Assignment 6

Dr. Nwala

Nathaniel Everett

March 21, 2018

Contents:

1. Find 3 users who are closest to you in terms of age, gender, and occupation. For each of those 3 users:

- what are their top 3 favorite films?
- bottom 3 least favorite films?

Based on the movie values in those 6 tables (3 users X (favorite + least)), choose a user that you feel is most like you. Feel free to note any outliers (e.g., "I mostly identify with user 123, except I did not like ``Ghost'' at all").

This user is the "substitute you".

Using R, I read each of the datasets into an R dataframe, then produce a subset of users, with the following code snippet:

```
df ← user[user$age == 23
        &user$gender == 'M'
        &user$occupation == 'student',]
```

This resulted in 8 users, so I picked the first three user IDs (33, 37, 66) in the table for the next part of this problem. This is where I make a dataframe for each of these three user IDs, substituting the u.data data frame and matching item ID (in data) with movie ID (in item), which will help in keeping with the movie names. The code snippet below is an example of the first user:

```
u33.data ←[data[data$user.id == 33,][c(“item.id”, “rating”)]
u33.data$item.id ← item$movie.title[match(u33.data$item.id, item$movie.id)]
```

Again the table that comes from each individual data frame shows more than 3 favorite and least favorite movies, so these will have to be narrowed down extensively. Not only that, User 33 does not have any ratings under 3, while every one of these users have more than 3 ratings at 5. For the sake of convenience, I only look at the first three and last three reviews in each table, sorted by rating. The R program used is known as substituteyou.r.

User 33

Item.id	Rating
Love Jones (1997)	3
Devil's Advocate, The (1997)	3
Scream 2 (1997)	3
Scream (1996)	4
Air Force One (1997)	4
Titanic (1997)	5

User 37

Item.id	Rating
Jurassic Park (1993)	1
Money Train (1995)	2
Arrival, The (1996)	2
Star Wars (1977)	5
Die Hard 2 (1990)	5
Stargate (1994)	5

User 66

Item.id	Rating
Excess Baggage (1997)	1
Muppet Treasure Island (1996)	1
English Patient, The (1996)	1
Ransom (1996)	5
Star Wars (1977)	5
Air Force One (1997)	5

Since I am not much of a movie buff, finding a user that is most similar to me basically refers to which movies I actually knew about and which ones I also liked greatly. Since I both know and like Star Wars, Stargate, and Die Hard 2, I chose my substitute to be User 37.

---

2. Which 5 users are most correlated to the substitute you? Which 5 users are least correlated (i.e., negative correlation)?

I use the R file correlation.r for this problem. It is the exact same as the substituteyou.r, except it will add more for the correlation equations. For convenience, I create a new name for u37.data, which is sub.me. Then to start, I gather a list of data frames for each of the user ratings in u.user:

```
user.list <- list() #gather list of data frames
for(n in 1:dim(user)) {
  user.list[[n]] <- data[data$user.id == n, ][c("item.id", "rating")]
}
```

R has a function called cor() for correlations, so I planned to use it accordingly. Another function will be defined, cordf, for correlation of the data frames. What this does is take two dataframes passed to it and the sapply() function, which runs over user.list, comparing each of the users to sub.me. The sapply function is used for the cors data frame, which is an empty list() initially. Note that df.one will be assigned as sub.me.

The correlateddata data frame will show a large table of the correlated data, along with which users are most in common with the substitute me. Since this is a large table, I have to find the ones who are most correlated (1.0) and those in common to pick out the five most correlated users. Likewise, the least most correlated users will be of value (-1.0). Again due to high amounts of users, for convenience, I take the ones with the highest incommon ratio and go down the list in numerical order.

	Correlation	incommon
754	1.0	4
93	1.0	3
310	1.0	3
433	1.0	3
596	1.0	3

	Correlation	incommon
491	-1.0	3
691	-1.0	3
80	-1.0	2
185	-1.0	2
228	-1.0	2

3. Compute ratings for all the films that the substitute you have not seen. Provide a list of the top 5 recommendations for films that the substitute you should see. Provide a list of the bottom 5 recommendations (i.e., films the substitute you is almost certain to hate).

This problem uses recommendation.r, which a continuation of the previous two R programs. Start with a correlation matrix for every movie, which gets rid of the basic parts of the data, transposing the row names:

```
item.data ← item[c(-1:-5)]
rownames ← rownames(item.data)
t.item.data ← as.data.frame(t(item.data))
```

From here on, there will be quite a lot of functions, most of them pertaining to the top 5 items, top 5 users, or both. There are also the bottom 5 for those respective functions as well. Before that, I make a recommendation based off of the similiarity between my substitute me (user 37) and a recommended movie, which is Braveheart (movie ID 22). The rest of the functions will return numeric vectors of either item IDs or user IDs. Afterwards, there is a get.ratings function, along with a get.hatings for the least favorites, both of which will actually return R data frames with item IDs as well as their scores. Take an example, the top 5 users function. The score is calculated first by using this function to get the top 5 correlated users, then their top 5 movies, then the top 5 most correlated movies to those movies.

```
Top.users ← top.5.user()
item.id ← unlist(lapply(top.users, top.5.user_items, tar.id=u.id))
item.id ← append(item.id,
                 unlist(lapply(unique(item.id), top.5.items,
                               u.id=37, #note, this is the substitute me
                               I.cors=item.cors
                               )
                        )
```

```
                    )
        as.data.frame(table(item.id))
```

The above line shows that the items in the vector (which were added from the top 5, only the ones similar) are counted and stored in a data frame. The resulting data frame is called rec, short for recommendations, and by analyzing the "total" column, I will pick out the top 5 and the bottom 5 movie recommendations. Again, for convenience, I only do the first 5 of each, and ignore the ones further down the table.

Top 5 recommendations for user 37

Item.id	Freq.x	Freq.y	total	Movie.title
125	5	1	4	Phenomenon (1996)
14	5	1	4	Postino, Il (1994)
20	5	1	4	Angels and Insects (1995)
258	4	0	4	Contact (1997)
36	5	1	4	Mad Love (1995)

Bottom 5 recommendations for user 37

Item.id	Freq.x	Freq.y	total	Movie.title
1016	0	3	-3	Con Air (1997)
2	0	3	-3	GoldenEye (1995)
300	0	3	-3	Air Force One (1997)
322	0	3	-3	Murder at 1600 (1997)
1013	0	2	-2	Anaconda (1997)

4. Choose your (the real you, not the substitute you) favorite and least favorite film from the data. For each film, generate a list of the top 5 most correlated and bottom 5 least correlated films. Based on your knowledge of the resulting films, do you agree with the results? In other words, do you personally like / dislike the resulting films?

Again, I shall build another R program off of the previous one, so that things are in a nice package, in which myfilms.r will be the ultimate culmination of the four problems. First I analyze the movie titles, picking out favorites and least favorites among them. Since I am not a huge movie buff, a lot of these movies I do not know about. I decided to pick Home Alone (1990) as my favorite (ID 94) and Robocop 3 (1993) as my least favorite (ID 1274). I make four data frames: labeled mytopdatamost, mytopdataleast, mybottomdatamost, and mybottomdataleast. The following code snippet shows how mytopdatamost was done:

```
mytopdatamost ← as.data.frame(item.cors[94,])
colnames(mytopdatamost) ← c('cor')
rows ← rownames(mytopdatamost)
head(mytopdatamost[order(mytopdatamost, decreasing=TRUE), , drop=FALSE])
```

Once I get the correlations done, again for convenience, I only look at the first five values. Note that the data frames do list my favorite film as the top value if sorted by correlation. I ignore this.

mytopdatamost

Movie id	Correlation	Movie title
63	1.0	Santa Clause, The (1994)
138	1.0	D3: The Mighty Ducks (1996)
139	1.0	Love Bug, The (1969)
225	1.0	101 Dalmatians (1996)
243	1.0	Jungle2Jungle (1997)

I'm only familiar with two of these movies, but I guess this is correlated accurately.

mytopdataleast

Movie id	Correlation	Movie title
172	-0.2330207	Empire Strikes Back, The (1980)
50	-0.2049800	Star Wars (1977)
101	-0.2049800	Heavy Metal (1981)
181	-0.2049800	Return of the Jedi (1983)
855	-0.2049800	Diva (1981)

I definitely do not agree with the Star Wars titles! The other two I haven't seen, but they're probably accurate. Maybe this correlation data was based off of genre.

mybottomdatamost

Movie id	Correlation	Movie title
38	1.0	Net, The (1995)
264	1.0	Mimic (1997)
758	1.0	Lawnmower Man 2: Beyond Cyberspace (1996)
925	1.0	Unforgettable (1996)
931	1.0	Island of Dr. Moreau, The (1996)

With the exception of Dr. Moreau, I know nothing about these movies, so I guess this is fine.

mybottomdataleast

Movie id	Correlation	Movie title
820	-0.204980	Space Jam (1996)
993	-0.204980	Hercules (1997)
1076	-0.204980	Pagemaster, The (1994)
29	-0.177123	Batman Forever (1995)
51	-0.177123	Legends of the Fall (1994)

I guess these are accurate as well.

Files included:
Assignment6report.pdf (this file)
substituteyou.r (first problem)
correlation.r (continuation of first, includes second problem)
recommendation.r (continuation of previous problems, includes third problem)
myfilm.r (full program, includes all four problems)