
## 7 Maya Lin: A Strong Clear Vision   1.46
## 8 Heavy                             1.46
## 9 Gate of Heavenly Peace, The      1.46
## 10 Death in the Garden (Mort en ce jardin, La) 1.46
```

```
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 Santa with Muscles                -3.04
## 2 B*A*P*S                           -3.04
## 3 3 Ninjas: High Noon On Mega Mountain -3.04
## 4 Barney's Great Adventure           -3.04
## 5 Merry War, A                      -3.04
## 6 Day of the Beast, The (Día de la Bestia, El) -3.04
## 7 Children of the Corn III           -3.04
## 8 Whiteboyz                         -3.04
## 9 Catfish in Black Bean Sauce        -3.04
## 10 Watcher, The                     -3.04
```

```
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 Lamerica                        1.46     1
## 2 Love & Human Remains            1.46     3
## 3 Enfer, L'                      1.46     1
## 4 Picture Bride (Bijo photo)     1.46     1
## 5 Red Firecracker, Green Firecracker (Pao Da Shuang Deng) 1.46     3
## 6 Faces                          1.46     1
## 7 Maya Lin: A Strong Clear Vision 1.46     2
## 8 Heavy                          1.46     1
## 9 Gate of Heavenly Peace, The     1.46     1
## 10 Death in the Garden (Mort en ce jardin, La) 1.46     1
```

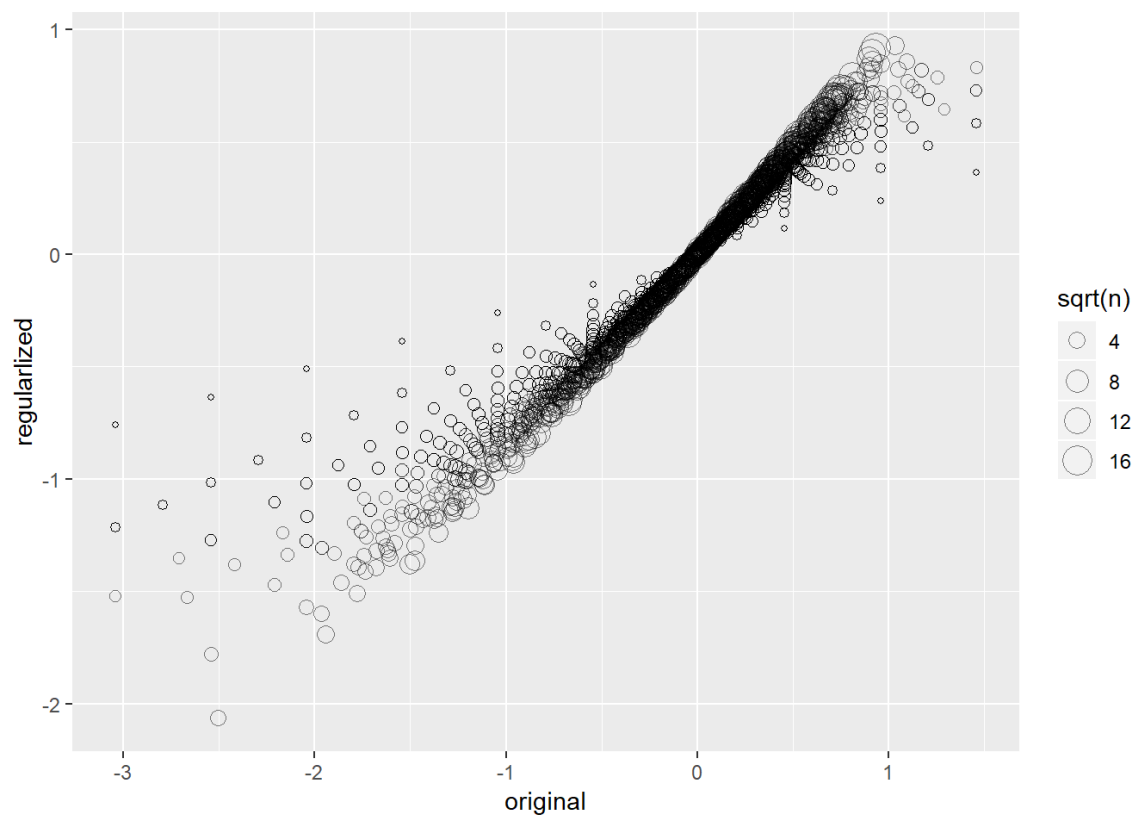
```
train_set %>% count(movieId) %>%
  left_join(movie_avgs) %>%
  left_join(movie_titles, by="movieId") %>%
  arrange(b_i) %>%
  select(title, b_i, n) %>%
  slice(1:10)
```

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

```
## # A tibble: 10 x 3
##   title                                b_i     n
##   <chr>                            <dbl> <int>
## 1 Santa with Muscles             -3.04     1
## 2 B*A*P*S                       -3.04     1
## 3 3 Ninjas: High Noon On Mega Mountain -3.04     1
## 4 Barney's Great Adventure       -3.04     1
## 5 Merry War, A                   -3.04     1
## 6 Day of the Beast, The (Día de la Bestia, El) -3.04     1
## 7 Children of the Corn III       -3.04     1
## 8 Whiteboyz                     -3.04     1
## 9 Catfish in Black Bean Sauce    -3.04     1
## 10 Watcher, The                  -3.04     1
```

```
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$b_i,
  regularized = movie_reg_avgs$b_i,
  n = movie_reg_avgs$n_i) %>%
  ggplot(aes(original, regularized, 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 All About Eve                       0.927   26
## 2 Shawshank Redemption, The          0.921  240
## 3 Godfather, The                     0.897  153
## 4 Godfather: Part II, The            0.871  100
## 5 Maltese Falcon, The                0.860   47
## 6 Best Years of Our Lives, The       0.859   11
## 7 On the Waterfront                  0.847   23
## 8 Face in the Crowd, A               0.833    4
## 9 African Queen, The                 0.832   36
## 10 All Quiet on the Western Front    0.824   11
```

```
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 Battlefield Earth   -2.06    14
## 2 Joe's Apartment     -1.78     7
## 3 Speed 2: Cruise Control -1.69    20
## 4 Super Mario Bros.   -1.60    13
## 5 Police Academy 6: City Under Siege -1.57    10
## 6 After Earth         -1.52     4
## 7 Disaster Movie      -1.52     3
## 8 Little Nicky        -1.51    17
## 9 Cats & Dogs         -1.47     6
## 10 Blade: Trinity     -1.46    11
```

```
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))

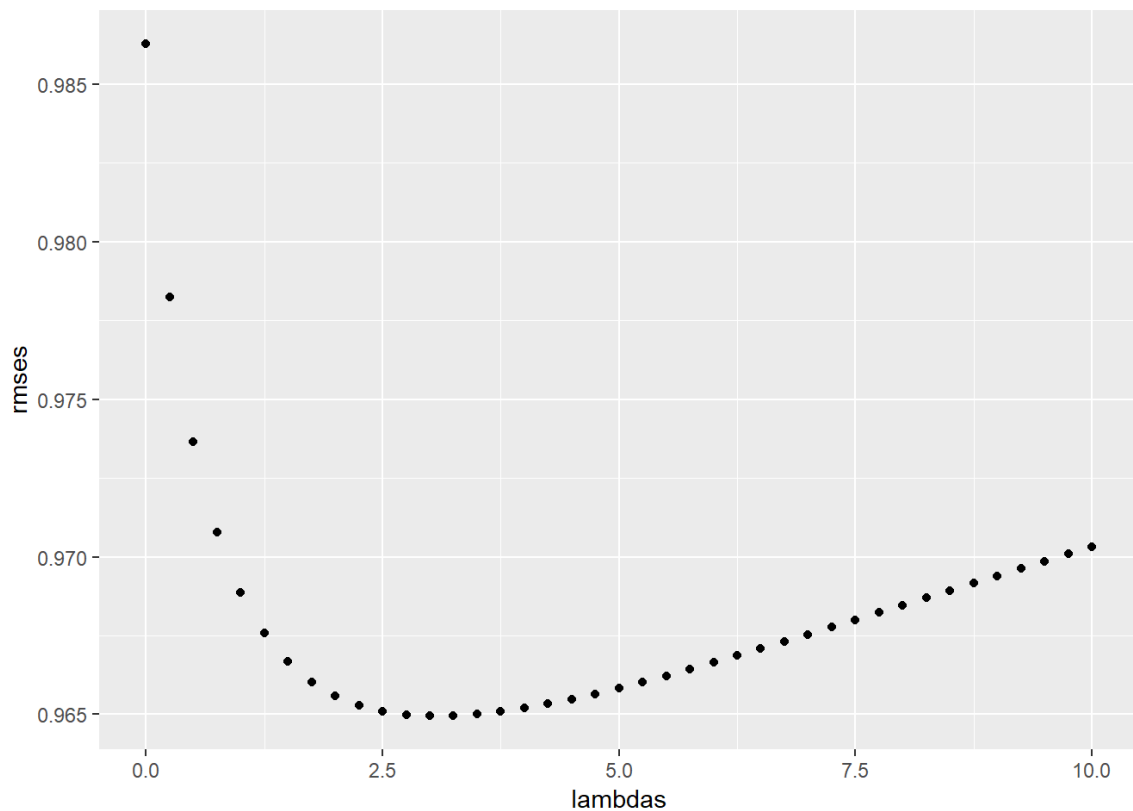
rmse_results
```

```
## # A tibble: 4 x 2
##   method                RMSE
##   <chr>              <dbl>
## 1 Just the average      1.05
## 2 Movie Effect Model    0.986
## 3 Movie + User Effects Model 0.908
## 4 Regularized Movie Effect Model 0.965
```

```
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+1)) %>%
    mutate(pred = mu + b_i) %>%
    pull(pred)
  return(RMSE(predicted_ratings, test_set$rating))
})
qplot(lambdas, rmses)
```

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

```
## [1] 3
```

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

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

  mu <- mean(train_set$rating)

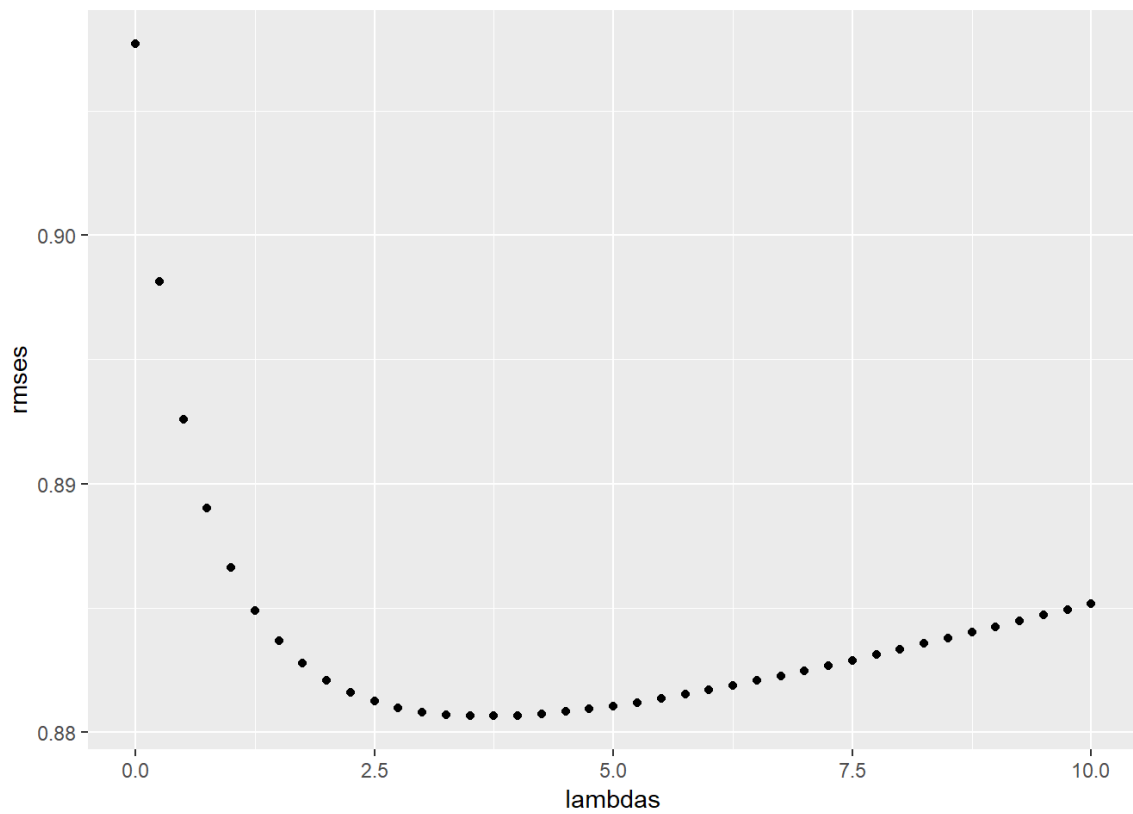
  b_i <- train_set %>%
    group_by(movieId) %>%
    summarize(b_i = sum(rating - mu)/(n()+1))

  b_u <- train_set %>%
    left_join(b_i, by="movieId") %>%
    group_by(userId) %>%
    summarize(b_u = sum(rating - b_i - mu)/(n()+1))

  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] 3.75
```

```
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	1.0482202
Movie Effect Model	0.9862839
Movie + User Effects Model	0.9077043
Regularized Movie Effect Model	0.9649457
Regularized Movie + User Effect Model	0.8806419