Assignment #2 - SQL and R

Stefano Biguzzi

9/1/2020

ETL Process

Create Connection to Local PostgreSQL database and getting tables

Creating connection

```
con <- dbConnect(
  RPostgres::Postgres(),
  dbname = "MovieRatings",
  host="localhost",
  port="5432",
  user="postgres",
  password=params$pwd)
```

Setting tables to dataframes

Loading tblPersons and removing any white spaces

```
#Loading person data
person_df <- dbGetQuery(
  con,
  ’Select * From public."tblPersons"’
)
```

```
#Clearing white spaces
for (i in names(person_df)) {
  person_df[[i]] <- trimws(
    person_df[[i]],
    which = c("both", "left", "right"),
    whitespace = "[ \t\r\n]"
  )
}
```

```
#Creating person table
kable(person_df,format = "markdown",caption = "Person Data")
```

Table 1: Person Data

PersonID	FirstName	Sex	Ethnicity	Relationship
PER_1	James	Male	Asian/Pacific Islander	Friend
PER_2	Ashley	Female	Multi-Racial	Girlfriend
PER_3	Anna	Female	White	Mother
PER_4	Alex	Male	White	Friend
PER_5	Samantha	Female	Black	Co-Worker
PER_6	Fabio	Male	White	Brother
PER_7	Daniella	Female	Black	Friend
PER_8	Justin	Male	Asian/Pacific Islander	Friend
PER_9	Dana	Female	White	Friend
PER_10	Brayan	Male	Black	Friend

Loading tblMovies and removing any white spaces

```r
#Loading movie data
movie_df <- dbGetQuery(
  con,
  ’Select * From public."tblMovies"’
)
```

```r
#Clearing movie data white spaces
for (i in names(movie_df)) {
  if (is.character(movie_df[[i]])) {
    movie_df[[i]] <- trimws(
      movie_df[[i]],
      which = c("both", "left", "right"),
      whitespace = "[ \t\r\n]"
    )
  }
}
```

```r
#Creating movie data table
kbl(movie_df, caption = "Movie Data", booktabs = T) %>%
    kable_styling(
      latex_options = c("scale_down", "hold_position")
    )
```

Table 2: Movie Data

MovieID	MovieTitle	BestPicture	OscarYear	Genre	RtCriticRating	ImdbRating
MOV_1	Avengers: Endgame	N	NA	Action/Adventure	0.94	8.4
MOV_2	Parasite	Y	2020	Thriller	0.99	8.6
MOV_3	The Shape of Water	Y	2018	Romance	0.92	7.3
MOV_4	Star Wars: Rise of Skywalker	N	NA	Action/Adventure	0.51	6.6
MOV_5	Joker	N	NA	Drama	0.68	8.5
MOV_6	1917	N	NA	Drama	0.89	8.3

Loading tblRatings and removing any white spaces

```
#Loading ratings data
ratings_df <- dbGetQuery(
  con,
  ’Select * From public."tblRatings"’
)
```

```
#Removing white spaces in rating data
for (i in names(ratings_df)) {
  if (is.character(ratings_df[[i]])) {
    ratings_df[[i]] <- trimws(
      ratings_df[[i]],
      which = c("both", "left", "right"),
      whitespace = "[ \t\r\n]"
    )
  }
}
```

```
#Creating ratings data table
kable(
  ratings_df,
  format = "markdown",
  caption = "Movie Ratings Raw"
)
```

Table 3: Movie Ratings Raw

PersonID	MovieID	MovieRating
PER_1	MOV_1	5.0
PER_1	MOV_2	5.0
PER_1	MOV_3	1.0
PER_1	MOV_4	3.0
PER_1	MOV_5	4.0
PER_1	MOV_6	4.0
PER_2	MOV_1	5.0
PER_2	MOV_2	5.0
PER_2	MOV_3	3.0
PER_2	MOV_4	4.0
PER_2	MOV_5	5.0
PER_2	MOV_6	NA
PER_3	MOV_1	NA
PER_3	MOV_2	4.0
PER_3	MOV_3	5.0
PER_3	MOV_4	NA
PER_3	MOV_5	4.0
PER_3	MOV_6	NA
PER_4	MOV_1	NA
PER_4	MOV_2	NA
PER_4	MOV_3	NA
PER_4	MOV_4	3.0
PER_4	MOV_5	NA
PER_4	MOV_6	NA

PersonID	MovieID	MovieRating
PER_5	MOV_1	4.0
PER_5	MOV_2	NA
PER_5	MOV_3	3.5
PER_5	MOV_4	2.0
PER_5	MOV_5	4.0
PER_5	MOV_6	NA
PER_6	MOV_1	5.0
PER_6	MOV_2	NA
PER_6	MOV_3	NA
PER_6	MOV_4	1.0
PER_6	MOV_5	1.0
PER_6	MOV_6	NA
PER_7	MOV_1	3.5
PER_7	MOV_2	NA
PER_7	MOV_3	NA
PER_7	MOV_4	NA
PER_7	MOV_5	4.5
PER_7	MOV_6	NA
PER_8	MOV_1	4.0
PER_8	MOV_2	5.0
PER_8	MOV_3	NA
PER_8	MOV_4	2.0
PER_8	MOV_5	3.0
PER_8	MOV_6	4.0
PER_9	MOV_1	NA
PER_9	MOV_2	5.0
PER_9	MOV_3	NA
PER_9	MOV_4	4.0
PER_9	MOV_5	NA
PER_9	MOV_6	NA
PER_10	MOV_1	4.0
PER_10	MOV_2	4.0
PER_10	MOV_3	5.0
PER_10	MOV_4	3.0
PER_10	MOV_5	5.0
PER_10	MOV_6	3.0

Dealing with missing data

In the ratings dataframe for the MovieRating column, I decided to create two dataframes, one for seen movies with ratings and one for non seen movies. This can allow us to run different statistical analyses on the different types of movies, seen and unseen.

Figure out how many nulls there are in the MovieRatings column

```
#Counting the nulls
kable(
  table(is.na(ratings_df$MovieRating)),
  format = "markdown"
)
```

Var1	Freq
FALSE	37
TRUE	23

Create dataframe for movie ratings that were seen

```
#Subsetting data
seen_ratings_df <- subset(
  ratings_df,!is.na(ratings_df$MovieRating)
)
```

```
#Creating table of seen movies
rownames(seen_ratings_df) <- NULL
kable(
  seen_ratings_df,
  format = "markdown",
  caption = "Rated Movies"
)
```

Table 5: Rated Movies

PersonID	MovieID	MovieRating
PER_1	MOV_1	5.0
PER_1	MOV_2	5.0
PER_1	MOV_3	1.0
PER_1	MOV_4	3.0
PER_1	MOV_5	4.0
PER_1	MOV_6	4.0
PER_2	MOV_1	5.0
PER_2	MOV_2	5.0
PER_2	MOV_3	3.0
PER_2	MOV_4	4.0
PER_2	MOV_5	5.0
PER_3	MOV_2	4.0

PersonID	MovieID	MovieRating
PER_3	MOV_3	5.0
PER_3	MOV_5	4.0
PER_4	MOV_4	3.0
PER_5	MOV_1	4.0
PER_5	MOV_3	3.5
PER_5	MOV_4	2.0
PER_5	MOV_5	4.0
PER_6	MOV_1	5.0
PER_6	MOV_4	1.0
PER_6	MOV_5	1.0
PER_7	MOV_1	3.5
PER_7	MOV_5	4.5
PER_8	MOV_1	4.0
PER_8	MOV_2	5.0
PER_8	MOV_4	2.0
PER_8	MOV_5	3.0
PER_8	MOV_6	4.0
PER_9	MOV_2	5.0
PER_9	MOV_4	4.0
PER_10	MOV_1	4.0
PER_10	MOV_2	4.0
PER_10	MOV_3	5.0
PER_10	MOV_4	3.0
PER_10	MOV_5	5.0
PER_10	MOV_6	3.0

Create dataframe for movies that were not seen

```r
#Subsetting movies with no rating
not_seen_df <-
  subset(ratings_df,is.na(ratings_df$MovieRating))
```

```r
#Dropping movie rating
not_seen_df <-
  subset(not_seen_df, select = -MovieRating)
```

```r
#Create the list of PersonID and MovieID
rownames(not_seen_df) <- NULL
kable(not_seen_df,format = "markdown",caption = "Not Seen Movie List")
```

Table 6: Not Seen Movie List

PersonID	MovieID
PER_2	MOV_6
PER_3	MOV_1
PER_3	MOV_4
PER_3	MOV_6
PER_4	MOV_1
PER_4	MOV_2

PersonID	MovieID
PER_4	MOV_3
PER_4	MOV_5
PER_4	MOV_6
PER_5	MOV_2
PER_5	MOV_6
PER_6	MOV_2
PER_6	MOV_3
PER_6	MOV_6
PER_7	MOV_2
PER_7	MOV_3
PER_7	MOV_4
PER_7	MOV_6
PER_8	MOV_3
PER_9	MOV_1
PER_9	MOV_3
PER_9	MOV_5
PER_9	MOV_6

Recommendation Analysis

To create a recommendation of a movie for the people that have not seen all six movies, I want to start by grabbing the *not_seen_df* and merge on the movie title and movie genre. Then I want to subset the *seen_ratings_df* to only the people that have a row in the *not_seen_df*. I will then summarize the *seen_ratings_df* to understand the number of movies and the average rating by genre. Finally I will merge the summary data to the *not_seen_df* and find a movies that match genre and the genre has an average rating above 3.5. I will recommend movies based on those criteria.

Merge the not seen data with movie data

Take the not seen data and merge it with the movie data to get movie title and movie genre. This will help in understanding the genre of the movies that were not watched

```r
#Merging the movie_df with not_seen_df
not_seen_movie_merge_df <- merge(
  not_seen_df,
  movie_df[,c("MovieID","MovieTitle","Genre")],
  by = "MovieID"
)
```

```r
#Creating unique list of movies not seen
not_seen_unique_df <- not_seen_movie_merge_df[
  row.names(unique(
    not_seen_movie_merge_df[,c(
      "MovieTitle", "Genre"
    )]
  )),
  c("MovieTitle","Genre")
]
```

```r
#Creating the table for not seen movie merge table
rownames(not_seen_unique_df) <- NULL
kable(
  not_seen_unique_df,
  format = "markdown",
  caption = "List of not seen movies"
)
```

Table 7: List of not seen movies

MovieTitle	Genre
Avengers: Endgame	Action/Adventure
Parasite	Thriller
The Shape of Water	Romance
Star Wars: Rise of Skywalker	Action/Adventure
Joker	Drama
1917	Drama

Subset the seen movie ratings

Subsetting the *seen_ratings_df* to only people that are in the *not_seen_df*

```r
#Subsetting seen movies to only people who are also in the not seen movie list
person_to_recommend_df <-
  subset(
    seen_ratings_df,
    PersonID %in% not_seen_df$PersonID)
```

```r
#Creating the table of ratings for people who have not seen all movies
rownames(person_to_recommend_df) <- NULL
kable(
  person_to_recommend_df,
  format = "markdown",
  caption = "Ratings for movies of people who show up in not seen list"
)
```

Table 8: Ratings for movies of people who show up in not seen list

PersonID	MovieID	MovieRating
PER_2	MOV_1	5.0
PER_2	MOV_2	5.0
PER_2	MOV_3	3.0
PER_2	MOV_4	4.0
PER_2	MOV_5	5.0
PER_3	MOV_2	4.0
PER_3	MOV_3	5.0
PER_3	MOV_5	4.0
PER_4	MOV_4	3.0
PER_5	MOV_1	4.0
PER_5	MOV_3	3.5

PersonID	MovieID	MovieRating
PER_5	MOV_4	2.0
PER_5	MOV_5	4.0
PER_6	MOV_1	5.0
PER_6	MOV_4	1.0
PER_6	MOV_5	1.0
PER_7	MOV_1	3.5
PER_7	MOV_5	4.5
PER_8	MOV_1	4.0
PER_8	MOV_2	5.0
PER_8	MOV_4	2.0
PER_8	MOV_5	3.0
PER_8	MOV_6	4.0
PER_9	MOV_2	5.0
PER_9	MOV_4	4.0

Merging movie data to subsetted ratings data frame

Adding column movie title and genre from *movie_df* to the new data frame *person_to_recommend_df*

```
#Adding movie info to table of ratings for people who have not seen all movies
person_to_recommend_df <- merge(
  merge(
    person_to_recommend_df,
    movie_df[,c("MovieID","MovieTitle","Genre")],
    by = "MovieID"
  ),
  person_df[,c("PersonID","FirstName")],
  by = "PersonID"
)
```

```
#Creating rating table for people who watched less than 6 movies
kable(
  person_to_recommend_df[c("FirstName","MovieTitle","Genre","MovieRating")],
  format = "markdown",
  caption = "Person and movie ratings for people who watched < 6 movies"
)
```

Table 9: Person and movie ratings for people who watched < 6 movies

FirstName	MovieTitle	Genre	MovieRating
Ashley	Avengers: Endgame	Action/Adventure	5.0
Ashley	The Shape of Water	Romance	3.0
Ashley	Parasite	Thriller	5.0
Ashley	Joker	Drama	5.0
Ashley	Star Wars: Rise of Skywalker	Action/Adventure	4.0
Anna	Parasite	Thriller	4.0
Anna	The Shape of Water	Romance	5.0
Anna	Joker	Drama	4.0
Alex	Star Wars: Rise of Skywalker	Action/Adventure	3.0

FirstName	MovieTitle	Genre	MovieRating
Samantha	The Shape of Water	Romance	3.5
Samantha	Avengers: Endgame	Action/Adventure	4.0
Samantha	Star Wars: Rise of Skywalker	Action/Adventure	2.0
Samantha	Joker	Drama	4.0
Fabio	Avengers: Endgame	Action/Adventure	5.0
Fabio	Joker	Drama	1.0
Fabio	Star Wars: Rise of Skywalker	Action/Adventure	1.0
Daniella	Joker	Drama	4.5
Daniella	Avengers: Endgame	Action/Adventure	3.5
Justin	Avengers: Endgame	Action/Adventure	4.0
Justin	Star Wars: Rise of Skywalker	Action/Adventure	2.0
Justin	1917	Drama	4.0
Justin	Parasite	Thriller	5.0
Justin	Joker	Drama	3.0
Dana	Star Wars: Rise of Skywalker	Action/Adventure	4.0
Dana	Parasite	Thriller	5.0

Creating summary statistics table

Summarizing the subsetted dataframe, *person_to_recommend_df*. Finding out the number of movies and the avg rating per person and genre

```
#Creating number of movies seen per genre
seen_summary_df <- data.frame(
  person_to_recommend_df %>%
    group_by(PersonID,FirstName) %>%
    count(Genre)
)
```

```
#Renaming the count column
names(seen_summary_df)[names(seen_summary_df) == "n"] <- "NumberMoviesSeen"
```

```
#Creating the average rating by person and genre
seen_rating_average_df <- data.frame(
  person_to_recommend_df %>%
    group_by(PersonID,Genre) %>%
    summarize_at(vars(MovieRating),mean)
)
```

```
#Creating the final version of seen summary dataframe
seen_summary_df <- merge(
  seen_summary_df,
  seen_rating_average_df,
  by = c("PersonID","Genre")
)
```

```
#Renaming the movie rating column to average genre rating
names(seen_summary_df)[names(seen_summary_df) == "MovieRating"] <- "AverageGenreRating"
```

```
#Creating table of seen summary dataframe
kable(
  seen_summary_df[,c(
    "FirstName",
    "Genre",
    "NumberMoviesSeen",
    "AverageGenreRating"
  )],
  format = "markdown",
  caption = "Summary stats for people with < 6 watched movies"
)
```

Table 10: Summary stats for people with < 6 watched movies

FirstName	Genre	NumberMoviesSeen	AverageGenreRating
Ashley	Action/Adventure	2	4.5
Ashley	Drama	1	5.0
Ashley	Romance	1	3.0
Ashley	Thriller	1	5.0
Anna	Drama	1	4.0
Anna	Romance	1	5.0
Anna	Thriller	1	4.0
Alex	Action/Adventure	1	3.0
Samantha	Action/Adventure	2	3.0
Samantha	Drama	1	4.0
Samantha	Romance	1	3.5
Fabio	Action/Adventure	2	3.0
Fabio	Drama	1	1.0
Daniella	Action/Adventure	1	3.5
Daniella	Drama	1	4.5
Justin	Action/Adventure	2	3.0
Justin	Drama	2	3.5
Justin	Thriller	1	5.0
Dana	Action/Adventure	1	4.0
Dana	Thriller	1	5.0

Creating recommended movie list

The final stage is to create a recommended movie list for people who have not seen all the six movies. ### Create recommended list This list includes movies in genres that were low rated by person. The next step would be to subset this data to include only the movies that are part of a genre above a 3.5 mean rating.

```
#Creating the recommend dataframe
recommend_df <- merge(
  not_seen_movie_merge_df,
  seen_summary_df,
  by = c("PersonID","Genre")
)
```

```r
#Creating table to show list of movies not seen and average genre rating
kable(
  recommend_df,
  format = "latex",
  booktabs=T,
  caption = "List movies not seen with average genre rating"
) %>% kable_styling(
  latex_options=c("scale_down","hold_position")
)
```

Table 11: List movies not seen with average genre rating

PersonID	Genre	MovieID	MovieTitle	FirstName	NumberMoviesSeen	AverageGenreRating
PER_2	Drama	MOV_6	1917	Ashley	1	5.0
PER_3	Drama	MOV_6	1917	Anna	1	4.0
PER_4	Action/Adventure	MOV_1	Avengers: Endgame	Alex	1	3.0
PER_5	Drama	MOV_6	1917	Samantha	1	4.0
PER_6	Drama	MOV_6	1917	Fabio	1	1.0
PER_7	Action/Adventure	MOV_4	Star Wars: Rise of Skywalker	Daniella	1	3.5
PER_7	Drama	MOV_6	1917	Daniella	1	4.5
PER_9	Action/Adventure	MOV_1	Avengers: Endgame	Dana	1	4.0

Subset full recommended list to higher rated genres

```r
#Subsetting recommend table to only pull movies with genre ratings above 3.5
final_df <- subset(recommend_df,AverageGenreRating >= 3.5)
```

```r
#Reordering columns for readability
col_order <- c("FirstName","MovieTitle","Genre","AverageGenreRating")
final_df <- final_df[,col_order]
```

```r
#Creating full list of movies to recommend
rownames(final_df) <- NULL
kable(
  final_df,
  format = "markdown",
  caption = "full list of movies with highest genre rating to recommend"
)
```

Table 12: full list of movies with highest genre rating to recommend

FirstName	MovieTitle	Genre	AverageGenreRating
Ashley	1917	Drama	5.0
Anna	1917	Drama	4.0
Samantha	1917	Drama	4.0
Daniella	Star Wars: Rise of Skywalker	Action/Adventure	3.5
Daniella	1917	Drama	4.5
Dana	Avengers: Endgame	Action/Adventure	4.0

Conclusion

```
#Subsetting final list to only one movie per person
final_table <- final_df %>%
  group_by(FirstName) %>%
  top_n(1, AverageGenreRating)
```

```
#Creating final recommend list with one movie per person
rownames(final_table) <- NULL
kable(
  final_table,
  format = "markdown",
  col.names = c("First Name","Movie","Genre","Average Genre Rating"),
  caption = "Final List of Recommended Movies",
  align = "lllc"
)
```

Table 13: Final List of Recommended Movies

First Name	Movie	Genre	Average Genre Rating
Ashley	1917	Drama	5.0
Anna	1917	Drama	4.0
Samantha	1917	Drama	4.0
Daniella	1917	Drama	4.5
Dana	Avengers: Endgame	Action/Adventure	4.0

Based on *Table 13: Final List of Recommended Movies*, I would have to recommend 1917 to Ashley, Anna, Samantha, and Daniella, while recommending Avengers: Endgame to Dana. These recommendations should be taken with a grain of salt as there is not enough data to make an especially accurate recommendation.

This accuracy issue could be remedied by collecting more ratings and more information about each movie. First, I would like to collect more movie ratings expanding the genre pool and making the genre rating averages more robust. Secondly, I would love to add additional data for each movie. Some examples include, the gender of the lead actor, release date (to group by month of release), and Oscar nominations and awards won.

Creating a larger dataset with more descriptive information about the movies and expanding the available genres of movies can help create a better movie recommendation list for each survey participant.