DATA607 - Week 2 Assignment

Donald Butler

9/05/2021

Introduction

Created a survey using Google Forms and asked friends and family to take part. The survey collected demographic data (Age,Sex,Marital Status), then asked them to rate 25 movies that released in 2019 on a scale of 1 to 5. There were 18 participants with 154 ratings of the 25 movies.

Create SQL Database

The following SQL Commands were used to generate a normalized database based on the responses collected in the survey.

```
-- 1. Create movie_review Schema
CREATE SCHEMA IF NOT EXISTS movie_review;
USE movie_review;

-- 2. Create and Populate Movies table
DROP TABLE IF EXISTS Movies;
CREATE TABLE Movies (
    MovieId INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    Title varchar(100) NOT NULL,
    ReleaseYear INT,
    RunTime INT,
    URL varchar(100) NOT NULL
);
INSERT INTO Movies (Title, ReleaseYear, RunTime, URL) VALUES
    ('Once Upon a Time ... In Hollywood',2019,161,
      'https://www.imdb.com/title/tt7131622'),
    ('Spider-Man: Far from Home',2019,129,
      'https://www.imdb.com/title/tt6320628'),
    ('Avengers: Endgame',2019,181,
      'https://www.imdb.com/title/tt4154796'),
    ('Joker',2019,122,
      'https://www.imdb.com/title/tt7286456'),
    ('Knives Out',2019,130,
      'https://www.imdb.com/title/tt8946378'),
    ('The Gentlemen',2019,113,
      'https://www.imdb.com/title/tt8367814'),
    ('Captain Marvel',2019,123,
      'https://www.imdb.com/title/tt4154664'),
    ('Doctor Sleep',2019,152,
```

```
      'https://www.imdb.com/title/tt5606664'),
    ('1917',2019,119,
      'https://www.imdb.com/title/tt8579674'),
    ('Ad Astra',2019,123,
      'https://www.imdb.com/title/tt2935510'),
    ('Little Women',2019,135,
      'https://www.imdb.com/title/tt3281548'),
    ('Rocketman',2019,121,
      'https://www.imdb.com/title/tt2066051'),
    ('Ford v Ferrari',2019,152,
      'https://www.imdb.com/title/tt1950186'),
    ('Jumanji: The Next Level',2019,123,
      'https://www.imdb.com/title/tt7975244'),
    ('Star Wars: Episode IX - The Rise of Skywalker',2019,141,
      'https://www.imdb.com/title/tt2527338'),
    ('The Irishman',2019,209,
      'https://www.imdb.com/title/tt1302006'),
    ('Yesterday',2019,116,
      'https://www.imdb.com/title/tt8079248'),
    ('John Wick: Chapter 3 - Parabellum',2019,130,
      'https://www.imdb.com/title/tt6146586'),
    ('Shazam!',2019,132,
      'https://www.imdb.com/title/tt0448115'),
    ('Fast & Furious Presents: Hobbs & Shaw',2019,137,
      'https://www.imdb.com/title/tt6806448'),
    ('Escape Room',2019,99,
      'https://www.imdb.com/title/tt5886046'),
    ('Gemini Man',2019,117,
      'https://www.imdb.com/title/tt1025100'),
    ('Terminator: Dark Fate',2019,128,
      'https://www.imdb.com/title/tt6450804'),
    ('Hellboy',2019,120,
      'https://www.imdb.com/title/tt2274648'),
    ('It Chapter Two',2019,169,
      'https://www.imdb.com/title/tt7349950');

-- 3. Create and Populate Reviewers table

DROP TABLE IF EXISTS Reviewers;
CREATE TABLE Reviewers (
    ReviewerId INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
    Age INT,
    Sex varchar(1),
    Marital varchar(1)
);
INSERT INTO Reviewers (Age,Sex,Marital) VALUES
    (48,'M','Y'),
    (44,'F','Y'),
    (28,'F','N'),
    (24,'F','N'),
    (72,'F','Y'),
    (27,'F','N'),
    (50,'M','Y'),
```

```
    (21,'F','N'),
    (53,'F','N'),
    (48,'M','Y'),
    (25,'F','Y'),
    (62,'F','N'),
    (23,'M','N'),
    (39,'F','Y'),
    (38,'M','Y'),
    (16,'M','N'),
    (41,'F','Y'),
    (44,'M','Y');

-- 4. Create and Populate Reviews table

DROP TABLE IF EXISTS Reviews;
CREATE TABLE Reviews (
    ReviewerId INT NOT NULL,
    MovieId INT NOT NULL,
    Rating INT
);
INSERT INTO Reviews (ReviewerId,MovieId,Rating) VALUES
    (1,1,5),(1,3,4),(1,4,5),(1,5,4),(1,6,4),(1,9,5),(1,12,5),
    (1,13,4),(1,15,3),(1,17,5),(1,18,3),(1,21,3),
    (2,1,5),(2,3,5),(2,4,4),(2,5,5),(2,6,3),(2,7,3),(2,11,4),
    (2,13,5),(2,14,4),(2,15,4),(2,17,3),(2,18,3),(2,22,4),
    (3,11,3),
    (4,1,4),(4,2,3),(4,3,5),(4,4,5),(4,5,2),(4,6,2),(4,7,5),
    (4,8,2),(4,9,2),(4,10,2),(4,11,2),(4,12,2),(4,13,2),
    (4,14,5),(4,15,5),(4,16,2),(4,17,2),(4,18,2),(4,19,2),
    (4,20,5),(4,21,5),(4,22,5),(4,23,5),(4,24,5),(4,25,5),
    (5,2,5),(5,14,4),(5,15,5),(5,16,5),
    (6,1,2),(6,2,3),(6,3,3),(6,4,2),(6,5,5),(6,6,2),(6,7,2),
    (6,8,2),(6,9,2),(6,10,2),(6,11,5),(6,12,5),(6,13,2),
    (6,14,2),(6,15,1),(6,16,2),(6,17,2),(6,18,1),(6,19,4),
    (6,20,2),(6,21,1),(6,22,1),(6,23,2),(6,24,1),(6,25,1),
    (7,2,3),(7,3,5),(7,4,3),(7,5,3),(7,8,2),(7,9,5),(7,10,4),
    (7,13,4),(7,15,2),(7,16,5),(7,18,4),(7,19,2),(7,20,2),
    (7,22,2),(7,23,2),(7,24,5),
    (8,5,5),(8,9,4),(8,15,3),
    (9,3,5),(9,15,5),
    (10,3,5),(10,7,5),(10,14,5),(10,15,5),(10,18,5),(10,22,5),
    (11,2,3),(11,6,4),(11,11,5),
    (12,4,2),
    (13,2,5),(13,3,5),(13,5,2),(13,9,5),(13,15,3),(13,16,5),
    (14,12,5),(14,16,4),
    (15,12,4),
    (16,1,5),(16,2,4),(16,3,5),(16,4,4),(16,5,4),(16,6,4),
    (16,7,4),(16,9,5),(16,12,3),(16,13,5),(16,14,3),(16,15,3),
    (16,16,5),(16,18,5),(16,19,4),(16,20,4),(16,22,4),(16,23,3),
    (16,24,4),(16,25,2),
    (17,2,3),(17,3,4),(17,4,4),(17,5,2),(17,7,1),(17,8,5),
    (17,9,3),(17,15,3),(17,16,1),(17,25,5),
    (18,14,4),(18,16,2),(18,18,5),(18,19,3);
```

Connect to SQL Database

Connecting to a MySQL database using the RMySQL package. For reproducing this work, the **RMySQL** package and **dbConnect** functions may need to be updated to match your database.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(RMySQL)
```

```
## Warning: package 'RMySQL' was built under R version 4.1.1

## Loading required package: DBI
```

```
cnn <- DBI::dbConnect(MySQL(),
                      user = rstudioapi::askForPassword('Username'),
                      password = rstudioapi::askForPassword('password'),
                      dbname = 'movie_review',
                      host = 'localhost')

reviewers_db <- dbReadTable(cnn,'reviewers')
movies_db <- dbReadTable(cnn,'movies')
reviews_db <- dbReadTable(cnn,'reviews')
```

Get Movie Review Survey Data

Read the data from the database and Tidy to give each movie its own variable in the data frame. If the movie was not reviewed by the reviewer, the rating will be null.

```
reviews <- reviewers_db %>%
  left_join(movies_db %>%
              left_join(reviews_db,by = 'MovieId') %>%
              select(ReviewerId,Title,Rating) %>%
              pivot_wider(names_from = Title,values_from = Rating),by = 'ReviewerId') %>%
  arrange(ReviewerId)

head(reviews)
```

```
##   ReviewerId Age Sex Marital Once Upon a Time ... In Hollywood
## 1          1  48   M       Y                                 5
## 2          2  44   F       Y                                 5
## 3          3  28   F       N                                NA
## 4          4  24   F       N                                 4
## 5          5  72   F       Y                                NA
## 6          6  27   F       N                                 2
##   Spider-Man: Far from Home Avengers: Endgame Joker Knives Out The Gentlemen
## 1                        NA                 4     5          4             4
## 2                        NA                 5     4          5             3
## 3                        NA                NA    NA         NA            NA
## 4                         3                 5     5          2             2
## 5                         5                NA    NA         NA            NA
## 6                         3                 3     2          5             2
##   Captain Marvel Doctor Sleep 1917 Ad Astra Little Women Rocketman
## 1             NA           NA    5       NA           NA         5
## 2              3           NA   NA       NA            4        NA
## 3             NA           NA   NA       NA            3        NA
## 4              5            2    2        2            2         2
## 5             NA           NA   NA       NA           NA        NA
## 6              2            2    2        2            5         5
##   Ford v Ferrari Jumanji: The Next Level
## 1              4                      NA
## 2              5                       4
## 3             NA                      NA
## 4              2                       5
## 5             NA                       4
## 6              2                       2
##   Star Wars: Episode IX - The Rise of Skywalker The Irishman Yesterday
## 1                                             3           NA         5
## 2                                             4           NA         3
## 3                                            NA           NA        NA
## 4                                             5            2         2
## 5                                             5            5        NA
## 6                                             1            2         2
##   John Wick: Chapter 3 - Parabellum Shazam!
## 1                                3      NA
## 2                                3      NA
## 3                               NA      NA
## 4                                2       2
## 5                               NA      NA
## 6                                1       4
##   Fast & Furious Presents: Hobbs & Shaw Escape Room Gemini Man
## 1                                    NA           3         NA
## 2                                    NA          NA          4
## 3                                    NA          NA         NA
## 4                                     5           5          5
## 5                                    NA          NA         NA
## 6                                     2           1          1
##   Terminator: Dark Fate Hellboy It Chapter Two
## 1                    NA      NA             NA
## 2                    NA      NA             NA
## 3                    NA      NA             NA
## 4                     5       5              5
```

```
## 5                    NA      NA             NA
## 6                     2       1              1
```

Most popular movie

Calculate the average rating for each movie and sort by the most popular.

```
summarize_at(reviews,vars(5:29),list(mean = mean),na.rm = TRUE) %>%
  pivot_longer(cols = everything(),names_to = 'Title',values_to = 'Rating') %>%
  arrange(desc(Rating))
```

```
## # A tibble: 25 x 2
##    Title                                Rating
##    <chr>                                 <dbl>
##  1 Avengers: Endgame_mean                 4.6
##  2 Once Upon a Time ... In Hollywood_mean  4.2
##  3 Rocketman_mean                         4
##  4 1917_mean                              3.88
##  5 Jumanji: The Next Level_mean           3.86
##  6 Little Women_mean                      3.8
##  7 Hellboy_mean                           3.75
##  8 Ford v Ferrari_mean                    3.67
##  9 Spider-Man: Far from Home_mean         3.62
## 10 Joker_mean                             3.62
## # ... with 15 more rows
```

Avengers: Endgame was the most popular movie rated in my survey, and **Ad Astra** was the least favorite.

Findings and Recommendations

Google Forms was pretty easy to setup and was able to get deployed in just an hour. Since the data set was small, I used a Google Sheet to read the data and generate the SQL needed to create the database tables. For a larger set, I would definitely look to normalizing and importing into the database.

I collected some demographic data with my reviews, but didn't make use of it in this analysis. I'm still pretty new to R and have a lot to learn about grouping summaries, and manipulating the data frames to do what I want.