jjimenez_data607_h2

Jean Jimenez

Data 607 Homework 2

SQL and R

Overview

This week's assignment was to survey people about movies. The goal was to use the data collected to practice creating tables/databases in SQL and using R to access the data from SQL.

4 people in the survey are real and 3 people have randomly generated data.

Each person was asked to rate the following movies from 1 (Horrible) to 5 (Great):

- Barbie
- Oppenheimer
- Choose Love
- Lets Clone Tyrone
- You People
- Murder Mystery 2

Data was recorded manually because it was collected through various mediums.

SQL

After obtaining the data on movie ratings, I created a new schema and database on mySQL called 607_hw2. Afterwards, I created a table and inserted the data that I collected. The following is the SQL code for that.

Creating the database and table:

```
CREATE DATABASE IF NOT EXISTS 607_hw2;

USE 607_hw2;

CREATE TABLE IF NOT EXISTS movie_ratings (
  ind_id INT NOT NULL,
  name VARCHAR(45) NOT NULL,
  movie_title VARCHAR(45) NOT NULL,
  movie_rating INT NOT NULL,
  PRIMARY KEY (ind_id, movie_title)
);
```

Adding the data to the table movie_ratings:

```
INSERT INTO movie_ratings (ind_id, name, movie_title, movie_rating)
VALUES (1, 'liz', 'barbie', 4),
       (1, 'liz', 'oppenheimer', 4),
       (1, 'liz', 'murder_mystery_2', 3),
       (1, 'liz', 'choose_love', 1),
       (1, 'liz', 'lets_clone_tyrone', 4),
       (1, 'liz', 'you_people', 5),
       (2, 'ray', 'barbie', 3),
       (2, 'ray', 'oppenheimer', 5),
       (2, 'ray', 'murder_mystery_2', 3),
       (2, 'ray', 'choose_love', 2),
       (2, 'ray', 'lets_clone_tyrone', 4),
       (2, 'ray', 'you_people', 4),
       (3, 'sam', 'barbie', 5),
       (3, 'sam', 'oppenheimer', 4),
       (3, 'sam', 'murder_mystery_2', 4),
       (3, 'sam', 'choose_love', 3),
       (3, 'sam', 'lets_clone_tyrone', 1),
       (3, 'sam', 'you_people', 4),
       (4, 'tracy', 'barbie', 2),
       (4, 'tracy', 'oppenheimer', 5),
       (4, 'tracy', 'murder_mystery_2', 4),
       (4, 'tracy', 'choose_love', 1),
       (4, 'tracy', 'lets_clone_tyrone', 5),
       (4, 'tracy', 'you_people', 3),
       (5, 'anne', 'barbie', 1),
       (5, 'anne', 'oppenheimer', 4),
```

```
        (5, 'anne', 'murder_mystery_2', 1),
        (5, 'anne', 'choose_love', 5),
        (5, 'anne', 'lets_clone_tyrone', 5),
        (5, 'anne', 'you_people', 4),
        (6, 'anthony', 'barbie', 5),
        (6, 'anthony', 'oppenheimer', 4),
        (6, 'anthony', 'murder_mystery_2', 2),
        (6, 'anthony', 'choose_love', 2),
        (6, 'anthony', 'lets_clone_tyrone', 4),
        (6, 'anthony', 'you_people', 2),
        (7, 'alex', 'barbie', 1),
        (7, 'alex', 'oppenheimer', 1),
        (7, 'alex', 'murder_mystery_2', 3),
        (7, 'alex', 'choose_love', 2),
        (7, 'alex', 'lets_clone_tyrone', 5),
        (7, 'alex', 'you_people', 3);
```

Note: this code wont necessarily work here on rmd file because I haven't yet figured out how to connect the rmd file with mySQL localhost.

Importing to R

Now, I will import the data from the mySQL table over to R to use.

To do this, I used the libraries DBI and RMYSQL.

I established connection to mySQL localhost, provided my credentials, and called in the data frame

(Note: code won't run on rmd file because I don't know how to do it without sharing password but will learn. If you choose to run this code, first run the SQL code above in your own SQL server, link your SQL credentials to the code below)

```
library(DBI)
library(RMySQL)

#connect to the database

connect = dbConnect(RMySQL::MySQL(),
                    dbname = "607_hw2",
                    host = "localhost",
                    user = "user",
```

```
                  password = "password")

#getting table using sql query

query = "SELECT * FROM movie_rating"

#now we have our Data frame

movie_ratings = dbGetQuery(connect, query)

head(movie_ratings)
```

Discussion

Handling Missing Data

There are many different ways to handle missing data. One way would be by creating new data/data points from ones that already exist.

After importing the data from SQL, I made a new data frame that had each movies median rating aggregated. I essentially took each movie, made a list of their ratings, found the median, and visualized it using ggplot.

```
#Lets make a data frame that shows the median rating per movie

#split function to extract the ratings of each movie

list_of_ratings = split(movie_ratings$rating, movie_ratings$movie_title)

list_of_ratings

#median rating for each movie

median_ratings = aggregate(rating ~ movie_title, data = movie_ratings, FUN = median)
median_ratings

#Barplot showing median rating using ggplot2

library(ggplot2)

# Create the bar plot
```

```r
ggplot(median_ratings, aes(x = movie_title, y = rating)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Median Ratings by Movie",
       x = "Movie Title",
       y = "Median Rating")
```

Another example of adding missing data from data that already exists is the following. I created a key for each movie and added their genre to the data frame. Then I visualized each person's individual movie genre preference.

```r
#will now add genre to each movie

genre_key = c("fantasy", "interactive", "sci_fi", "comedy", "drama","comedy")
names(genre_key) = median_ratings$movie_title

# Convert to data frame for merging

genre_df = data.frame(movie_title = names(genre_key), genre = genre_key)

# Merge with the movie_ratings data frame

movie_ratings = merge(movie_ratings, genre_df, by = "movie_title")

head(movie_ratings)
unique(movie_ratings$movie_title)
#now ill use ggplot to plot the rating distribution of each genre

ggplot(movie_ratings, aes(x = genre, y = rating, color = ind_name)) +
  geom_jitter(width = 0.3, height = 0) +
  labs(title = "Distribution of Ratings by Genre",
       x = "Genre",
       y = "Rating") +
  theme(legend.position = "bottom")

#going to use dplyr to
#collect/aggregate every person's movie ratings and genres

library(dplyr)

#Take movie ratings data frame
```

```r
#pipe it through group_by, which will group it by each individual person and genre
#pipe the grouped data into summarise which will spit out the average rating per genre
#ungroup so that it is separated by each individual person

aggrigate_ratings = movie_ratings %>%
  group_by(ind_name, genre) %>%
  summarise(avg_rating=mean(rating)) %>%
  ungroup()

aggrigate_ratings

#we can use this in the future to calculate stuff for each individual person

#heat map for each person and their avg rating per genre

ggplot(aggrigate_ratings, aes(x = ind_name, y = genre)) +
  geom_tile(aes(fill = avg_rating), color = "white") +
  scale_fill_gradient(low = "white", high = "steelblue") +
  labs(title = "Average Ratings Heatmap",
       x = "Individual",
       y = "Genre")
```

Another point bought up from the assignment was what happens if the people sampled didn't watch all the movies. After all, I didn't provide the people that I sampled an option to select 'N/A'. If I had provided the N/A option, I would remove data points that are NULL., depending on the sample size and other factors.

There are many different ways to handle missing data.

To make this assignment better the next time, I will provide a N/A option and use either google/microsoft forms, or RedCap. Also, I would increase the sample size and make my code work without showing mySQL password.