Building a Non Personalized Recommendation Engine

Author: Sondos Atwi

==========

Initialization

Import libraries

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(reshape)
```

```
##
## Attaching package: 'reshape'
```

```
## The following object is masked from 'package:dplyr':
##
##     rename
```

Load the critics dataset.

```
critics = read.csv("../Data/critics.csv", check.names=FALSE)
#Transform data to long format to use dplyr
mcritics = melt(critics, id=c("User"), variable_name= "movie", na.rm = TRUE)
```

1) Top 5 movies, by rating mean

Calculate mean rating for each movie, ordered with the highest rating listed first, and plot the top 5.

Function to get the top movies in a dataframe:
df: dataframe to be provided (should be the original movies df)
n: top n movies
m: 0 by default. Should be used in case a minimum number of ratings is required

```
topMovies = function(df = mcritics, n = 5, m = 0){
  top = df %>%
    group_by(movie) %>%
    filter(n() >= m) %>%
    summarise("meanRating" = mean(value)) %>%
    arrange(desc(meanRating)) %>%
    slice(1:n)
```

```
  top
}
```

The below function will print the output in a special format

```
customPrint = function(data, rd=0, col1, col2 = "movie"){
  df=data.frame(paste(format(round(data[[col1]],rd), nsmall = rd), paste("\'", data[[col2]], "\'", sep =

names(df) = NULL
print(df, digits = NULL, quote = FALSE, right = FALSE, row.names = FALSE)
}
```

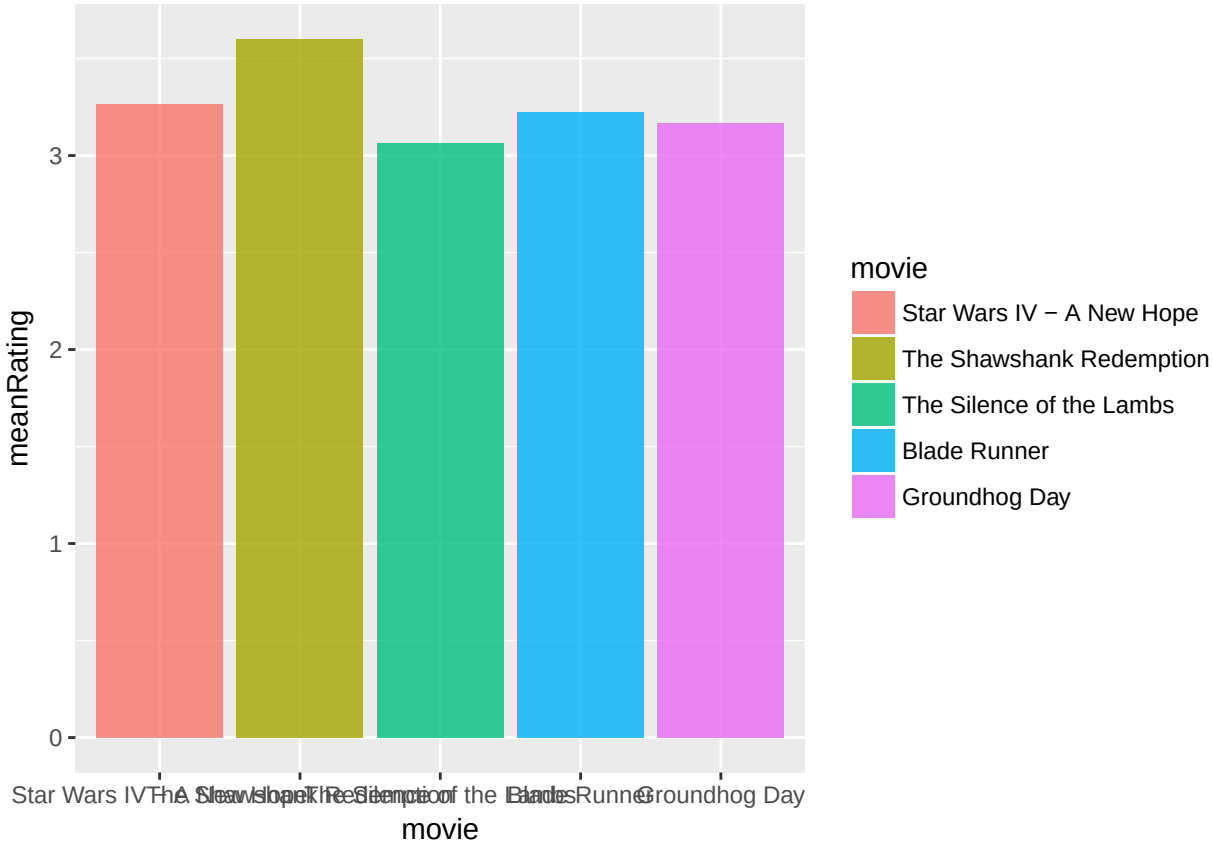
```
#find top 5 movies in the mcritics dataframe
top5 = topMovies()
```

Print the top 5 movies and Plot

```
customPrint(top5, 6, "meanRating")
```

```
##
##  3.600000, 'The Shawshank Redemption'
##  3.266667, 'Star Wars IV - A New Hope'
##  3.222222, 'Blade Runner'
##  3.166667, 'Groundhog Day'
##  3.062500, 'The Silence of the Lambs'
```

```
ggplot(data=top5, aes(movie, meanRating, fill = movie)) +
  geom_bar(stat="identity", alpha = 0.8)
```



2) Top 5 movies, by rating distribution

Calculate the percentage of ratings for each movie that are 4 stars or higher. Order with the highest percentage first, and plot the top 5 movies with more high ratings distribution.

Function to get the rating distribution by dividing the number of records having rating (value) >= 4 by the total number of ratings for each movie.
df: dataframe to be provided (should be the original movies df)
n: top n records
m: 0 by default. Should be used in case a minimum number of ratings is required

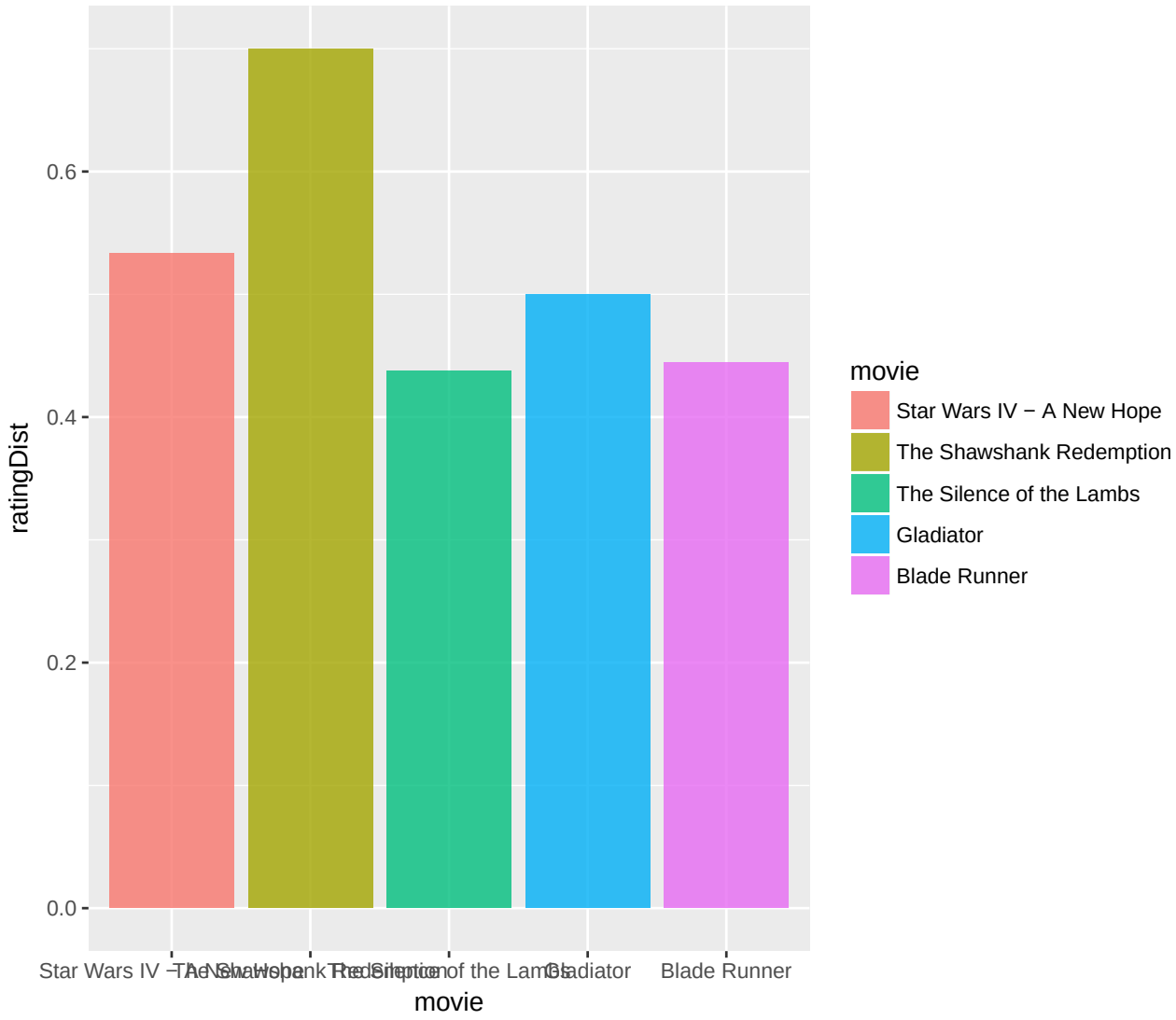
```
ratingDist = function(df = mcritics, n = 5, m = 0){
  rd = df %>%
    group_by(movie) %>%
    filter(n() >= m) %>%
    summarise(ratingDist = sum(value >=4)/n()) %>%
    select(ratingDist, movie) %>%
    arrange(desc(ratingDist)) %>%
    slice(1:n)
  rd
}
```

```
ratings =  ratingDist()

customPrint(ratings, 6, "ratingDist")
```

```
##
##  0.700000, 'The Shawshank Redemption'
##  0.533333, 'Star Wars IV - A New Hope'
##  0.500000, 'Gladiator'
##  0.444444, 'Blade Runner'
##  0.437500, 'The Silence of the Lambs'
```

```
ggplot(data=ratings, aes(movie, ratingDist, fill = movie)) +
  geom_bar(stat="identity", alpha = 0.8)
```



3) Top 5 movies, by quantity of ratings

Count the number of ratings for each movie, order the one with most number of ratings first, submit the top 5.

Function to count the number of ratings for each movie.
df: dataframe to be provided (should be the original movies df)
n: top n records

```
ratingsNb = function(df = mcritics, n = 5){
  rNb = df %>%
    group_by(movie) %>%
    summarise(total = n()) %>%
    arrange(desc(total)) %>%
    slice(1:n)

  rNb
}
```

```r
#find the total number of ratings for each
ratingsdf = ratingsNb()
customPrint(ratingsdf, col1 = "total")
```

```
##
##  17, 'Toy Story'
##  16, 'The Silence of the Lambs'
##  15, 'Star Wars IV - A New Hope'
##  14, 'Star Wars VI - Return of the Jedi'
##  13, 'Independence Day'
```

4) People who watched Star Wars IV also watched ...

Calculate movies that most often occur with other movie. For each movie, calculate the percentage of the other movie raters who also rated that movie. Order with the highest percentage first, and submit the top 5.

Funtion to create a dataframe with people who rated a movie.
n: default is 0. This is the rating value to filter on.

```r
movieUsers = function(df = mcritics, mv, n = 0){
  mvUsers =  df %>%
  filter(movie == mv) %>%
  filter(value >= n)

  mvUsers
}
```

Function to create dataframe with the other movies liked by the users who liked a movie *mv*.
df1: original movies dataframe
df2: dataframe with users who liked movie *mv*

```r
uAlsoWatched = function(df1 = mcritics, df2, mv)
{
  liked = df1 %>%
    filter(User %in% df2$User) %>%
    filter(movie != mv) %>%
    group_by(movie)

  liked
}
```

```r
#dataframe with users who watched "Star Wars IV - A New Hope"
SWUsers = movieUsers(mv = "Star Wars IV - A New Hope")
#total number of users who wathed star wars
totalStars = nrow(SWUsers)

#The other movies watched by those who watched Star Wars
SWalsoWatched = uAlsoWatched(df2 = SWUsers, mv = "Star Wars IV - A New Hope")

#For each movie other than Star Wars, divide the number of users who watched that movie by the total nu
swMostLiked = SWalsoWatched %>%
  summarise(alsoWatched = sum(User %in% SWUsers$User)/totalStars) %>%
  arrange(desc(alsoWatched)) %>%
  slice(1:5)
```

```
customPrint(swMostLiked, 6, "alsoWatched")
```

```
##
##  0.933333, 'Toy Story'
##  0.866667, 'Star Wars VI - Return of the Jedi'
##  0.800000, 'The Silence of the Lambs'
##  0.733333, 'Independence Day'
##  0.666667, 'Total Recall'
```

5) People who liked Babe also liked ...

Calculate the movies better rated of people who liked a movie. Select the people who liked the movie "Babe" (4 or 5 stars) and provide the top 5 movies they liked most.

```
#Users who liked the movie "Babe", with rating >= 4
babesUsers = movieUsers(mcritics, "Babe", 4)

#Other movies liked by the users who liked the movie "Babe"
alsoLiked = uAlsoWatched(df2 = babesUsers, mv = "Babe")

#Get the top 5 by calculating the mean rating value of the other movies
alsoLiked = alsoLiked %>%
  summarise(rate = mean(value)) %>%
  arrange(desc(rate), movie) %>%
  slice(1:5)

customPrint(alsoLiked, 3, "rate")
```

```
##
##  5.000, 'Pulp Fiction'
##  5.000, 'Groundhog Day'
##  4.500, 'The Shawshank Redemption'
##  4.333, 'Toy Story'
##  4.000, 'Blade Runner'
```

6) movieLens

Explore a real data set and provide non-personalized ratings. You can use the movieLens database. You can find movieLens' dataset here: http://files.grouplens.org/datasets/movielens/ml-10m-README.html

```
movies = read.csv("../Data/ml-latest-small/movies.csv")
ratings = read.csv("../Data/ml-latest-small/ratings.csv")

#Select only movieId and title columns
movies = movies %>%
  select(1:2)

#select only userId, movieId and rating
ratings = ratings %>%
  select(1:3)
```

```
#Merge dataframes and rename columns
data = merge(movies, ratings, by = "movieId")
data = rename(data, c("title" = "movie", "rating" = "value", "userId" = "User"))
```

Explore dataset

```
#Create a dataframe to calculate the total number of ratings for each movie
movieRatingCount = data %>%
  group_by(movie) %>%
  summarise(total = n()) %>%
  arrange(desc(total))

#get the summary of the total column to see the movie rating count distribution
summary(movieRatingCount$total)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    1.00    3.00   11.03    9.00  341.00
```

Get top 5 movies:

From the summary above, we can see that some movies have only 1 rating. These could be misleading when analyzing the most popular movies.
Therefore, we will only take into consideration the ones that have been rated by at least 10 users.

```
#Get the top 5 movies that have been rated by at least 10 users
top5movies = topMovies(data, m = 10)
customPrint(top5movies, 6, "meanRating")
```

```
##
##  4.636364, 'Best Years of Our Lives, The (1946)'
##  4.541667, 'Inherit the Wind (1960)'
##  4.487500, 'Godfather, The (1972)'
##  4.487138, 'Shawshank Redemption, The (1994)'
##  4.458333, 'Tom Jones (1963)'
```

Top 5 movies by rating distribution

Get the 10 movies that have the highest percentage of ratings >=4.
Here we will also filter on those movies that have at least 10 user ratings.

```
#Get the top 10 movies by rating distribution that have been rated by at least 10 users
movieRatings = ratingDist(data, 10, 10)

customPrint(movieRatings, 6, "ratingDist")
```

```
##
##  1.000000, 'Inherit the Wind (1960)'
##  1.000000, 'Rabbit-Proof Fence (2002)'
##  1.000000, 'Shadow of a Doubt (1943)'
##  1.000000, 'Smoke (1995)'
##  1.000000, 'Tom Jones (1963)'
##  0.944444, 'Little Big Man (1970)'
##  0.941176, 'Mister Roberts (1955)'
##  0.937500, 'You Can Count on Me (2000)'
##  0.928571, 'Roger & Me (1989)'
```

```
##  0.923077, 'Exotica (1994)'
```

Get the movies with the highest number of ratings

```
movieRatingNb = ratingsNb(data, 10)

customPrint(movieRatingNb, col1="total", col2="movie")
```

```
##
##  341, 'Forrest Gump (1994)'
##  324, 'Pulp Fiction (1994)'
##  311, 'Shawshank Redemption, The (1994)'
##  304, 'Silence of the Lambs, The (1991)'
##  291, 'Star Wars: Episode IV - A New Hope (1977)'
##  274, 'Jurassic Park (1993)'
##  259, 'Matrix, The (1999)'
##  247, 'Toy Story (1995)'
##  244, 'Schindler's List (1993)'
##  237, 'Terminator 2: Judgment Day (1991)'
```

Those who watched Toy Story also watched..

```
#Users who watched the Toy Story movie
toyStoryUsers = movieUsers(data, "Toy Story (1995)")
totalToyStory = nrow(toyStoryUsers)

#Find what the users who watched toy Story also watched
tsAlsoWatched = uAlsoWatched(data, toyStoryUsers,  "Toy Story (1995)")

#For each movie other than Toy Story, divide the number of users who watched that movie by the total nu
tsAlsoWatched =  tsAlsoWatched %>%
  summarise(alsoWatched = sum(User %in% toyStoryUsers$User)/totalToyStory) %>%
  arrange(desc(alsoWatched)) %>%
  slice(1:5)

customPrint(tsAlsoWatched, 6, "alsoWatched", "movie")
```

```
##
##  0.712551, 'Forrest Gump (1994)'
##  0.663968, 'Star Wars: Episode IV - A New Hope (1977)'
##  0.619433, 'Pulp Fiction (1994)'
##  0.595142, 'Jurassic Park (1993)'
##  0.591093, 'Shawshank Redemption, The (1994)'
```

Those who liked 'Little Big Man (1970)' also liked..

```
#create the lbm dataframe with the users who rated 'Little Big Man (1970)" with a 4 or above
lbmUsers = movieUsers(data, "Little Big Man (1970)")

#Find what the users who watched Little Big Man (1970) also watched
lbmAlsoWatched = uAlsoWatched(data, lbmUsers, "Little Big Man (1970)")

#Get the top 5 liked movies by Little Black Man users
mostLikedlbm = lbmAlsoWatched %>%
```

```
                  summarise(rate = mean(value)) %>%
                  arrange(desc(rate), desc(movie)) %>%
                  slice(1:5)

customPrint(mostLikedlbm, 3, "rate", "movie")
```

```
##
##  5.000, 'Zorba the Greek (Alexis Zorbas) (1964)'
##  5.000, 'Z (1969)'
##  5.000, 'Yojimbo (1961)'
##  5.000, 'Without a Clue (1988)'
##  5.000, 'Wings (1927)'
```