Building a Collaborative Filtering 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
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

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

Load the critics dataset

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

1) Pearson correlation coefficient

Calculate the Pearson correlation coefficient between Victoria and Nuria

The below compare function will get all the common movies between two users and will either: return correlation (if method == 0) or return the plot (method != 0)

```
correlations = function(u1, u2, method = 0){
  #Filter on the two users and on movies that they both rated
  common = mcritics %>%
```

```
          filter(User %in% c(u1, u2)) %>%
          filter(!is.na(value)) %>%
          group_by(Movie) %>%
          filter(n()==2)

  #create a new dataframe with every movie and the rating of each user
  values = data.frame(movie = unique(common$Movie),
                      r1 = common$value[common$User==u1],
                      r2 = common$value[common$User==u2])

  #calculate correlation
  if(method == 0){
    cor(values$r1, values$r2)}

  #return plot
  else{
   ggplot(values, aes(x= r1, y = r2, color = movie)) +
   geom_point() +
   geom_abline(slope = 1, linetype = 2) +
   xlim(1,5) +
   ylim(1,5)
  }
}
```

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)
}
```

```
correlations("Victoria", "Nuria")
```

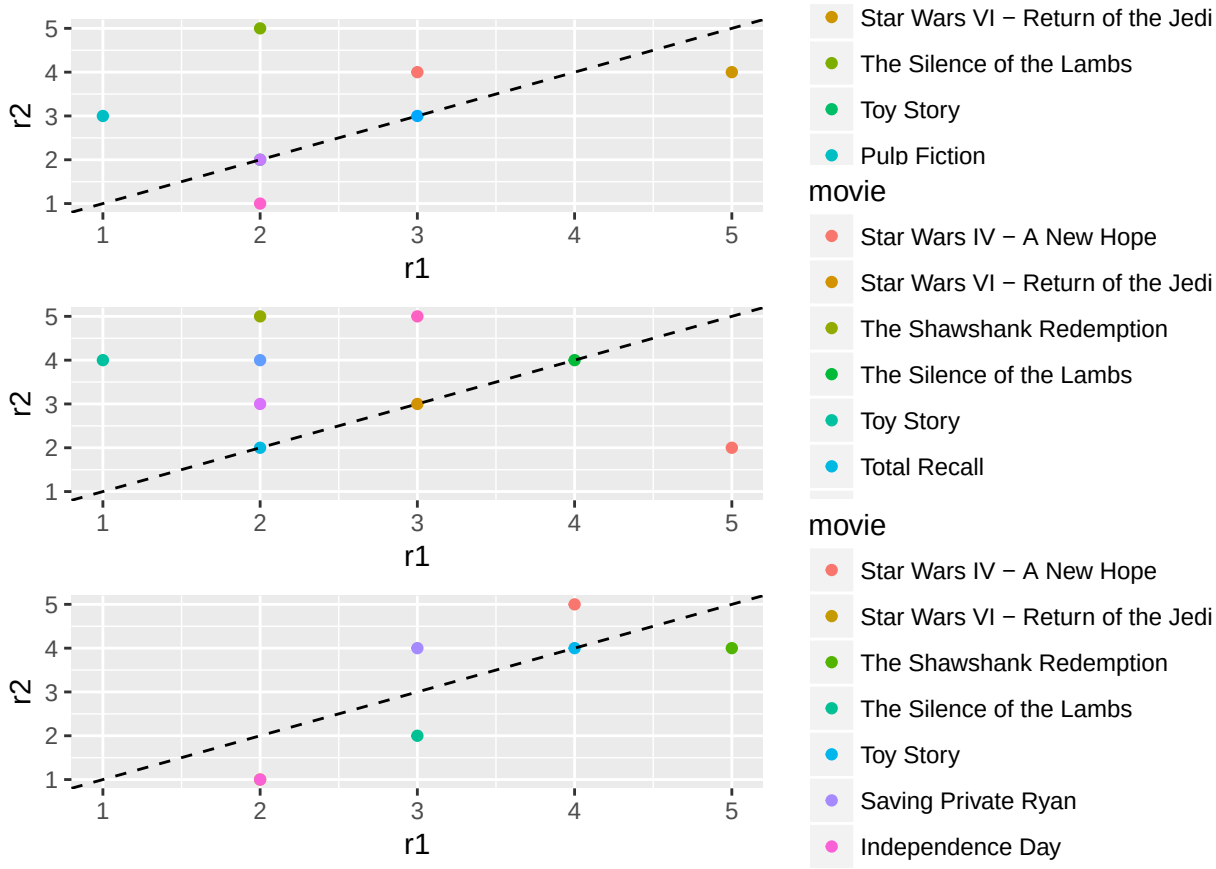
```
## [1] 0.3651484
```

2) Compare two movie critics

Compare the three users and plot the comparisons

```
p1 = correlations("Nuria", "Victoria", 1)
p2 = correlations("Maria", "Nerea", 1)
p3 = correlations("Chris", "Jim", 1)

grid.arrange(p1,p2,p3, nrow=3)
```



In the first and second plot we can see a lot of points significantly far from the straight line. The correlation between the users in both cases is weak. In the third plot, we can see most points very close to the straight line. The correlation is thus strong between Chris and Jim.

3) Top Recommendations

Get the top 5 movies recommended for Victoria. by using a weighted average of every other critic's rankings.

```
#Create a dataframe with the correlations of all users with Victoria
userCorrs = mcritics %>%
  filter(User != "Victoria") %>%
  distinct(User) %>%
  group_by(User) %>%
  mutate(correlation = correlations("Victoria", as.character(User)))

#Create a dataframe with the rating averages of all users
userAvgs = mcritics %>%
  filter(!is.na(value)) %>%
  group_by(User) %>%
  summarize(avg = mean(value))

#Get all movies not watched by Victoria
vicMovies = mcritics %>%
  filter(User == "Victoria") %>%
  filter(is.na(value))
```

```r
#Create a new dataframe with all the movies that were not watched by Victoria and the ratings provided
#Add correlation with victoria and average rating per user columns
movies = mcritics %>%
  filter(!is.na(value)) %>%
  filter(Movie %in% vicMovies$Movie) %>%
  filter(User != "Victoria") %>%
  mutate(cor_with_vic = 0, avg_user = 0)

#For each row in the movies dataframe, add the correlation with Victoria and the user's average rating
for(i in 1:nrow(movies)) {
   user = as.character(movies$User[i])
   movies$cor_with_vic[i] = userCorrs$correlation[userCorrs$User == user]
   movies$avg_user[i] = userAvgs$avg[userAvgs$User == user]
}


#get the total of correlations with Victoria and the average ratings for Victoria
totCorr = sum(abs(userCorrs$correlation))
vicAvg = userAvgs$avg[userAvgs$User == "Victoria"]

#For each movie, get the weighted average rating for Victoria
expectedVic = movies %>%
  group_by(Movie) %>%
  summarize(weightedAvg = vicAvg + sum((value - avg_user) * cor_with_vic)/totCorr) %>%
  arrange(desc(weightedAvg)) %>%
  slice(1:5)

customPrint(expectedVic, 7, "weightedAvg", "Movie")
```

```
##
##  3.7917013, 'The Matrix'
##  3.5077653, 'Forrest Gump'
##  3.3311883, 'The Sixth Sense'
##  3.1149183, 'Shakespeare in Love'
##  2.9124513, 'Blade Runner'
```

4) Top similar critics

Return the top 5 critics that are most similar to Victoria.

```r
#Use the userCorrs dataframe created above with all user correlations with Victoria and get the top 5
userCorrs = as.data.frame(userCorrs)
simVic = userCorrs %>%
  arrange(desc(correlation)) %>%
  slice(1:5)


customPrint(simVic, 7, "correlation", "User")
```

```
##
##  0.9449112, 'Rachel'
##  0.5976143, 'Ana'
##  0.5789794, 'Oriol'
```

```
##  0.4925922, 'Maria'
##  0.4273247, 'Carles'
```