Movielens Capstone Project HarvardX

MovieLens Introduction

The MovieLens data set was collected by GroupLens Research. Can we predict movie ratings based on user preferance, age of a movie? Using the MovieLens data set and penalized least squares, the following R script calculates the RMSE based on user ratings, movieId and the age of the movie.

The MovieLens data set contains 10000054 rows, 10677 movies, 797 genres and 69878 users.

The steps performed for analysis of the data - Created an age of movie column - Graphic displays of movie, users and ratings in order to find a pattern or insight to the
behavior of the data. - Explored Genres to determine if ratings could be predicted by genre. - Explored the Coefficient of Determination R-Squared - Graphically explored the linear correlation coefficient, r-value - Calculate RMSE based on movieId, userId, and age of the movie.

After exploring the movies through graphical representations and calculating RMSE, I found the best predictor for ratings was movieId, userId. The age of the movie didn't change the rmse.

The final RMSE is 0.8252

The following are the libraries I used to explore the data. Explorations that didn't seem to lead to an insight were taken out of the script.

DownLoad the data

```
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"))
```

Build the data set

```
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))
```

#Explore the size of the data set

```
movielens <- left_join(ratings, movies, by = "movieId")
nrow(movielens)
## [1] 10000054
n_distinct(movielens$movieId)
## [1] 10677
n_distinct(movielens$genres)
## [1] 797
n_distinct(movielens$userId)
## [1] 69878
```

#Validation set will be 10% of the 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")
edx <- rbind(edx, removed)
```

##Data Cleaning and, data exploration and Data Visulization #In order to determine if age of the movie is a factor for predicting rating, I extracted the premier date of the movie, and then calculated the age of the movie. I will also looked at individual genres for genre effect, as well as, effects of user ratings.

```
head(edx)
##   userId movieId rating timestamp                         title
## 1      1     122      5 838985046              Boomerang (1992)
## 2      1     185      5 838983525               Net, The (1995)
## 4      1     292      5 838983421               Outbreak (1995)
## 5      1     316      5 838983392               Stargate (1994)
## 6      1     329      5 838983392 Star Trek: Generations (1994)
## 7      1     355      5 838984474       Flintstones, The (1994)
##                            genres
## 1                Comedy|Romance
## 2         Action|Crime|Thriller
## 4  Action|Drama|Sci-Fi|Thriller
## 5       Action|Adventure|Sci-Fi
## 6 Action|Adventure|Drama|Sci-Fi
## 7       Children|Comedy|Fantasy
glimpse(edx)
## Observations: 9,000,055
## Variables: 6
## $ userId    <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ movieId   <dbl> 122, 185, 292, 316, 329, 355, 356, 362, 364, 370, 37...
```

```
## $ rating    <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5...
## $ timestamp <int> 838985046, 838983525, 838983421, 838983392, 83898339...
## $ title     <chr> "Boomerang (1992)", "Net, The (1995)", "Outbreak (19...
## $ genres    <chr> "Comedy|Romance", "Action|Crime|Thriller", "Action|D...
```

#How many distinct movie, users and genres

```
n_distinct(edx$movieId)
## [1] 10677
n_distinct(edx$genres)
## [1] 797
n_distinct(edx$userId)
## [1] 69878
nrow(edx)
## [1] 9000055
```

#Convert Timestamp to year

```
edx <- mutate(edx, year_rated = year(as_datetime(timestamp)))
head(edx)
##   userId movieId rating timestamp                         title
## 1      1     122      5 838985046              Boomerang (1992)
## 2      1     185      5 838983525               Net, The (1995)
## 3      1     292      5 838983421               Outbreak (1995)
## 4      1     316      5 838983392               Stargate (1994)
## 5      1     329      5 838983392 Star Trek: Generations (1994)
## 6      1     355      5 838984474       Flintstones, The (1994)
##                            genres year_rated
## 1                  Comedy|Romance       1996
## 2           Action|Crime|Thriller       1996
## 3  Action|Drama|Sci-Fi|Thriller       1996
## 4       Action|Adventure|Sci-Fi       1996
## 5 Action|Adventure|Drama|Sci-Fi       1996
## 6       Children|Comedy|Fantasy       1996
```

Extract the premier date and calculate the age of the movie. Explore whether or not the age of the movie effects predicted ratings

#extracting the premier date

```
premier <- stringi::stri_extract(edx$title, regex = "(\\d{4})", comments =
TRUE ) %>% as.numeric()
```

#Add the premier date

```
edx_with_title_dates <- edx %>% mutate(premier_date = premier)
head(edx_with_title_dates)
##   userId movieId rating timestamp                          title
## 1      1     122      5 838985046              Boomerang (1992)
## 2      1     185      5 838983525               Net, The (1995)
## 3      1     292      5 838983421               Outbreak (1995)
## 4      1     316      5 838983392               Stargate (1994)
## 5      1     329      5 838983392 Star Trek: Generations (1994)
## 6      1     355      5 838984474       Flintstones, The (1994)
##                            genres year_rated premier_date
## 1                  Comedy|Romance       1996         1992
## 2           Action|Crime|Thriller       1996         1995
## 3  Action|Drama|Sci-Fi|Thriller       1996         1995
## 4       Action|Adventure|Sci-Fi       1996         1994
## 5 Action|Adventure|Drama|Sci-Fi       1996         1994
## 6       Children|Comedy|Fantasy       1996         1994
```

#After extracting the premier date from the title, check for accuracy

#drop the timestamp

```
edx_with_title_dates <- edx_with_title_dates %>% select(-timestamp)

head(edx_with_title_dates)
##   userId movieId rating                          title
## 1      1     122      5              Boomerang (1992)
## 2      1     185      5               Net, The (1995)
## 3      1     292      5               Outbreak (1995)
```

```
## 4      1     316      5               Stargate (1994)
## 5      1     329      5 Star Trek: Generations (1994)
## 6      1     355      5       Flintstones, The (1994)
##                          genres year_rated premier_date
## 1                Comedy|Romance       1996         1992
## 2         Action|Crime|Thriller       1996         1995
## 3  Action|Drama|Sci-Fi|Thriller       1996         1995
## 4       Action|Adventure|Sci-Fi       1996         1994
## 5 Action|Adventure|Drama|Sci-Fi       1996         1994
## 6       Children|Comedy|Fantasy       1996         1994
```

#looking at the dates - are they correct?

```
edx_with_title_dates %>% filter(premier_date > 2018) %>% group_by(movieId,
title, premier_date) %>% summarize(n = n())
## # A tibble: 6 x 4
## # Groups:   movieId, title [?]
##   movieId title                                          premier_date
n
##     <dbl> <chr>                                                 <dbl>
<int>
## 1     671 Mystery Science Theater 3000: The Movie (1996)         3000
3280
## 2    2308 Detroit 9000 (1973)                                    9000
22
## 3    4159 3000 Miles to Graceland (2001)                         3000
714
## 4    5310 Transylvania 6-5000 (1985)                             5000
195
## 5    8864 Mr. 3000 (2004)                                        3000
146
## 6   27266 2046 (2004)                                           2046
426
edx_with_title_dates %>% filter(premier_date < 1900) %>% group_by(movieId,
title, premier_date) %>% summarize(n = n())
## # A tibble: 8 x 4
## # Groups:   movieId, title [?]
##   movieId title                                       premier_date     n
```

```
##     <dbl> <chr>                                                <dbl> <int>
## 1    1422 Murder at 1600 (1997)                                1600  1566
## 2    4311 Bloody Angels (1732 Høtten: Marerittet Har e…        1732     9
## 3    5472 1776 (1972)                                          1776   185
## 4    6290 House of 1000 Corpses (2003)                         1000   367
## 5    6645 THX 1138 (1971)                                      1138   464
## 6    8198 1000 Eyes of Dr. Mabuse, The (Tausend Augen …        1000    24
## 7    8905 1492: Conquest of Paradise (1992)                    1492   134
## 8   53953 1408 (2007)                                          1408   466
```

#Fix the incorrect dates

```
edx_with_title_dates[edx_with_title_dates$movieId == "27266", "premier_date"]
<- 2004
edx_with_title_dates[edx_with_title_dates$movieId == "671", "premier_date"]
<- 1996
edx_with_title_dates[edx_with_title_dates$movieId == "2308", "premier_date"]
<- 1973
edx_with_title_dates[edx_with_title_dates$movieId == "4159", "premier_date"]
<- 2001
edx_with_title_dates[edx_with_title_dates$movieId == "5310", "premier_date"]
<- 1985
edx_with_title_dates[edx_with_title_dates$movieId == "8864", "premier_date"]
<- 2004
edx_with_title_dates[edx_with_title_dates$movieId == "1422", "premier_date"]
<- 1997
edx_with_title_dates[edx_with_title_dates$movieId == "4311", "premier_date"]
<- 1998
edx_with_title_dates[edx_with_title_dates$movieId == "5472", "premier_date"]
<- 1972
edx_with_title_dates[edx_with_title_dates$movieId == "6290", "premier_date"]
<- 2003
edx_with_title_dates[edx_with_title_dates$movieId == "6645", "premier_date"]
<- 1971
edx_with_title_dates[edx_with_title_dates$movieId == "8198", "premier_date"]
<- 1960
edx_with_title_dates[edx_with_title_dates$movieId == "8905", "premier_date"]
<- 1992
```

```
edx_with_title_dates[edx_with_title_dates$movieId == "53953", "premier_date"]
<- 2007
```

#Calculate the age of the movie

#Calculate the age of a movie

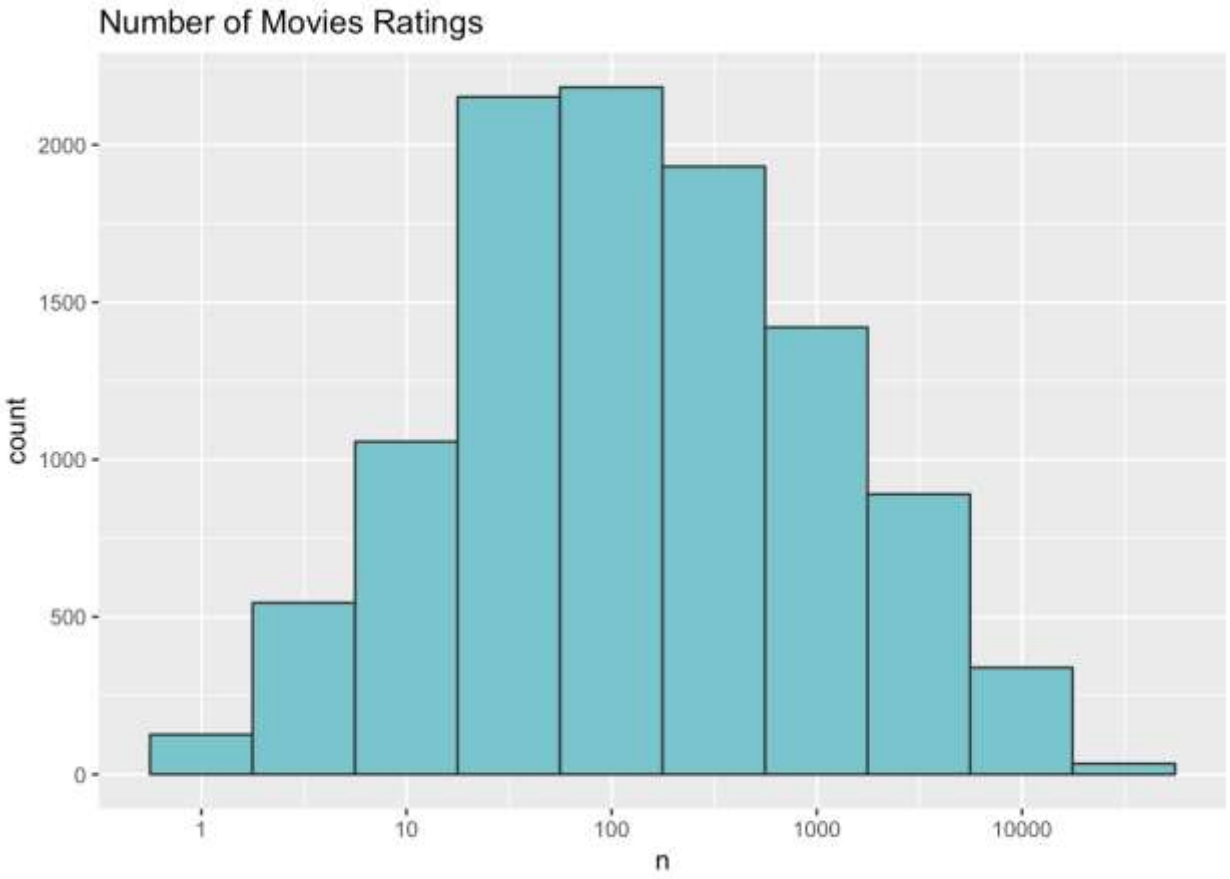
```
edx_with_title_dates <- edx_with_title_dates %>% mutate(age_of_movie = 2018 -
premier_date,
                                                        rating_date_range =
year_rated - premier_date)
head(edx_with_title_dates)
##   userId movieId rating                         title
## 1      1     122      5              Boomerang (1992)
## 2      1     185      5               Net, The (1995)
## 3      1     292      5               Outbreak (1995)
## 4      1     316      5               Stargate (1994)
## 5      1     329      5 Star Trek: Generations (1994)
## 6      1     355      5       Flintstones, The (1994)
##                          genres year_rated premier_date age_of_movie
## 1                Comedy|Romance       1996         1992           26
## 2          Action|Crime|Thriller       1996         1995           23
## 3  Action|Drama|Sci-Fi|Thriller       1996         1995           23
## 4       Action|Adventure|Sci-Fi       1996         1994           24
## 5 Action|Adventure|Drama|Sci-Fi       1996         1994           24
## 6       Children|Comedy|Fantasy       1996         1994           24
##   rating_date_range
## 1                 4
## 2                 1
## 3                 1
## 4                 2
## 5                 2
## 6                 2
```

Graph the data

#Distribution of Movie Ratings

```
edx %>% group_by(movieId) %>% summarize(n = n()) %>%
```

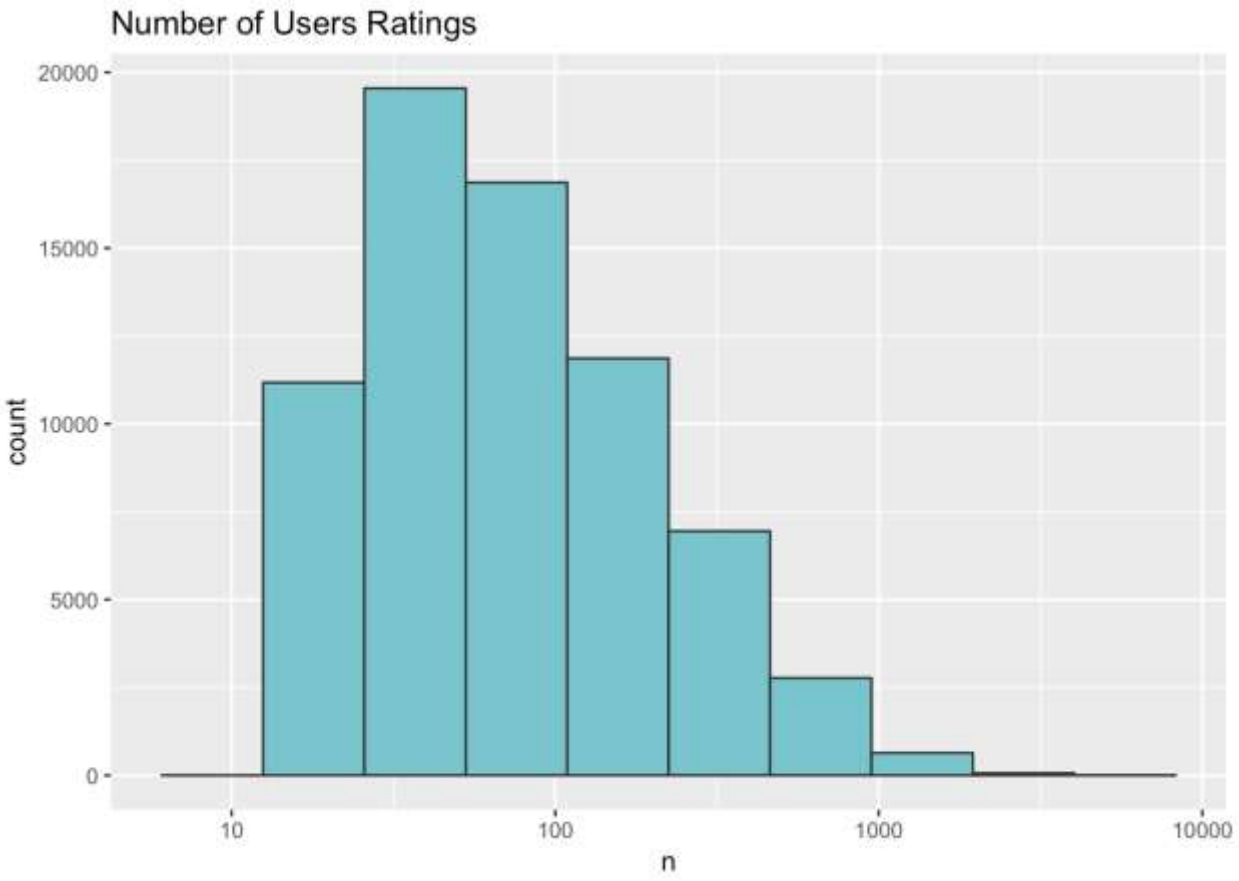
```
  ggplot(aes(n)) + geom_histogram(fill = "cadetblue3", color = "grey20", bins
= 10) +
  scale_x_log10() +
  ggtitle("Number of Movies Ratings")
```



#Number of Ratings by userId

#Distribution of Users

```
edx %>% group_by(userId) %>% summarize(n = n()) %>%
  ggplot(aes(n)) + geom_histogram(fill = "cadetblue3", color = "grey20", bins
= 10) +
  scale_x_log10() +
  ggtitle("Number of Users Ratings")
```



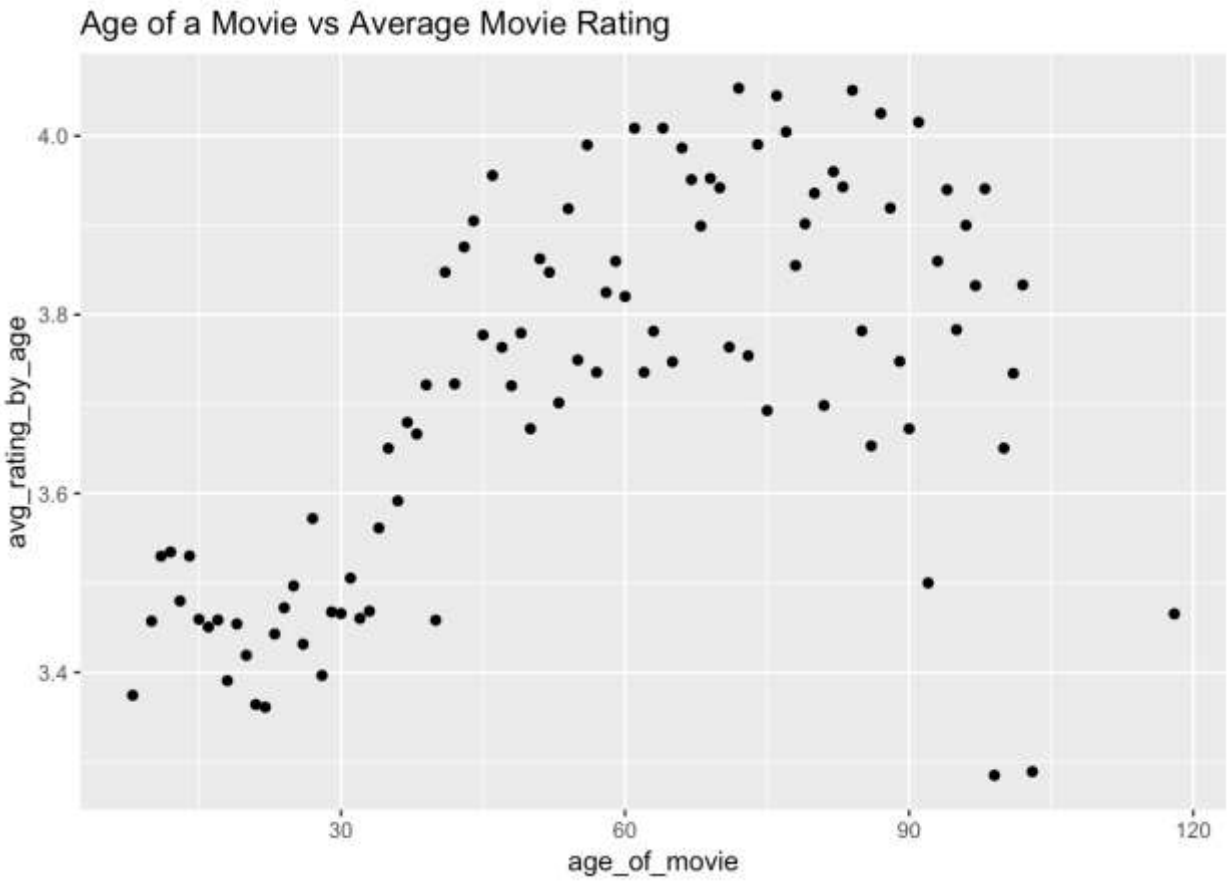
Calculate movie rating average, user rating average, average rating by age of movie, average rating by year

```
#Movie rating averages
movie_avgs <- edx_with_title_dates %>% group_by(movieId) %>%
summarize(avg_movie_rating = mean(rating))
user_avgs <- edx_with_title_dates %>% group_by(userId) %>%
summarize(avg_user_rating = mean(rating))
year_avgs <- edx_with_title_dates%>% group_by(year_rated) %>%
summarize(avg_rating_by_year = mean(rating)) #year the movie was rated
age_avgs <- edx_with_title_dates %>% group_by(age_of_movie) %>%
summarize(avg_rating_by_age = mean(rating)) #age of movie
head(age_avgs)
## # A tibble: 6 x 2
##   age_of_movie avg_rating_by_age
```

```
##          <dbl>             <dbl>
## 1            8              3.37
## 2           10              3.46
## 3           11              3.53
## 4           12              3.53
## 5           13              3.48
## 6           14              3.53
head(user_avgs)
## # A tibble: 6 x 2
##   userId avg_user_rating
##    <int>           <dbl>
## 1      1            5
## 2      2            3.29
## 3      3            3.94
## 4      4            4.06
## 5      5            3.92
## 6      6            3.95
```

#What is the relationship to the age of a movie and the movies average rating?
#Graph age of movie vs average movie rating

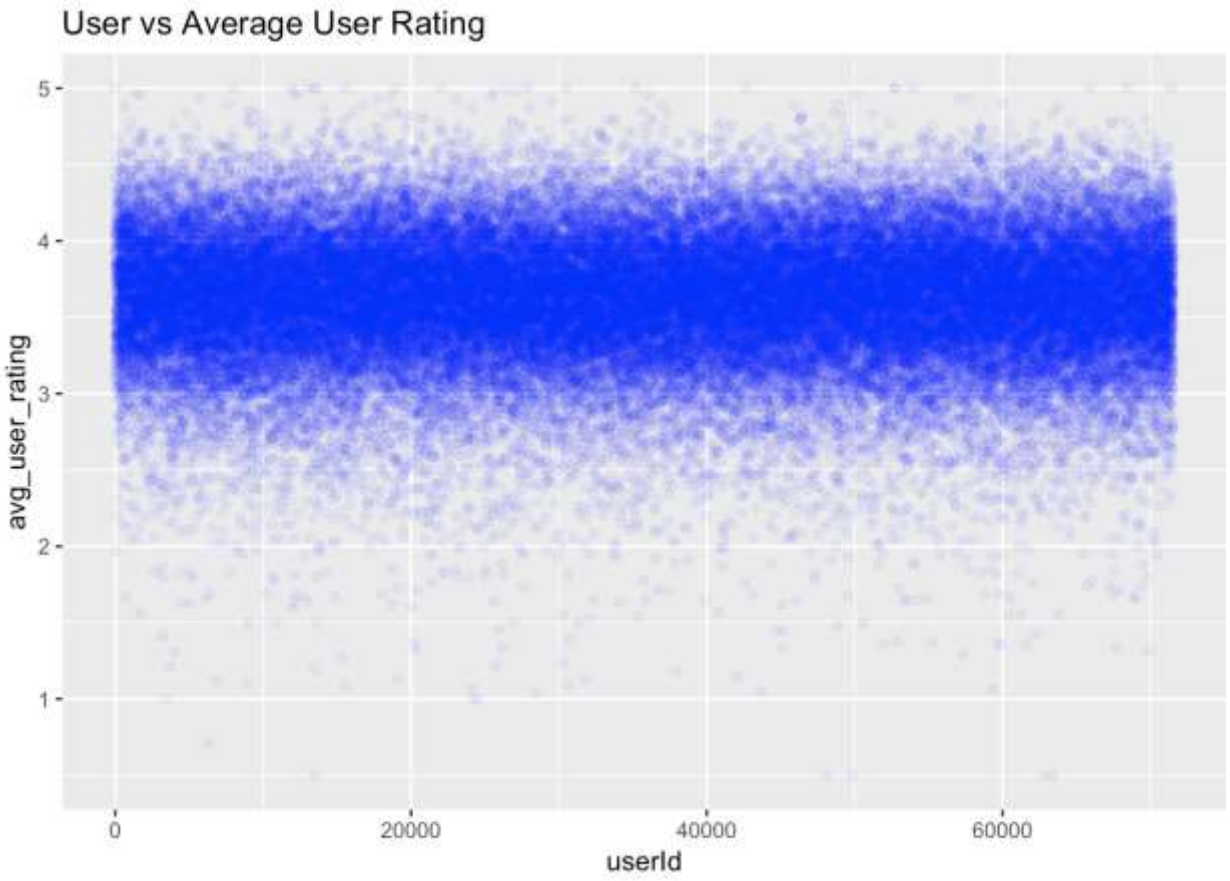
```
# age of movie vs average movie rating
age_avgs %>%
  ggplot(aes(age_of_movie, avg_rating_by_age)) +
  geom_point() +
  ggtitle("Age of a Movie vs Average Movie Rating")
```



The above plot shows more variability as movies age. The plot, also, shows higher ratings the older a movies is up to 90 years old, then the ratings drop.

 # userId vs average movie rating

```
user_avgs %>%
  ggplot(aes(userId, avg_user_rating)) +
  geom_point(alpha = 1/20, colour = "blue") +
  ggtitle("User vs Average User Rating")
```



#From the above graph, we can see average ratings by user are pretty consistent between 2.5 and 4.5

#Calculating the lm of the age of a movie vs average rating

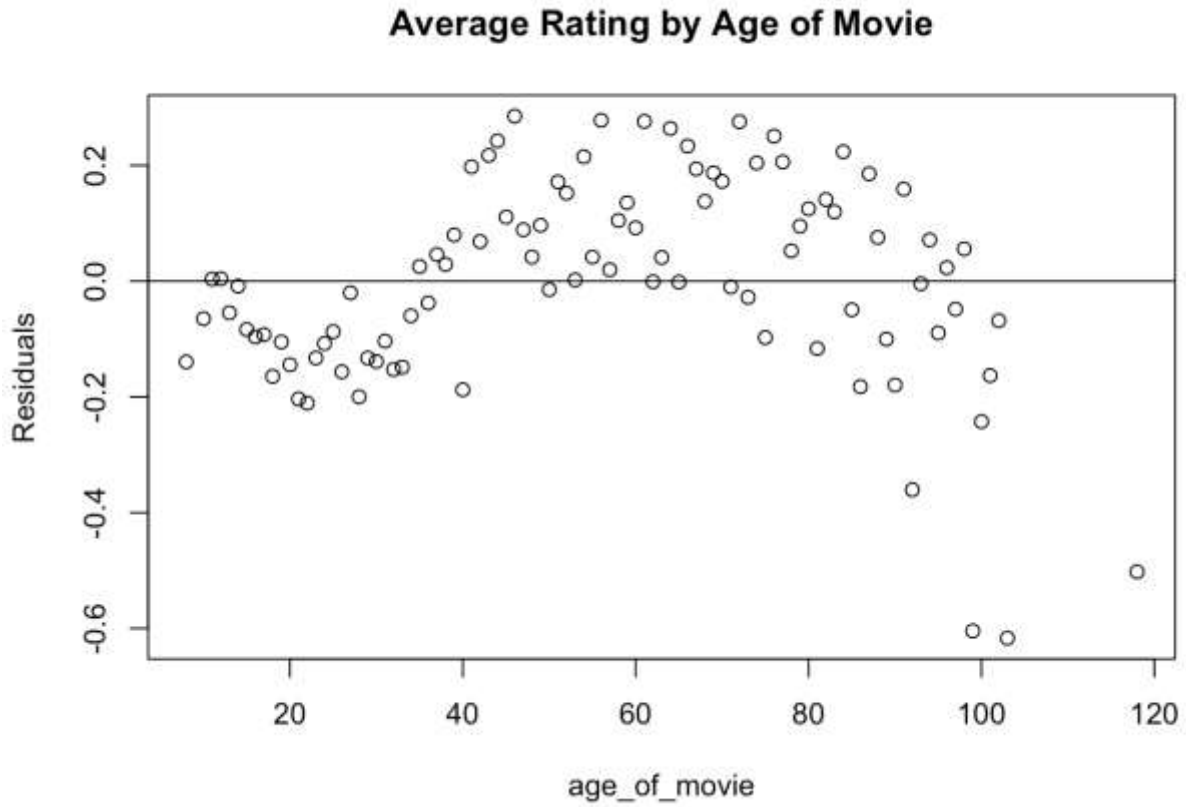
```
summary(lm(avg_rating_by_age ~ age_of_movie, data = age_avgs))
##
## Call:
## lm(formula = avg_rating_by_age ~ age_of_movie, data = age_avgs)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.61684 -0.10389  0.00276  0.12759  0.28508
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.4809443  0.0409983  84.905  < 2e-16 ***
## age_of_movie 0.0041241  0.0006489   6.356 7.38e-09 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1781 on 94 degrees of freedom
## Multiple R-squared:  0.3006, Adjusted R-squared:  0.2931
## F-statistic:  40.4 on 1 and 94 DF,  p-value: 7.377e-09
```

#We can see that R-square is small at 0.30

#Plot the Residuals

```
avg_rating.lm <- lm(avg_rating_by_age ~ age_of_movie, data = age_avgs)
avg_rating.res <- resid(avg_rating.lm)

plot(age_avgs$age_of_movie, avg_rating.res,
     ylab='Residuals', xlab='age_of_movie',
     main = 'Average Rating by Age of Movie') + abline(0,0)
```



```
## integer(0)
```

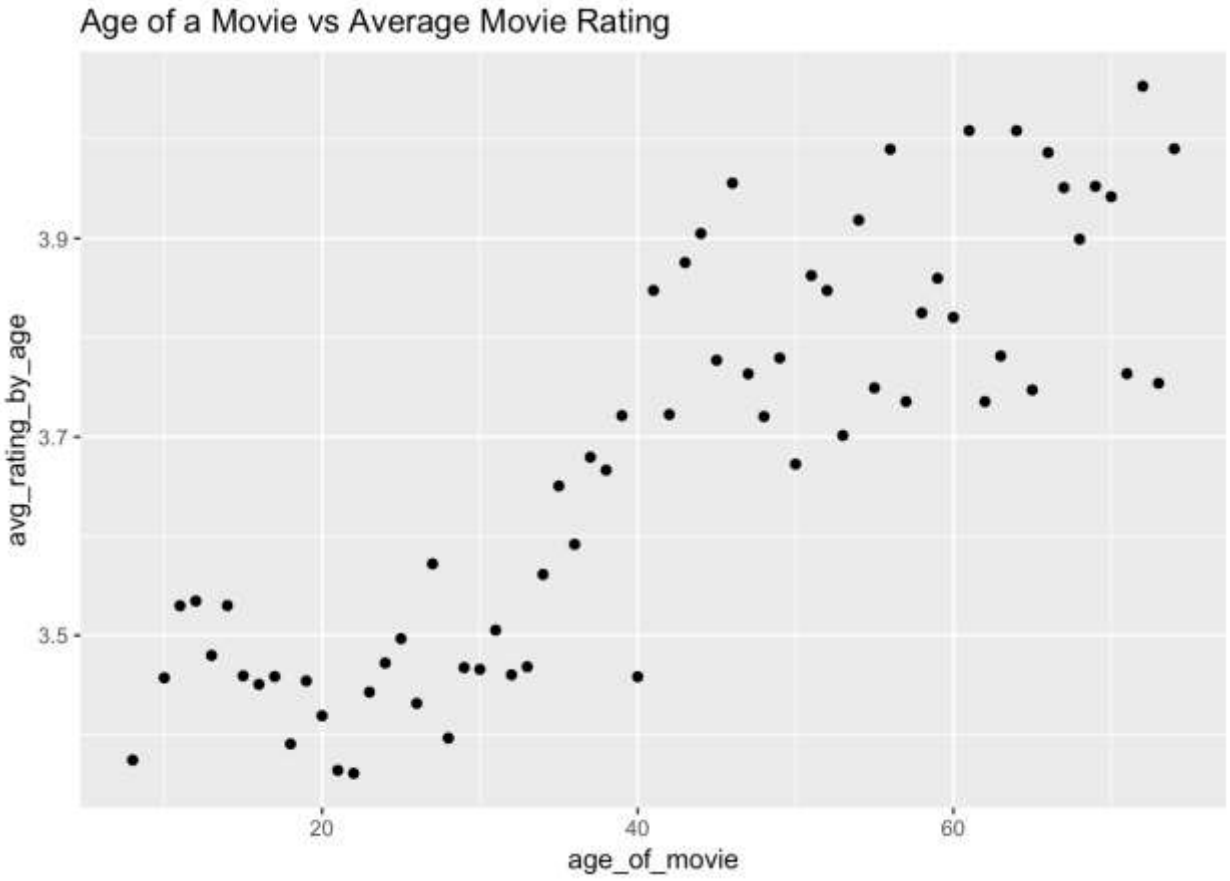
#The R-squared is fairly small at 0.30; 30% of the variation in movie ratings can be prdicted by # explore the data graphically to see if age of the movie and rating are coorelated

#Movies less than 75 years old

```
age_of_movie_less_than75 <- age_avgs %>% filter(age_of_movie <75)
```

 # age of movie less than 75 years old vs average movie rating

```
age_of_movie_less_than75 %>%
  ggplot(aes(age_of_movie, avg_rating_by_age)) +
  geom_point() +
  ggtitle("Age of a Movie vs Average Movie Rating")
```



#Calculate the R-squared value

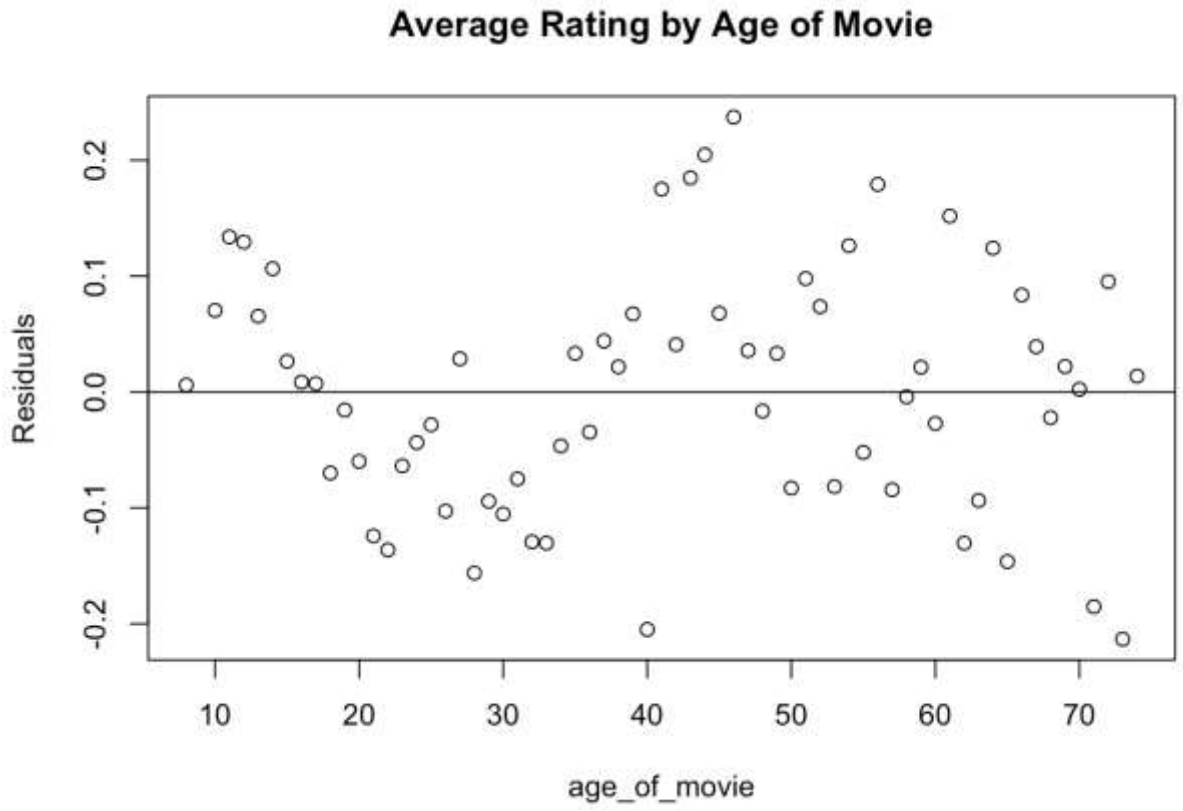
```
age_lessthan75_rating.lm <- lm(avg_rating_by_age ~ age_of_movie, data =
age_of_movie_less_than75)
summary(age_lessthan75_rating.lm)
##
## Call:
## lm(formula = avg_rating_by_age ~ age_of_movie, data =
age_of_movie_less_than75)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.21323 -0.07992  0.00663  0.06785  0.23721
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.2946266  0.0307644  107.09   <2e-16 ***
## age_of_movie 0.0092153  0.0006738   13.68   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1044 on 64 degrees of freedom
## Multiple R-squared:  0.7451, Adjusted R-squared:  0.7411
## F-statistic: 187.1 on 1 and 64 DF,  p-value: < 2.2e-16
```

#The R-squared increased to 0.745

#Plot the residuals

```
head(age_of_movie_less_than75)
## # A tibble: 6 x 2
##   age_of_movie avg_rating_by_age
##          <dbl>             <dbl>
## 1            8              3.37
## 2           10              3.46
## 3           11              3.53
## 4           12              3.53
## 5           13              3.48
## 6           14              3.53
```

```
age_lessthan75.res <- resid(age_lessthan75_rating.lm)

plot(age_of_movie_less_than75$age_of_movie, age_lessthan75.res,
     ylab='Residuals', xlab='age_of_movie',
     main = 'Average Rating by Age of Movie') + abline(0,0)
```

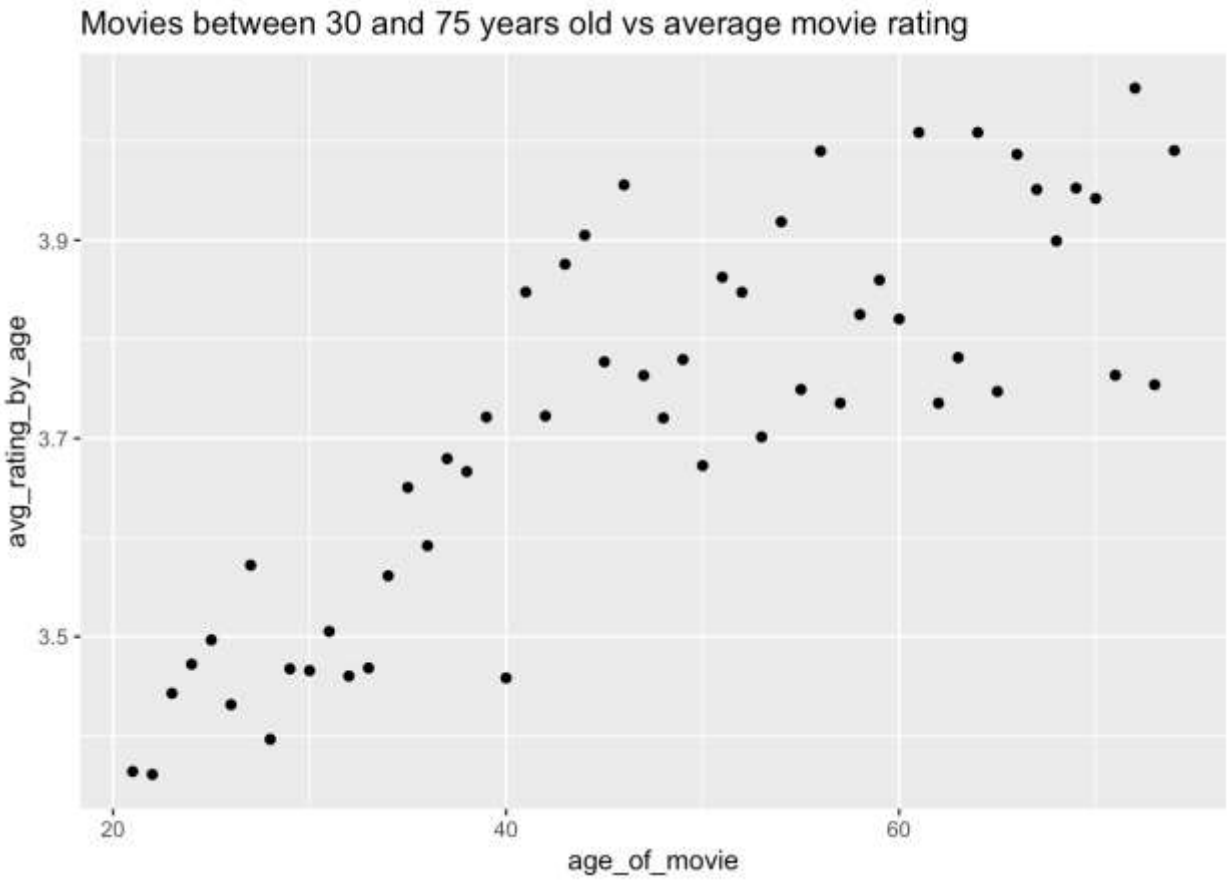



```
## integer(0)
```

#Let's look at moveies between 20 and 75 years old as the graph looks more linear in that time frame

```
#Movies between 20 and 75 years old
age_between20_and_75 <- age_avgs %>% filter((age_of_movie > 20) &
(age_of_movie < 75))
# graph the age of movie between 30 and 75 years old
age_between20_and_75 %>%
  ggplot(aes(age_of_movie, avg_rating_by_age)) +
```

```
  geom_point() + ggtitle("Movies between 30 and 75 years old vs average movie
rating")
```



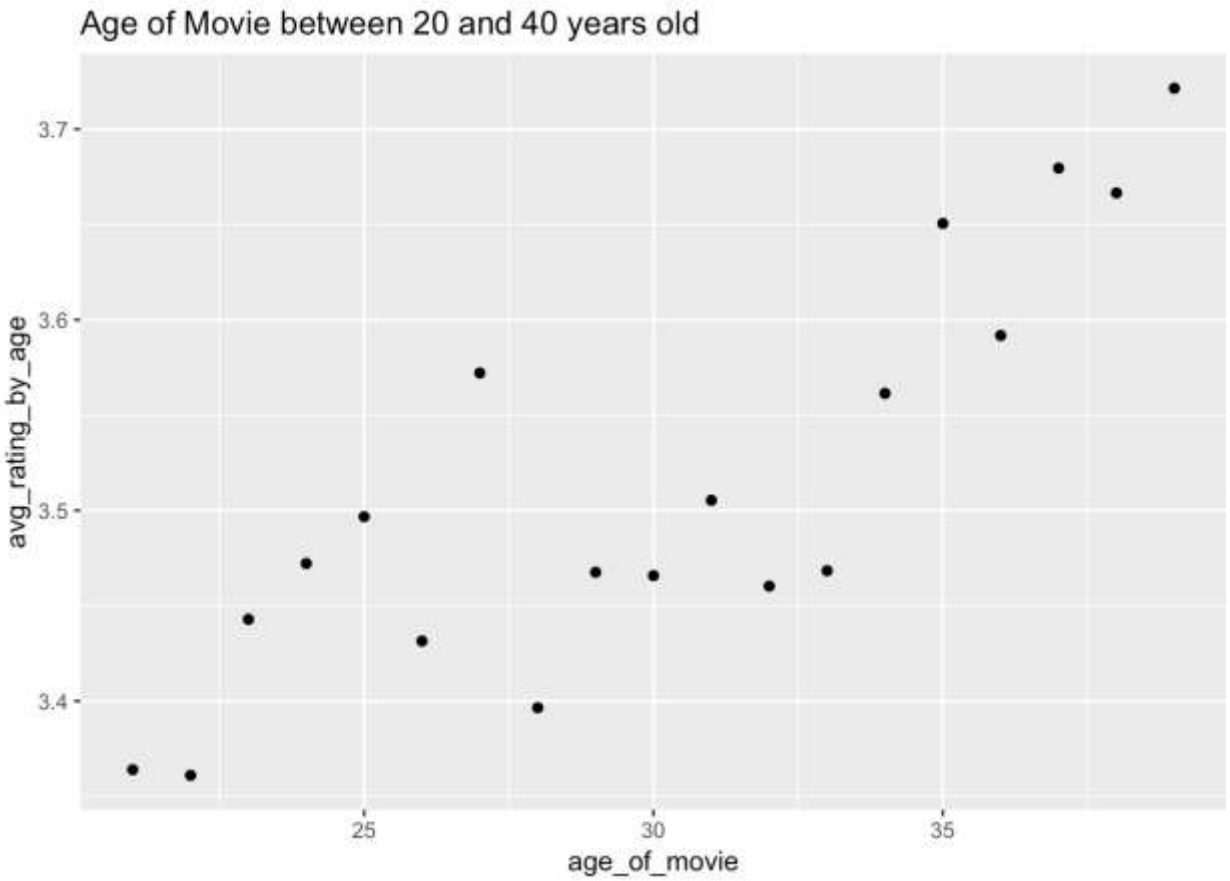
#The plot above appears to be a linear trend; however, the r-square is 0.69

```
summary(lm(avg_rating_by_age ~ age_of_movie, data = age_between20_and_75))
##
## Call:
## lm(formula = avg_rating_by_age ~ age_of_movie, data =
age_between20_and_75)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.235567 -0.077940 -0.009169  0.068137  0.246532
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.2313562  0.0473472   68.25  < 2e-16 ***
```

```
## age_of_movie 0.0103880  0.0009471   10.97 3.88e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1085 on 52 degrees of freedom
## Multiple R-squared:  0.6982, Adjusted R-squared:  0.6924
## F-statistic: 120.3 on 1 and 52 DF,  p-value: 3.882e-15
```

#The R-squared value is lower at 0.6981

```
# graph the age of movie between 20 and 40 years old
age_between20_and_40 <- age_avgs %>% filter((age_of_movie > 20) &
(age_of_movie < 40))

age_between20_and_40 %>%
  ggplot(aes(age_of_movie, avg_rating_by_age)) +
  geom_point() + ggtitle('Age of Movie between 20 and 40 years old')
```



#The above graph is displying a linear trend with older movies having higher ratings

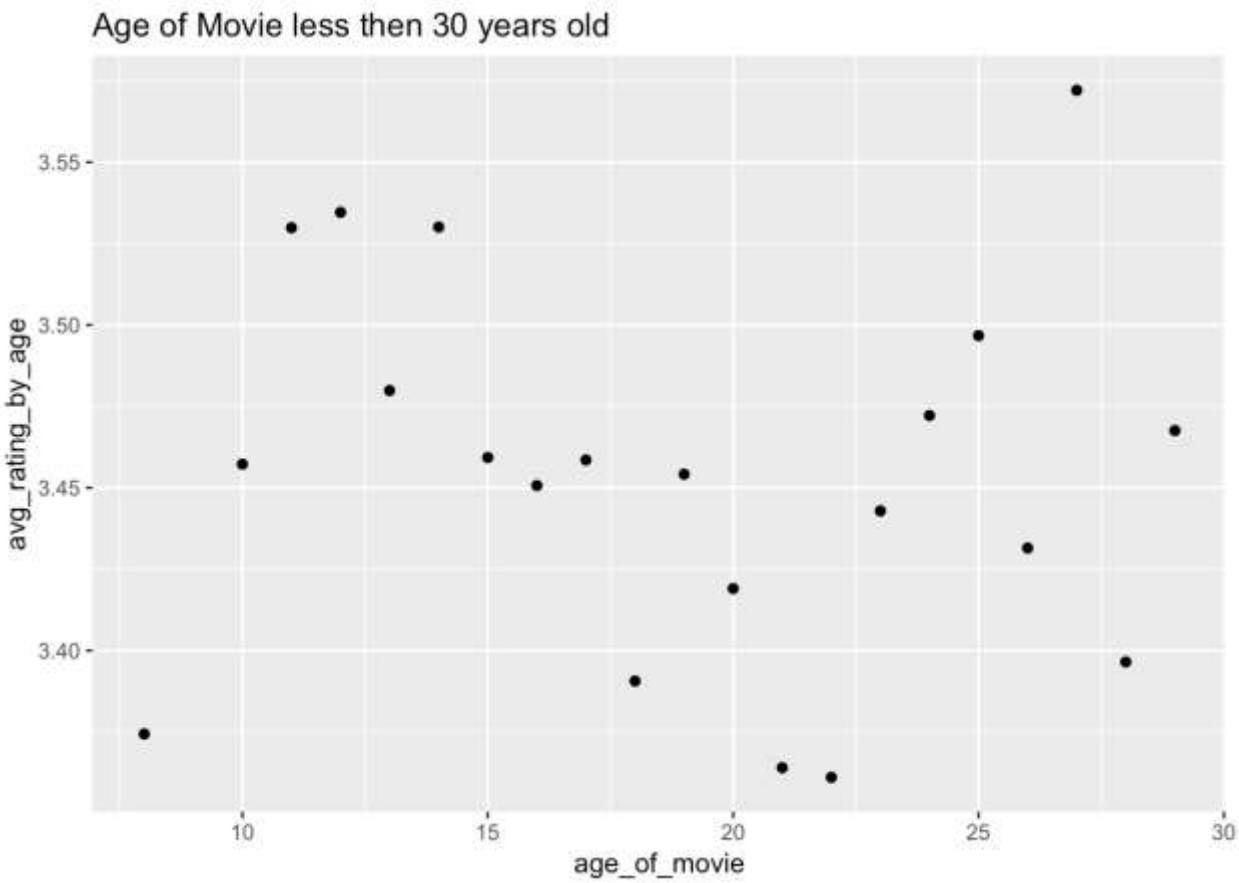
#calculate a linear model

```
summary(lm(avg_rating_by_age ~ age_of_movie, data = age_between20_and_40))
##
## Call:
## lm(formula = avg_rating_by_age ~ age_of_movie, data =
age_between20_and_40)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.09444 -0.02806 -0.01751  0.05332  0.10592
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.031416   0.075493  40.155  < 2e-16 ***
## age_of_movie 0.016103   0.002476   6.505 5.39e-06 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0591 on 17 degrees of freedom
## Multiple R-squared:  0.7134, Adjusted R-squared:  0.6965
## F-statistic: 42.31 on 1 and 17 DF,  p-value: 5.393e-06
```

The R-squared value is much higher than at 0.71

```
#Movies between 0 and 30 years old
age_less_than30 <- age_avgs %>% filter((age_of_movie < 30))

#Graph movies less than 30 years old and average movie rating
age_less_than30 %>%
  ggplot(aes(age_of_movie, avg_rating_by_age)) +
  geom_point() + ggtitle('Age of Movie less then 30 years old')
```



#For movies less than 30 years old there appears to be quite a bit of variation. We can see from the linear model that r-squared is nearly zero.

```
summary(lm(avg_rating_by_age ~ age_of_movie, data = age_less_than30))
##
## Call:
## lm(formula = avg_rating_by_age ~ age_of_movie, data = age_less_than30)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.091058 -0.034589 -0.000233  0.021613  0.123826
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.4688469  0.0420239  82.545   <2e-16 ***
## age_of_movie -0.0007611  0.0021095  -0.361    0.722
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.05933 on 19 degrees of freedom
## Multiple R-squared:  0.006805,   Adjusted R-squared:  -0.04547
## F-statistic: 0.1302 on 1 and 19 DF,  p-value: 0.7222
```

#The age of a movie did seem to effect the outcome of the average rating. This is possibly due to a higher number of ratings for older movies.

#Do Genres have an effect on ratings? ##I extracted the genres from the data with the idea to do an analysis on each genre. Some of the exploration I did here was removed as it didn't appear to effect the RMSE and this analysis keep growing! But I did get some nice graphs pertaining to genres.

#Genres split the data into single genres

```
dat <- edx_with_title_dates %>% separate_rows(genres, sep ="\\|")
```

```
head(dat)
##   userId movieId rating            title   genres year_rated premier_date
## 1      1     122      5 Boomerang (1992)   Comedy       1996         1992
## 2      1     122      5 Boomerang (1992)  Romance       1996         1992
## 3      1     185      5  Net, The (1995)   Action       1996         1995
## 4      1     185      5  Net, The (1995)    Crime       1996         1995
## 5      1     185      5  Net, The (1995) Thriller       1996         1995
## 6      1     292      5  Outbreak (1995)   Action       1996         1995
##   age_of_movie rating_date_range
## 1           26                 4
## 2           26                 4
## 3           23                 1
## 4           23                 1
## 5           23                 1
## 6           23                 1
```

#Count the number of movies using movieId in each genre

```
genre_count_by_movieId <- dat %>% group_by(movieId, genres) %>% summarize(n =
n())
head(genre_count_by_movieId)
```

```
## # A tibble: 6 x 3
## # Groups:   movieId [2]
##   movieId genres        n
##     <dbl> <chr>     <int>
## 1       1 Adventure 23790
## 2       1 Animation 23790
## 3       1 Children  23790
## 4       1 Comedy    23790
## 5       1 Fantasy   23790
## 6       2 Adventure 10779
```

#Total number of movies in each genre

```
number_of_genres <- dat %>% group_by(genres) %>% summarize(n = n())
number_of_genres
## # A tibble: 20 x 2
##    genres                   n
##    <chr>                <int>
##  1 (no genres listed)       7
##  2 Action             2560545
##  3 Adventure          1908892
##  4 Animation           467168
##  5 Children            737994
##  6 Comedy             3540930
##  7 Crime              1327715
##  8 Documentary          93066
##  9 Drama              3910127
## 10 Fantasy            925637
## 11 Film-Noir          118541
## 12 Horror             691485
## 13 IMAX                 8181
## 14 Musical            433080
## 15 Mystery            568332
## 16 Romance           1712100
## 17 Sci-Fi            1341183
## 18 Thriller          2325899
## 19 War                511147
```

```
## 20 Western             189394
```

#List the genres. Movies are either in one genre or multiple genres

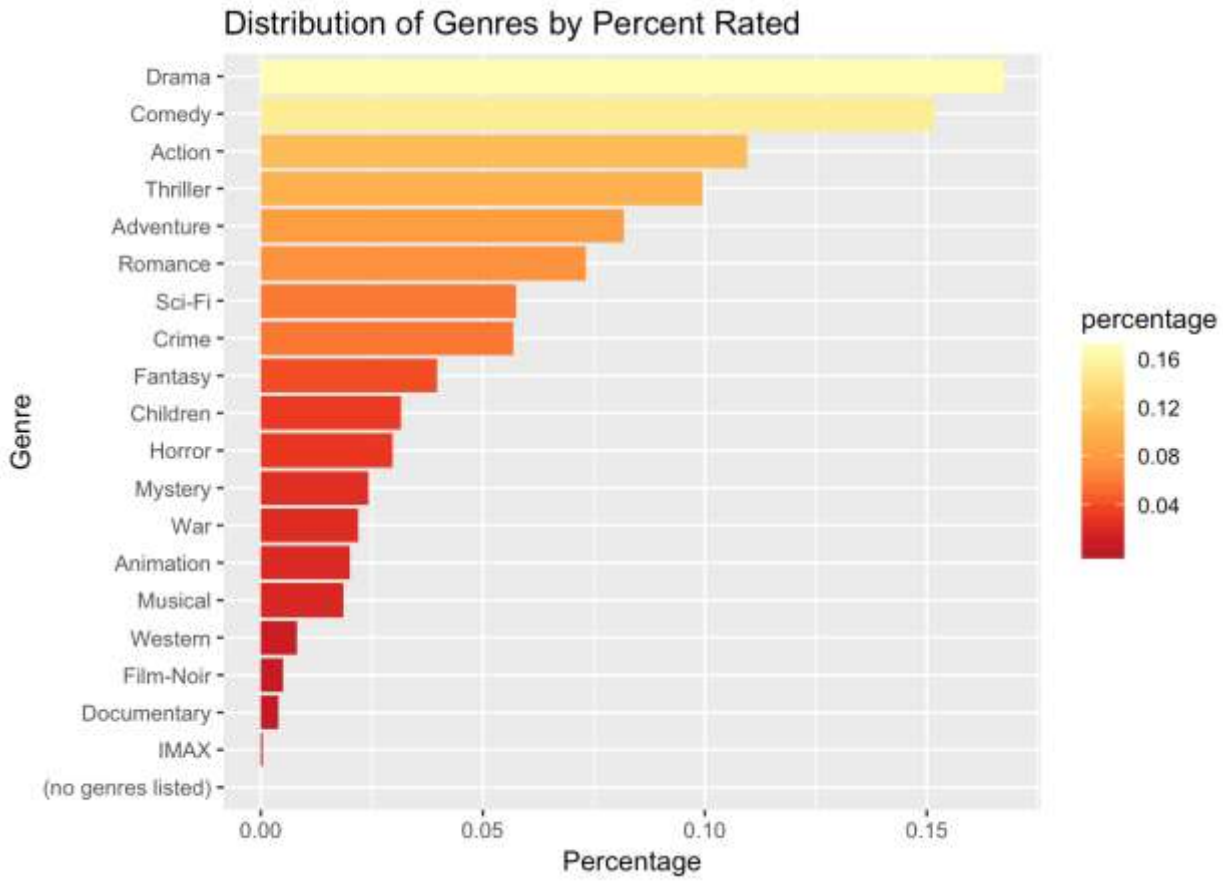
```
genre_list <- number_of_genres$genres
genre_list
##  [1] "(no genres listed)" "Action"             "Adventure"
##  [4] "Animation"          "Children"           "Comedy"
##  [7] "Crime"              "Documentary"        "Drama"
## [10] "Fantasy"            "Film-Noir"          "Horror"
## [13] "IMAX"               "Musical"            "Mystery"
## [16] "Romance"            "Sci-Fi"             "Thriller"
## [19] "War"                "Western"
```

#Explore the distribution of ratings by genre

```
#Distribution of Ratings per Genre
temp <- dat %>%
  group_by(genres) %>%
  summarize(n=n()) %>%
  ungroup() %>%
  mutate(sumN = sum(n), percentage = n/sumN) %>%
  arrange(-percentage)
```

#Bar Graph of Genre's

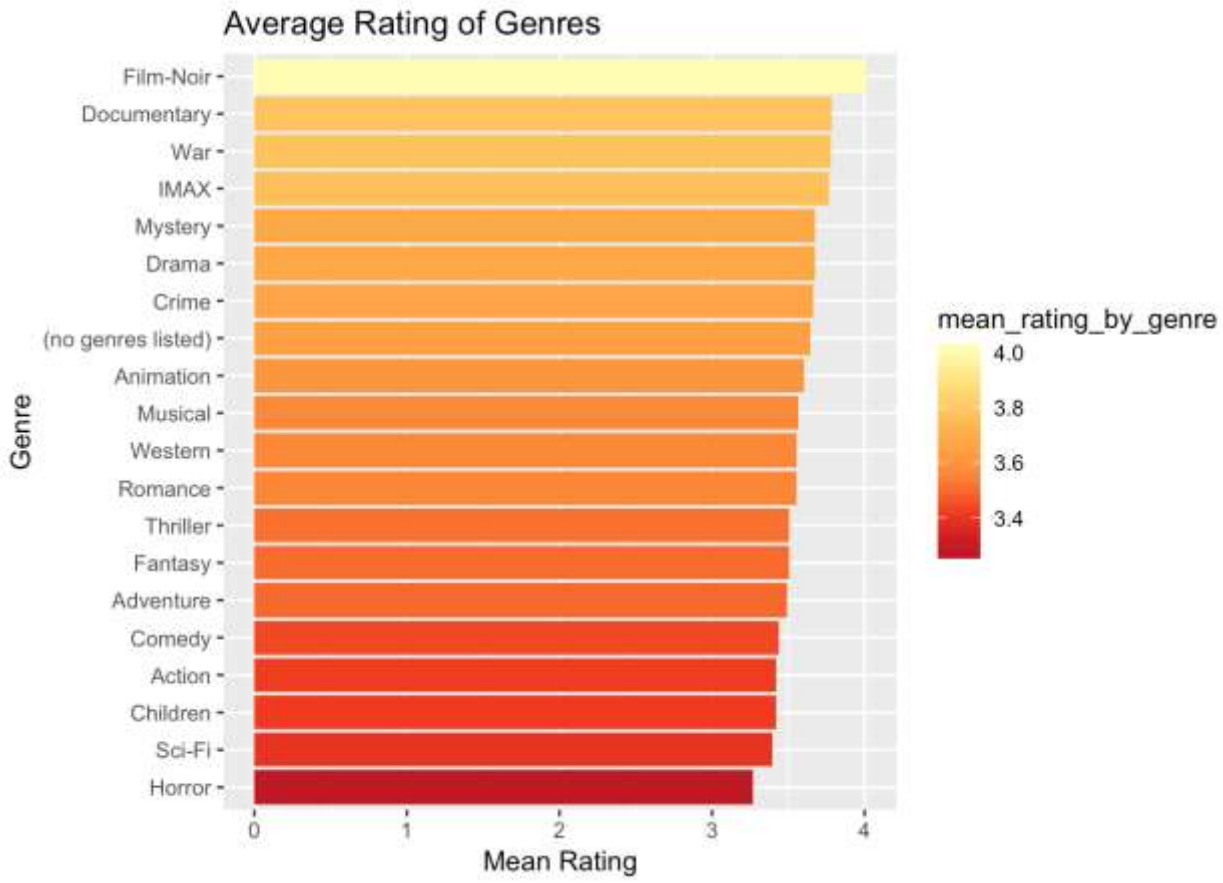
```
temp %>%
  ggplot(aes(reorder(genres, percentage), percentage, fill= percentage)) +
  geom_bar(stat = "identity") + coord_flip() +
  scale_fill_distiller(palette = "YlOrRd") + labs(y = "Percentage", x =
"Genre") +
  ggtitle("Distribution of Genres by Percent Rated")
```



#From the above graph, we can see Drama had the highest percentage of ratings.

#Genre's Mean rating

```
temp <- dat %>%
  group_by(genres) %>%
  summarize(mean_rating_by_genre=mean(rating)) %>%
  arrange(-mean_rating_by_genre)

temp %>%
  ggplot(aes(reorder(genres, mean_rating_by_genre), mean_rating_by_genre,
fill= mean_rating_by_genre)) +
  geom_bar(stat = "identity") + coord_flip() +
  scale_fill_distiller(palette = "YlOrRd") + labs(y = "Mean Rating", x =
"Genre") +
  ggtitle("Average Rating of Genres")
```

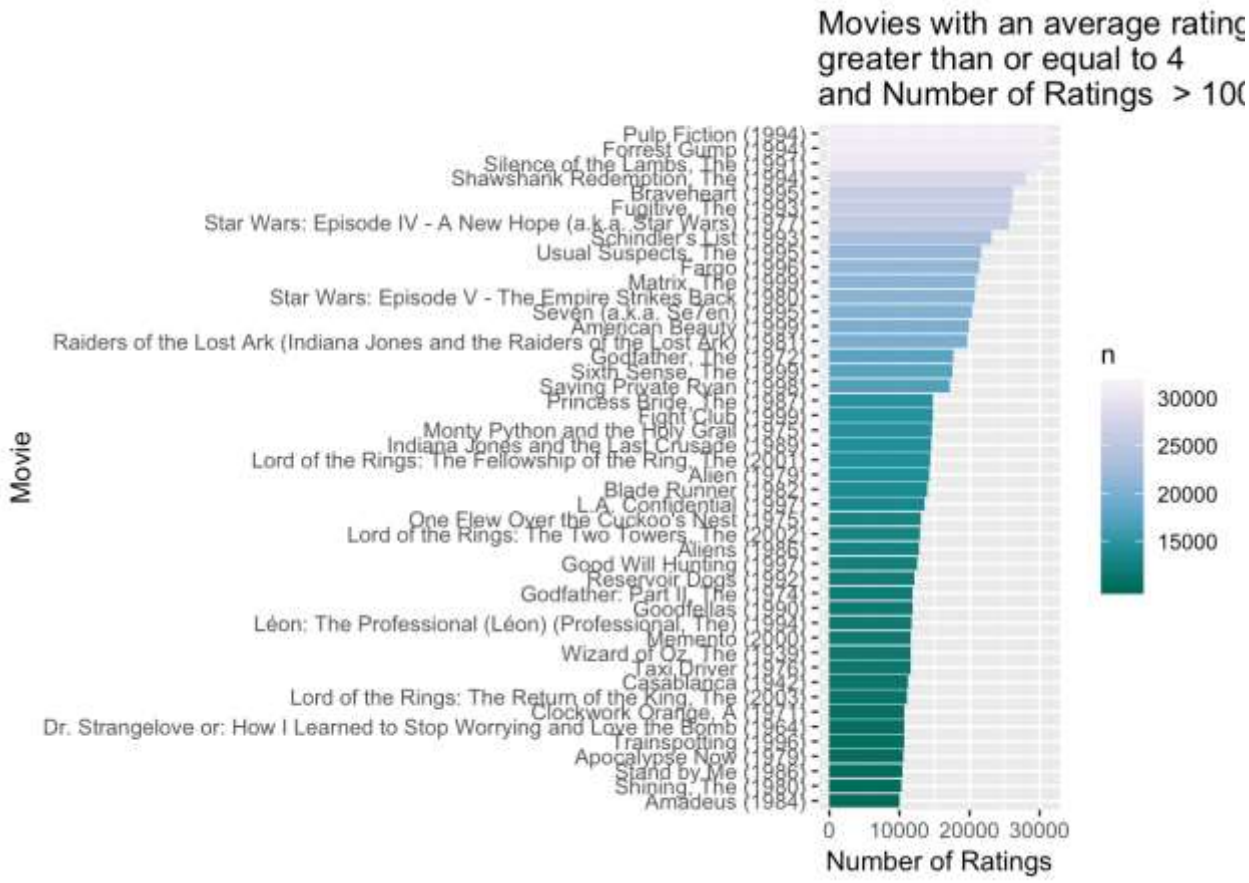


#Film Noir had the highest average rating, while Horror had the lowest average rating.

#Explore movie ratings based on number of ratings and value of the rating

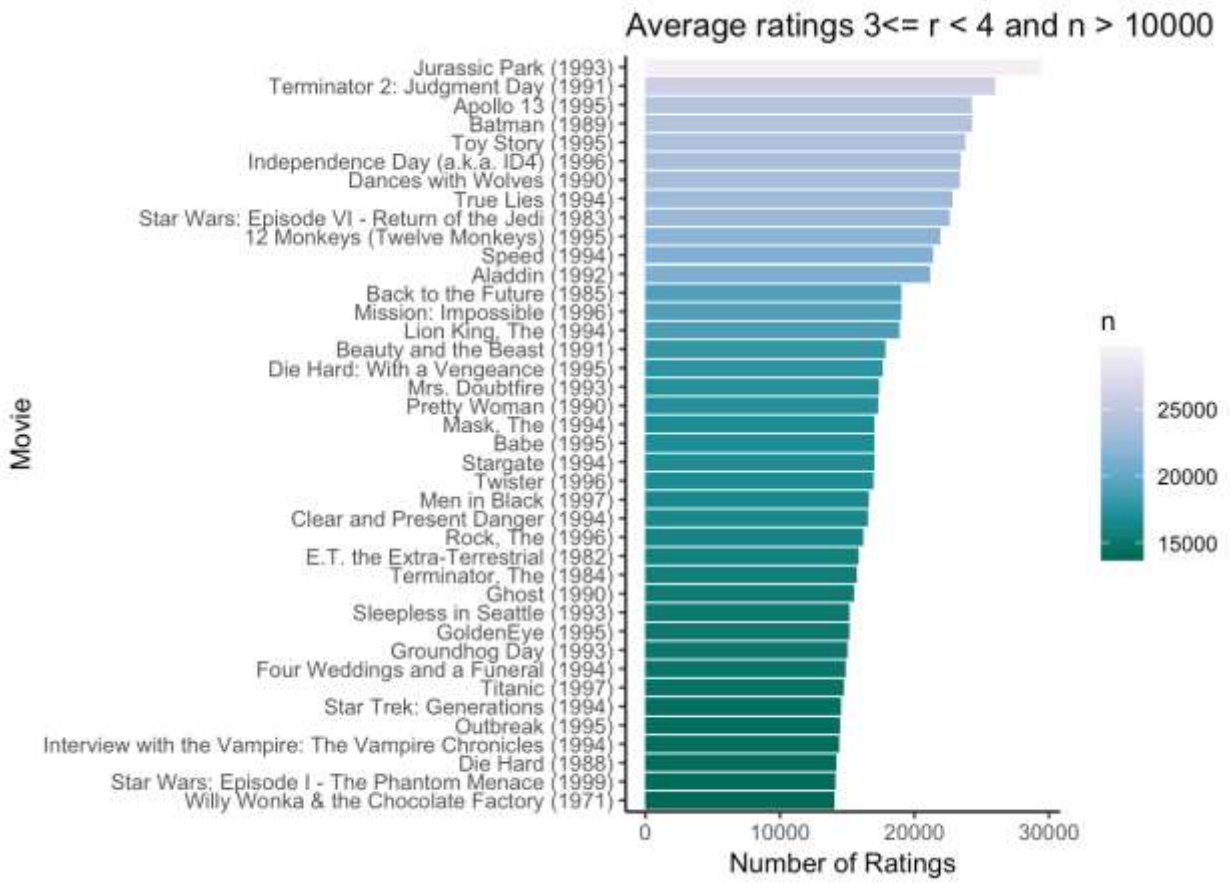
#Graph of movies with more than 10000 ratings and a mean rating greater than 4.

```
avg_rating_greater_than_4 <- edx %>% group_by(title) %>%
     summarize(mean_rating= mean(rating), n = n()) %>% filter(mean_rating
>=4) %>% arrange(desc(n, mean_rating))

avg_rating_greater_than_4 %>% filter(n >=10000) %>%
  ggplot(aes(reorder(title, n), n, fill = n)) +
  geom_bar(stat = "identity") + coord_flip() + scale_fill_distiller(palette =
"PuBuGn") + xlab("Movie") +ylab('Number of Ratings') +
  ggtitle("Movies with an average rating\ngreater than or equal to 4\nand
Number of Ratings  > 10000")
```



Examine Movies with ratings between 3 and 4 and more than 10000 ratings

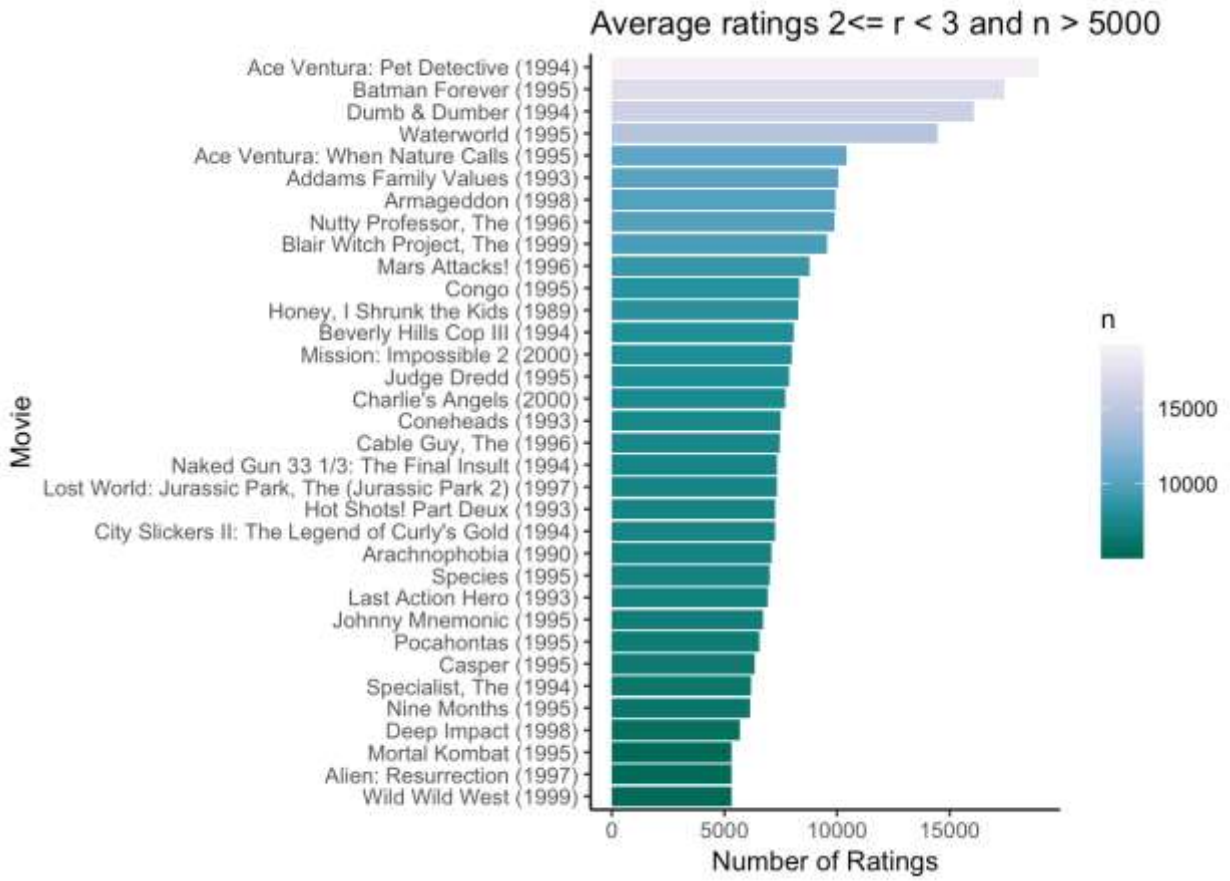
```
avg_between3_4 <- edx %>% group_by(title) %>%
     summarize(mean_rating= mean(rating), n = n()) %>% filter(n > 10000,
(mean_rating >= 3 & mean_rating < 4)) %>% arrange(desc(n, mean_rating))

p <- avg_between3_4 %>% slice(1:40)
p %>%
  ggplot(aes(reorder(title, n), n, fill = n)) +
  geom_bar(stat = "identity") + coord_flip() + scale_fill_distiller(palette =
"PuBuGn") +
  ggtitle("Average ratings 3<= r < 4 and n > 10000") + xlab('Movie') +
ylab('Number of Ratings') +
  theme_classic()
```



#Movies with an average rating between 2 and 3 lets look at number of ratings greater than 5000

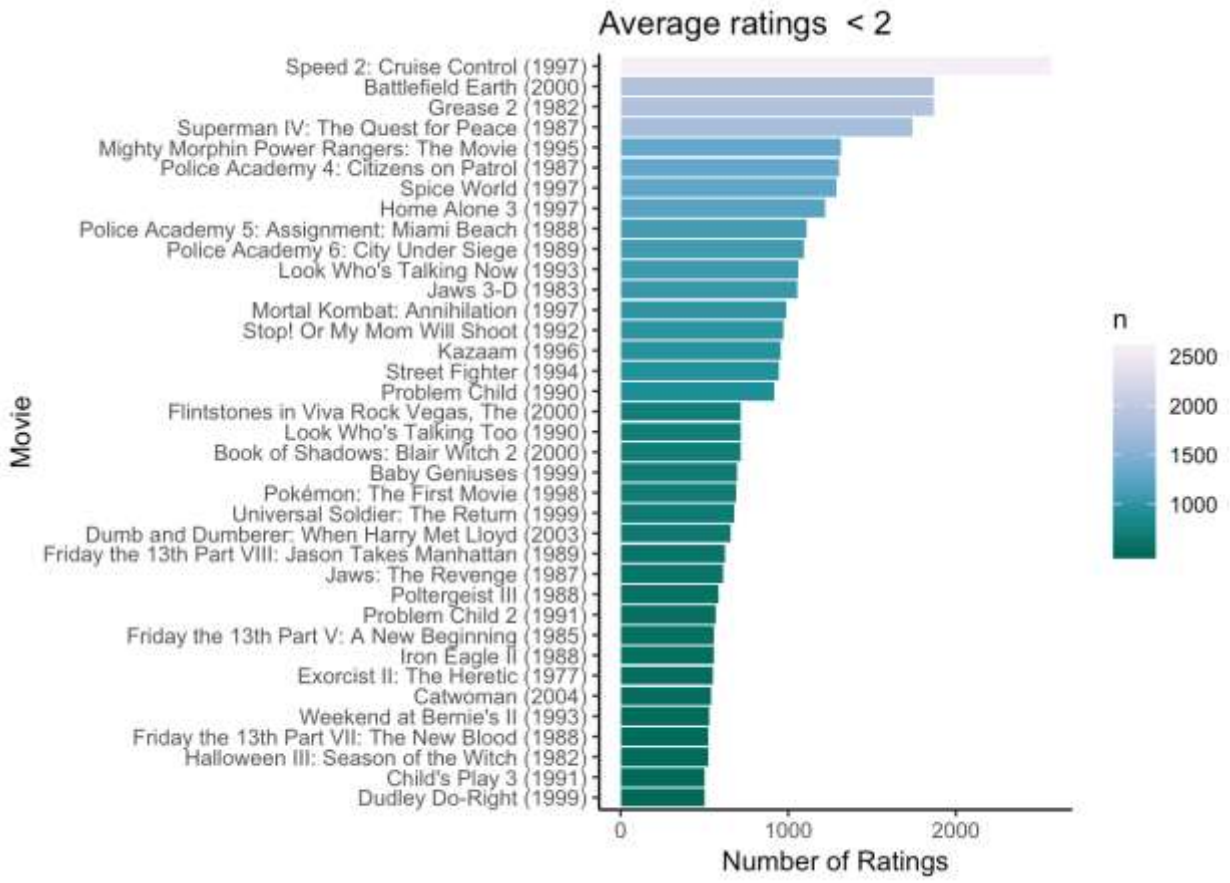
```
avg_between2_3 <- edx %>% group_by(title) %>%
     summarize(mean_rating= mean(rating), n = n()) %>% filter(n > 5000,
(mean_rating >= 2 & mean_rating < 3)) %>% arrange(desc(n, mean_rating))
```

```
avg_between2_3 %>%
  ggplot(aes(reorder(title, n), n, fill = n)) +
  geom_bar(stat = "identity") + coord_flip() + scale_fill_distiller(palette =
"PuBuGn") +
  ggtitle("Average ratings 2<= r < 3 and n > 5000") + xlab('Movie') +
ylab('Number of Ratings') +
  theme_classic()
```



#Less than 10000 ratings and a rating less than 2 and number of ratings greater than 500

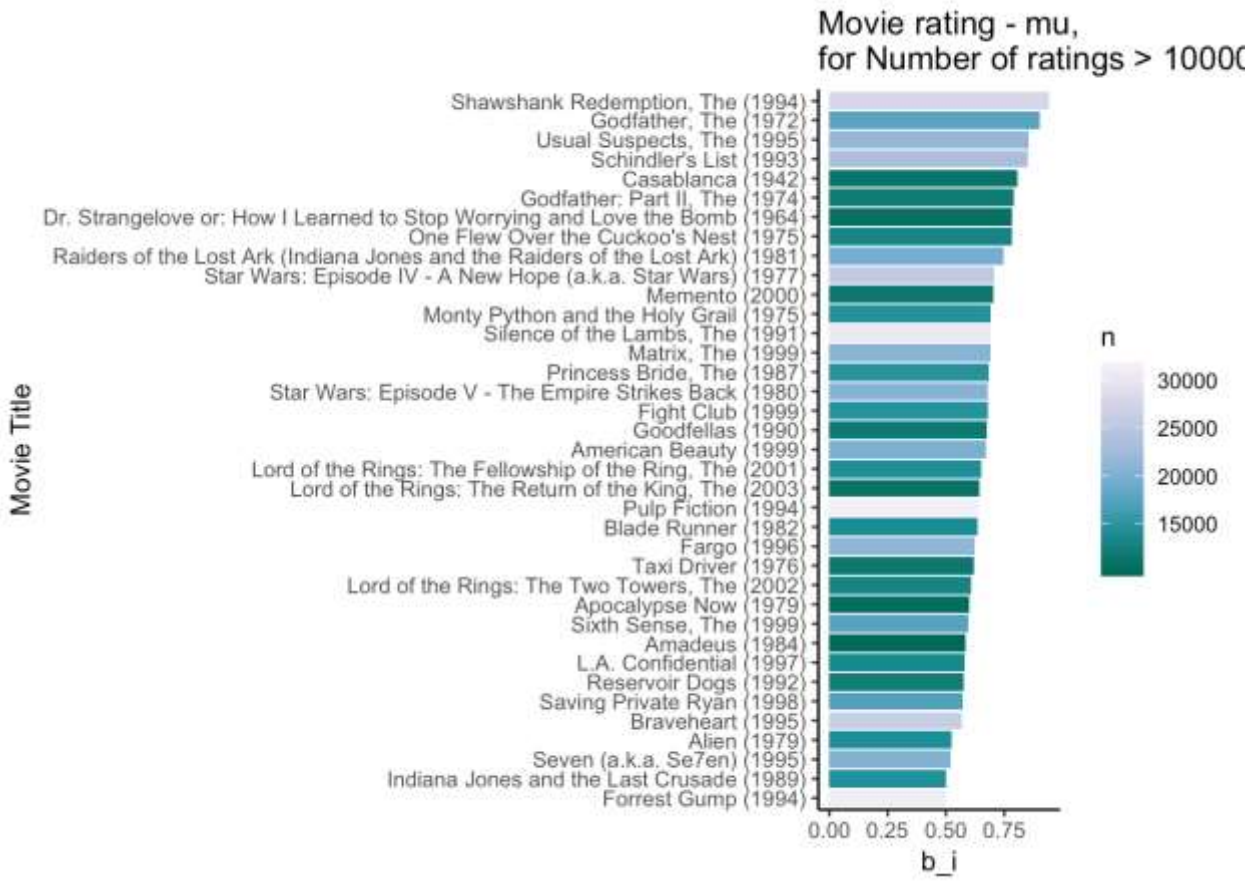
```
avg_rating_less_than_2 <- edx %>% group_by(title) %>%
     summarize(mean_rating= mean(rating), n = n()) %>% filter(n > 500,
mean_rating < 2) %>% arrange(desc(n, mean_rating))

avg_rating_less_than_2 %>%
  ggplot(aes(reorder(title, n), n, fill = n)) +
  geom_bar(stat = "identity") + coord_flip() + scale_fill_distiller(palette =
"PuBuGn") +
  ggtitle("Average ratings  < 2") + xlab('Movie') + ylab('Number of Ratings')
+
  theme_classic()
```



#Compute the least squares for movieId

#Which movies have a large number of ratings and a rating larger than the average mu

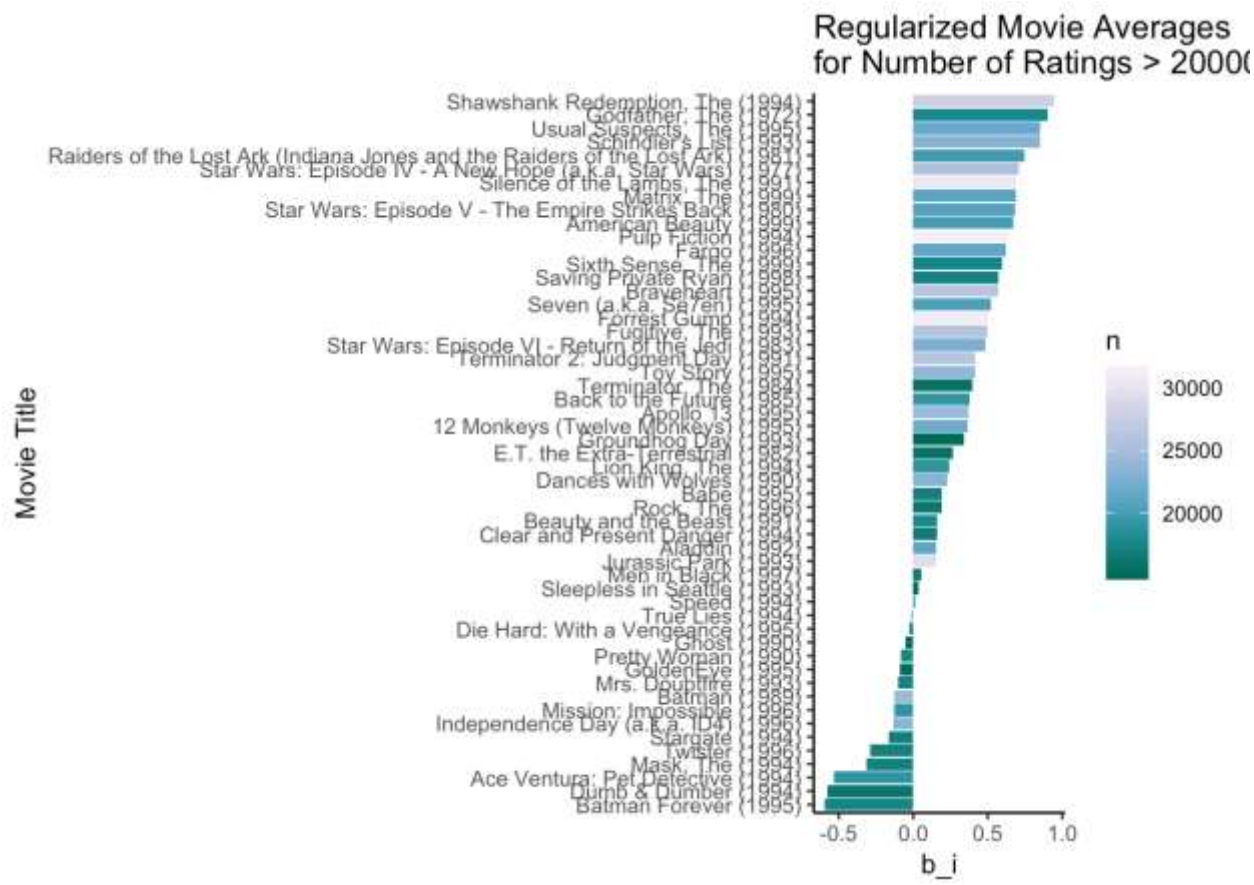
```
mu <- mean(edx$rating)
edx %>% group_by(title) %>%
  summarize(b_i = mean(rating - mu), n = n()) %>% filter(b_i > 0.5, n >
10000) %>%
  ggplot(aes(reorder(title, b_i), b_i, fill = n)) +
  geom_bar(stat = "identity") + coord_flip() + scale_fill_distiller(palette =
"PuBuGn") +
  ggtitle("") + xlab("Movie Title") +
  ggtitle("Movie rating - mu,\nfor Number of ratings > 10000") +
  theme_classic()
```



#Regularized Movie Averages

```
movie_avgs <- edx %>% group_by(movieId) %>% summarize(b_i = mean(rating - mu))
movie_reg_avgs <- edx %>%
  group_by(movieId) %>%
  summarize(b_i = sum(rating - mu)/(n()+1), n_i = n())

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


edx_with_avgs <- edx %>% group_by(title, movieId) %>% summarize(n = n()) %>%
  left_join(movie_reg_avgs, by = "movieId") %>%
  arrange(desc(b_i, n))

edx_with_avgs %>% filter(n > 15000) %>%
  ggplot(aes(reorder(title, b_i), b_i, fill = n)) +
```

```
  geom_bar(stat = "identity") + coord_flip() + scale_fill_distiller(palette =
"PuBuGn") +
  ggtitle("") + xlab("Movie Title") + ggtitle('Regularized Movie
Averages\nfor Number of Ratings > 20000') +
  theme_classic()
```

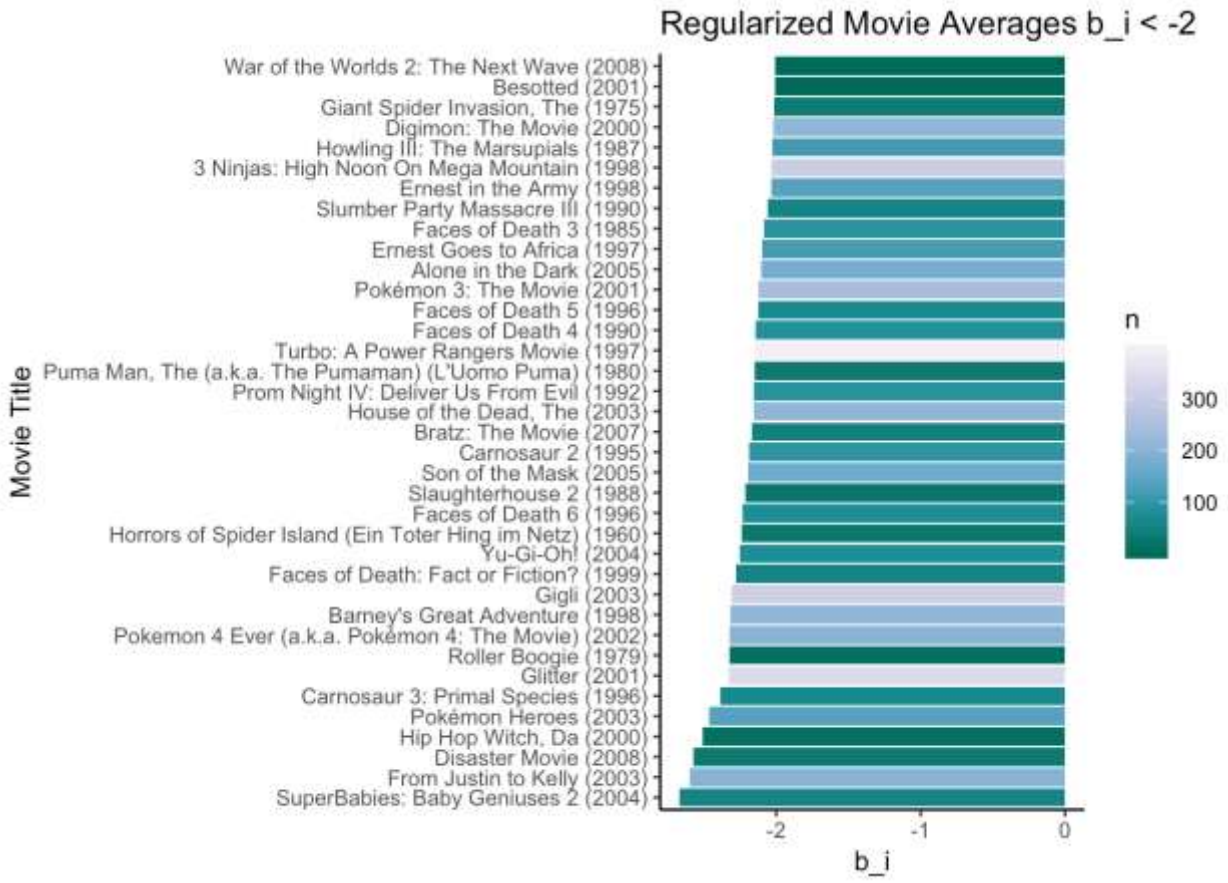


#Regularized Movie Averages for the movies with regularized ratings less than 2

```
head(edx_with_avgs)
```

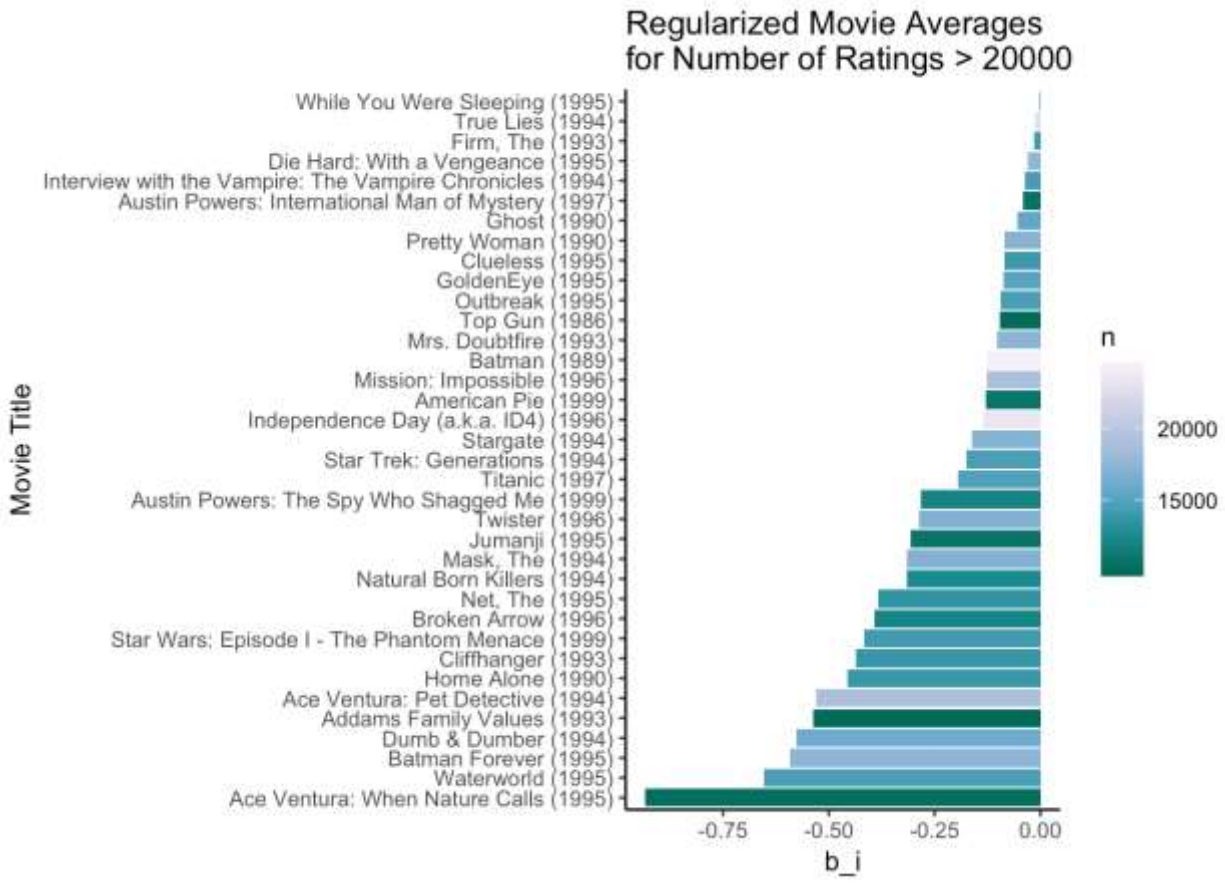
```
## # A tibble: 6 x 5
## # Groups:   title [6]
##   title                                          movieId     n   b_i   n_i
##   <chr>                                            <dbl> <int> <dbl> <int>
## 1 More (1998)                                       4454     7 1.05      7
## 2 Satan's Tango (Sátántangó) (1994)                33264     2 0.992     2
## 3 Human Condition II, The (Ningen no joken II) …   26048     4 0.990     4
## 4 Human Condition III, The (Ningen no joken III…   26073     4 0.990     4
```

```
## 5 Who's Singin' Over There? (a.k.a. Who Sings O…    5194     4 0.990     4
## 6 Shawshank Redemption, The (1994)                   318 28015 0.943 28015
p <- edx_with_avgs %>% arrange(b_i) %>% filter(b_i < -2) %>% arrange((b_i))
p
## # A tibble: 37 x 5
## # Groups:   title [37]
##    title                                          movieId     n   b_i   n_i
##    <chr>                                            <dbl> <int> <dbl> <int>
##  1 SuperBabies: Baby Geniuses 2 (2004)               8859    56 -2.67    56
##  2 From Justin to Kelly (2003)                       6483   199 -2.60   199
##  3 Disaster Movie (2008)                            61348    32 -2.57    32
##  4 Hip Hop Witch, Da (2000)                          7282    14 -2.51    14
##  5 Pokémon Heroes (2003)                             6371   137 -2.47   137
##  6 Carnosaur 3: Primal Species (1996)                3574    68 -2.39    68
##  7 Glitter (2001)                                    4775   339 -2.33   339
##  8 Roller Boogie (1979)                              8856    15 -2.32    15
##  9 Pokemon 4 Ever (a.k.a. Pokémon 4: The Movie)…     5672   202 -2.32   202
## 10 Barney's Great Adventure (1998)                   1826   208 -2.31   208
## # ... with 27 more rows
p %>%
 ggplot(aes(reorder(title, b_i), b_i, fill = n)) +
  geom_bar(stat = "identity") + coord_flip() + scale_fill_distiller(palette =
"PuBuGn") +
  ggtitle("") + xlab("Movie Title") + ggtitle('Regularized Movie Averages b_i
< -2') +
  theme_classic()
```



#Movies with number of ratings larger than 1000 and regularized average less than 0.

```
edx_with_avgs %>% filter(n > 10000, b_i < 0.0) %>%
  ggplot(aes(reorder(title, b_i), b_i, fill = n)) +
  geom_bar(stat = "identity") + coord_flip() + scale_fill_distiller(palette =
"PuBuGn") +
  ggtitle("") + xlab("Movie Title") + ggtitle('Regularized Movie
Averages\nfor Number of Ratings > 20000') +
  theme_classic()
```



#Explore correlation between ratings, users, movieId age of movie and number of ratings

#Is there a correlation

#Number of movie ratings per movie

```
n_movies_ratings <- edx_with_title_dates %>% group_by(movieId) %>%
summarize(n = n())
```

#Average Movie Rating for each movie

```
avg_movie_rat <- edx_with_title_dates %>% group_by(movieId) %>%
summarize(avg_m_r = mean(rating))
```

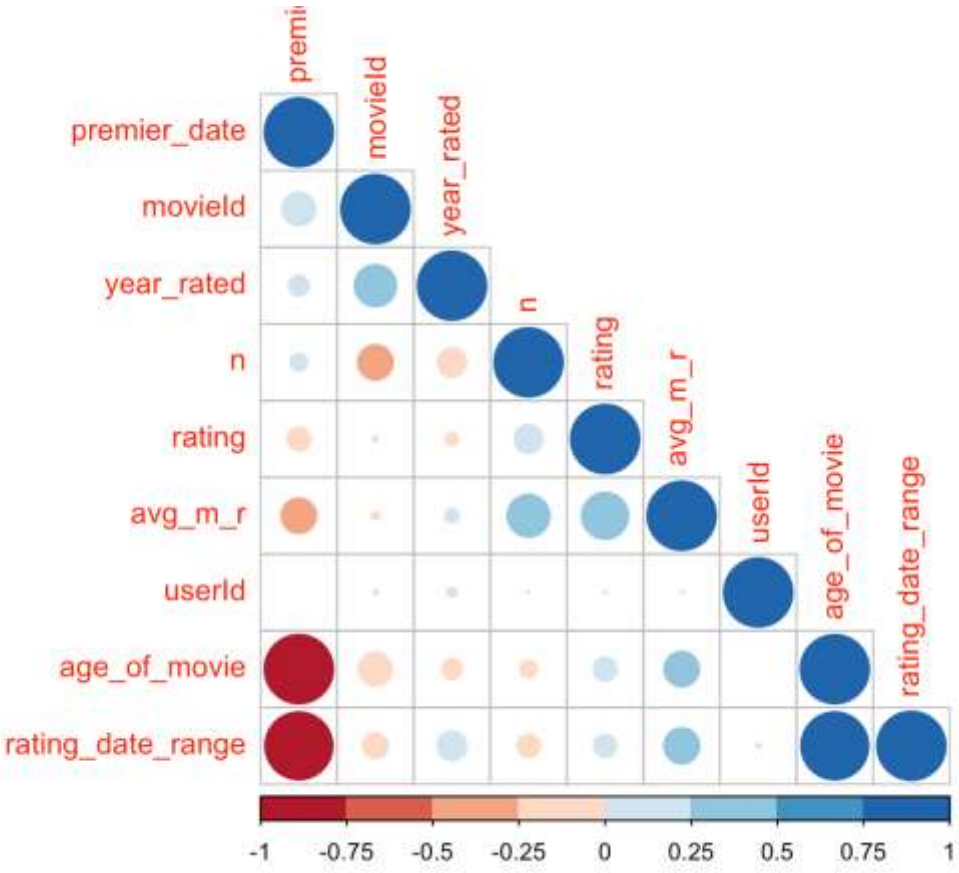
#Create correlation data

```
cor_dat <- edx_with_title_dates %>% select(rating, movieId, userId,
year_rated, age_of_movie, rating_date_range, premier_date) %>%
  left_join(n_movies_ratings, by = "movieId") %>%
```

```
  left_join(avg_movie_rat, by = 'movieId')
head(cor_dat)
##   rating movieId userId year_rated age_of_movie rating_date_range
## 1      5     122      1       1996           26                 4
## 2      5     185      1       1996           23                 1
## 3      5     292      1       1996           23                 1
## 4      5     316      1       1996           24                 2
## 5      5     329      1       1996           24                 2
## 6      5     355      1       1996           24                 2
##   premier_date     n  avg_m_r
## 1         1992  2178 2.858586
## 2         1995 13469 3.129334
## 3         1995 14447 3.418011
## 4         1994 17030 3.349677
## 5         1994 14550 3.337457
## 6         1994  4831 2.487787
```

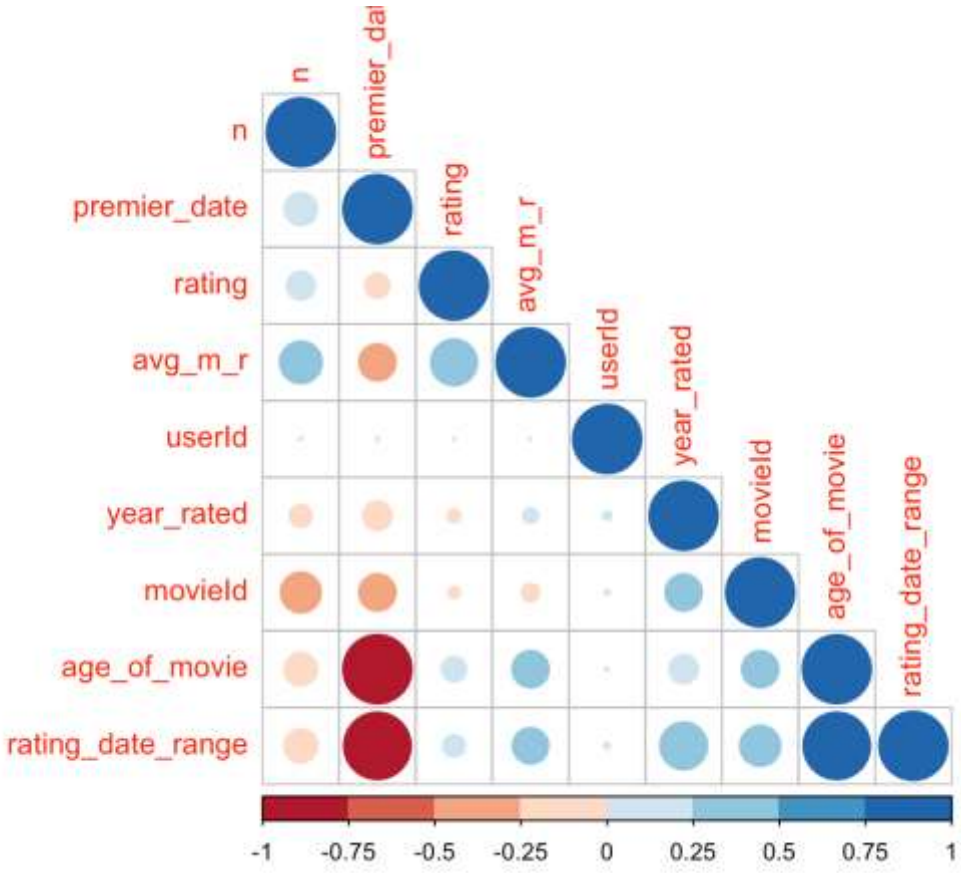
#Graph the correlation

```
temp <- cor_dat %>% select(one_of("rating", "movieId", "userId",
"year_rated", "age_of_movie",
                                                    "rating_date_range",
"premier_date", "n", "avg_m_r")) %>% as.matrix()
M <- cor(temp, use = "pairwise.complete.obs")

corrplot(M, order = "hclust", addrect = 2, type = "lower", col = brewer.pal(n
= 8, name = "RdBu"))
```



#What is the effect of the age of the movie

```
corr_by_age_of_movie <- cor_dat %>% filter((age_of_movie >20) & (age_of_movie
< 70))
temp <- corr_by_age_of_movie %>% select(one_of("rating", "movieId", "userId",
"year_rated", "age_of_movie",
                                    "rating_date_range", "n", "premier_date",
"avg_m_r")) %>% as.matrix()
M <- cor(temp, use = "pairwise.complete.obs")

corrplot(M, order = "hclust", addrect = 2, type = "lower", col = brewer.pal(n
= 8, name = "RdBu"))
```

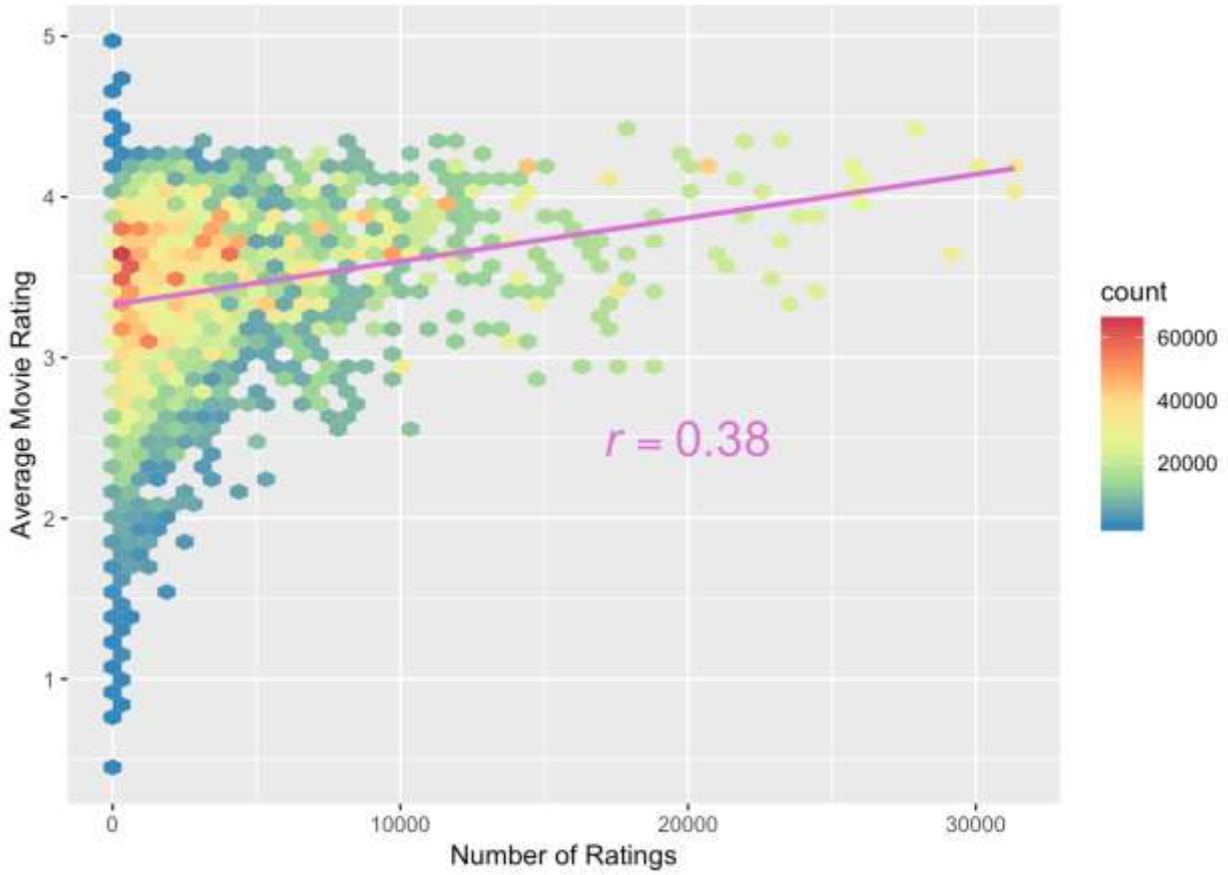


#Is there a relationship between number of ratings and the average rating

```
get_cor <- function(df){
  m <- cor(df$x, df$y, use="pairwise.complete.obs");
  eq <- substitute(italic(r) == cor, list(cor = format(m, digits = 2)))
  as.character(as.expression(eq));
}
```

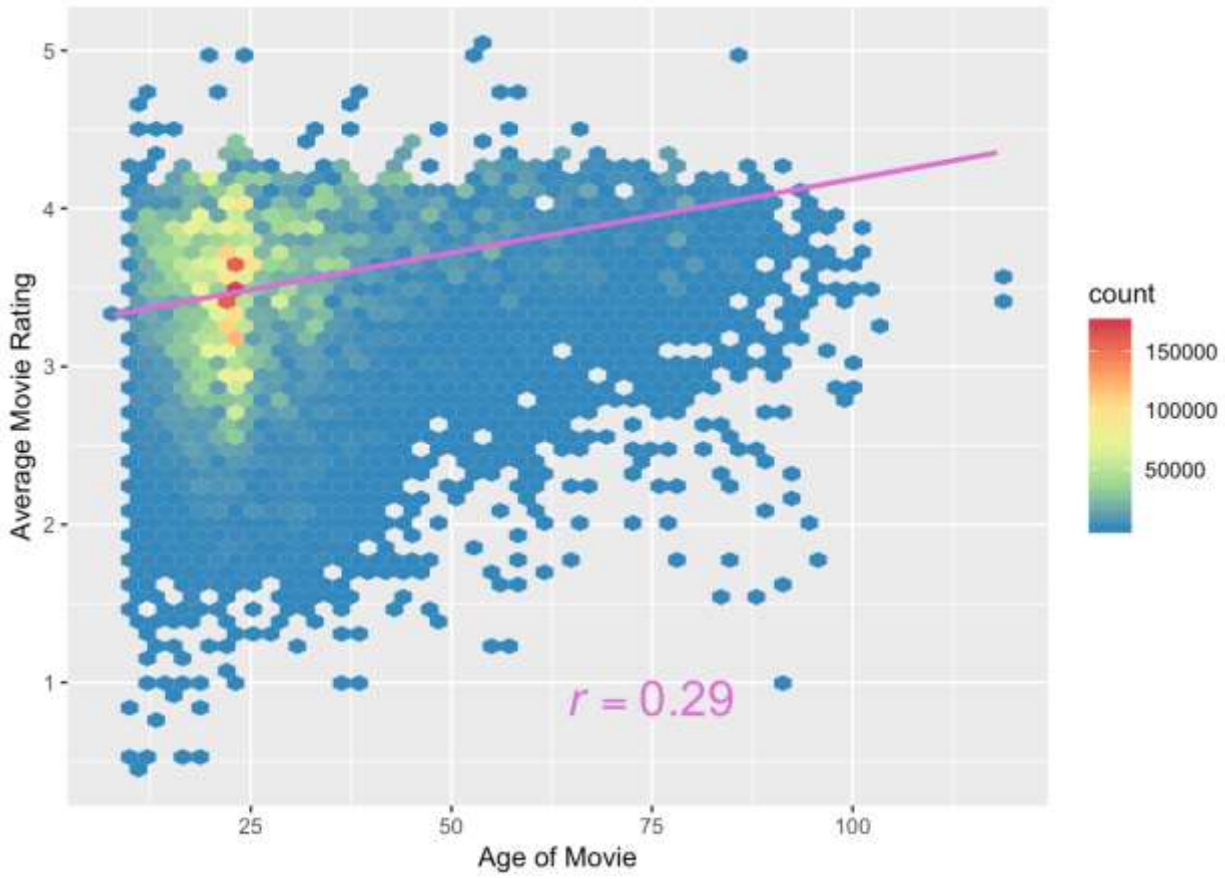
#Number of ratings vs avg movie ratings

```
cor_dat %>%
  ggplot(aes(n, avg_m_r)) + stat_bin_hex(bins = 50) +
scale_fill_distiller(palette = "Spectral") +
  stat_smooth(method = "lm", color = "orchid", size = 1) +
  annotate("text", x = 20000, y = 2.5, label = get_cor(data.frame(x =
cor_dat$n, y = cor_dat$avg_m_r)),
           parse = TRUE, color = "orchid", size = 7) + ylab("Average Movie
Rating") + xlab("Number of Ratings")
```



#Is there an Age Effect on Movie Ratings?

```
cor_dat %>%
  ggplot(aes(age_of_movie, avg_m_r)) + stat_bin_hex(bins = 50) +
scale_fill_distiller(palette = "Spectral") +
  stat_smooth(method = "lm", color = "orchid", size = 1) +
  annotate("text", x = 75, y = 0.9, label = get_cor(data.frame(x =
corr_by_age_of_movie$age_of_movie, y = corr_by_age_of_movie$avg_m_r)),
           parse = TRUE, color = "orchid", size = 7) + ylab("Average Movie
Rating") + xlab('Age of Movie')
```



#Calculate the RMSE

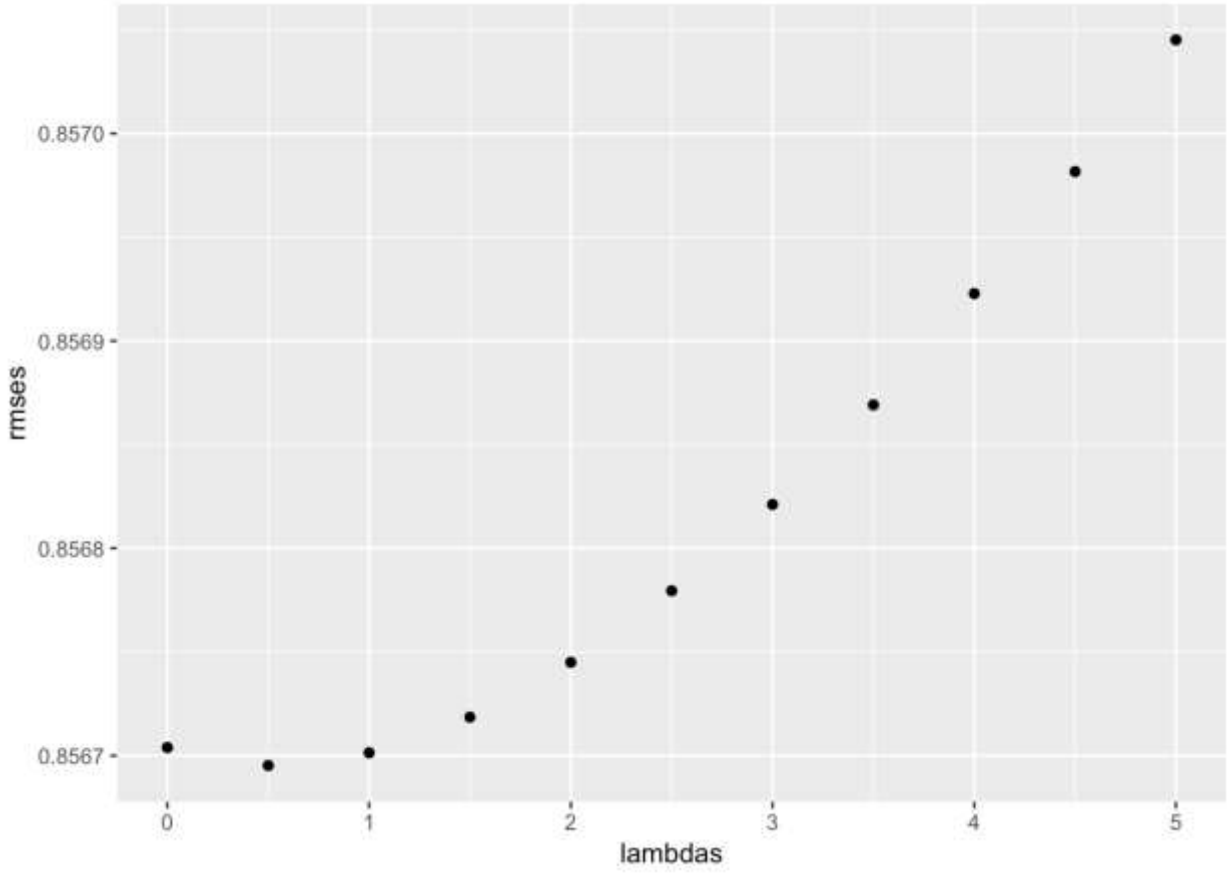
#RMSE function

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

#Choose the tuning value

```
lambdas <- seq(0,5,.5)
rmses <- sapply(lambdas, function(l){
  mu <- mean(edx_with_title_dates$rating)

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

  b_u <- edx_with_title_dates %>%
```

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

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

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

qplot(lambdas, rmses)
```




```
lambdas[which.min(rmses)]
## [1] 0.5
```

Using the model on the Validation data

```
mu <- mean(validation$rating)
l <- 0.15
b_i <- validation %>%
    group_by(movieId) %>%
    summarize(b_i = sum(rating - mu)/(n() + l))

b_u <- validation %>%
    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) %>% .$pred

RMSE(predicted_ratings, validation$rating)
## [1] 0.8252108
```

I originally calculated a b_a for the age of a movie but found it didn't lower my RMSE so took it out and didn't include it in this script.

#I used movieId and userId to calculate the RMSE and was able to achieve an RMSE = 0.826

#The code below utilizes the package "Metrics", which resulted in the same RMSE. I included this as a check for my calculations.

```
library(Metrics)
##
## Attaching package: 'Metrics'
## The following objects are masked from 'package:modelr':
##
```

```
##     mae, mape, mse, rmse
## The following objects are masked from 'package:caret':
##
##     precision, recall
rmse(validation$rating, predicted_ratings)
## [1] 0.8252108
```

#This was an amazing assignment; I'm sure I did way more than I needed to as this script is quite long. However, I wanted to use this data as a learning tool to explore as much as possible from what I've learned in this R adventure.