DATA607 - Assign2

Rachel Greenlee

9/4/2020

Introduction

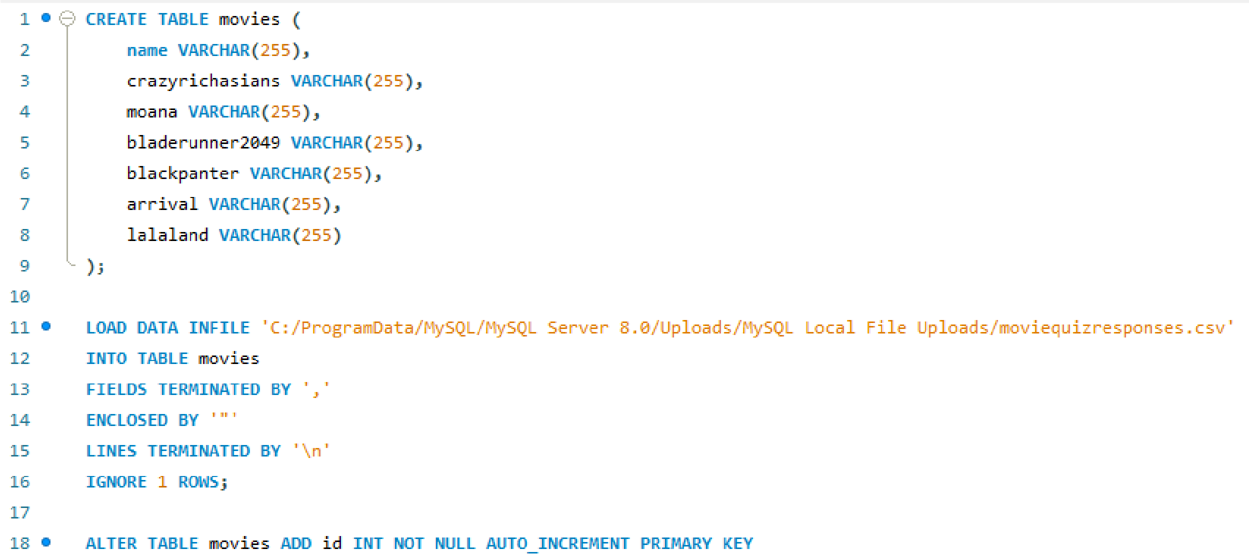
For this assignment I went through the following steps in order to collect data using Google Forms and ultimately have a small dataframe in R with my friends' movie ratings ready for further analysis.

Step 1 - Data Collection

I used Google Forms to set up an informal survey with just 1 matrix question listing all movies that I could send to my 5 friends. You can see that survey at this link: Movie Survey

Step 2 - Export to CSV and Import into MySQL

Google Forms conveniently provides an export to CSV option, which I downloaded to my local files. In MySQL Workbench I was able to write code to create the empty table and load the data from the CSV file. Finally, I added an ID as a primary key. A screenshot of the code from my MySQL session is below. (I also learned how to put an image in RMarkdown!)



```
CREATE TABLE movies (
    name VARCHAR(255),
    crazyrichasians VARCHAR(255),
    moana VARCHAR(255),
    bladerunner2049 VARCHAR(255),
    blackpanter VARCHAR(255),
    arrival VARCHAR(255),
    lalaland VARCHAR(255)
);

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/MySQL Local File Uploads/moviequizresponses.csv'
INTO TABLE movies
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;

ALTER TABLE movies ADD id INT NOT NULL AUTO_INCREMENT PRIMARY KEY
```

Figure 1: Image of MySQL Code.

Step 3 - Connect R to MySQL to create dataframe

First I load the RMySQL package in order to connect with MySQL. I also load the keyring package so I can store my password and hide it from the world. I won't show the code I've used to store them, but it looks like this:

key_set_with_value(service = "mysql", username = "root", password = "XXXXXX")

```
## Loading required package: DBI
```

Now we must access the dataset from MySQL.And let's preview it quick as well.

```
mysqlpass <- key_get('mysql', 'root')

rachdb = dbConnect(MySQL(), user='root', password=mysqlpass,
  dbname='607assign2_movies', host='localhost')

movies <- dbGetQuery(rachdb, "select * from movies")

movies
```

```
##      name crazyrichasians moana bladerunner2049 blackpanter  arrival   lalaland
## 1     Ben        Not seen     2               5           4        4        3\r
## 2   Suman               4     3        Not seen           5 Not seen        2\r
## 3 Daniela        Not seen     4        Not seen           5 Not seen        3\r
## 4    Kate        Not seen     5        Not seen           5 Not seen Not seen\r
## 5    José               2     2               3           5 Not seen Not seen\r
##   id
## 1  1
## 2  2
## 3  3
## 4  4
## 5  5
```

Step 4 - Cleaning the Dataframe

I see my movie ratings imported as characters, lets change them to numerics. In doing so, the "Not seen" value will render a NA - which is just fine for our purposes.

```
movies$crazyrichasians <- as.numeric(movies$crazyrichasians)
movies$moana <- as.numeric(movies$moana)
movies$bladerunner2049 <- as.numeric(movies$bladerunner2049)
movies$blackpanter <- as.numeric(movies$blackpanter)
movies$arrival <- as.numeric(movies$arrival)
movies$lalaland <- as.numeric(movies$lalaland)
```

Next lets put the ID field first.

```
movies <- movies[, c(8, 1, 2, 3, 4, 5, 6, 7)]
movies
```

```
##   id    name crazyrichasians moana bladerunner2049 blackpanter arrival lalaland
## 1  1     Ben              NA     2               5           4       4        3
## 2  2   Suman               4     3              NA           5      NA        2
## 3  3 Daniela              NA     4              NA           5      NA        3
## 4  4    Kate              NA     5              NA           5      NA       NA
## 5  5    José               2     2               3           5      NA       NA
```

Step 5 - Analysis

Is there a movie that you would recommend or not recommend to one of the participants? Explain your reasoning. For my two friends, José and Kate, who have not watched (NA)La La Land - I would recommend they do not watch it in the future as my first three friends all rated it only a 2-3, despite there being some, at first glance, variability in their ratings of the other movies. Also, Black Panther appears to be a universal favorite of my friends with everyone having seen the movie, and 4/5 ratings being a perfect 5!

To manipulate this dataframe a bit further and try to find something meaningful to plot from such a small dataset, I've made a new dataframe that just displays the minimum and maximum values for each of the 6 movies after dropping the first too irrelevant rows (ID and name).

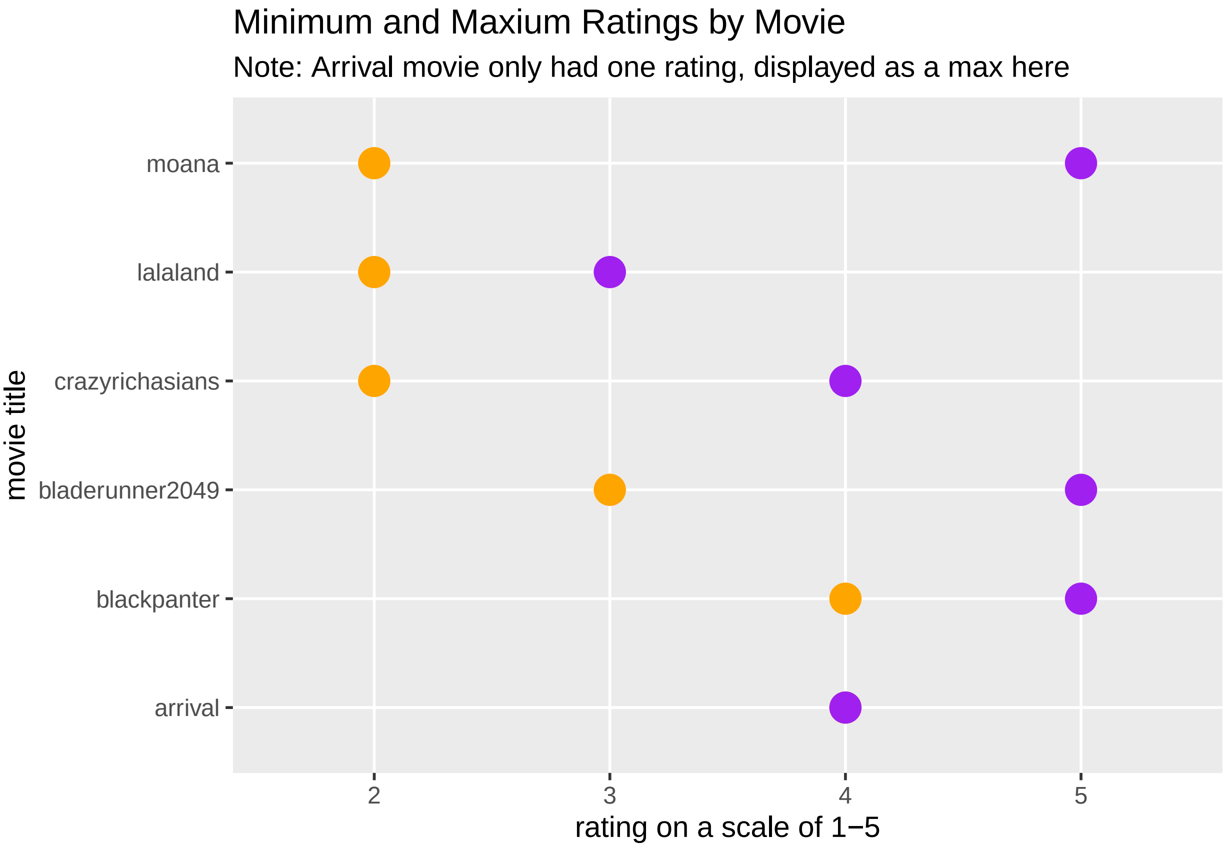
```
maxs <- sapply(movies, max, na.rm = TRUE)
mins <- sapply(movies, min, na.rm = TRUE)

movieminmax <- data.frame(mins, maxs, stringsAsFactors = FALSE)

movieminmax[-c(1, 2),]
```

```
##                 mins maxs
## crazyrichasians    2    4
## moana              2    5
## bladerunner2049    3    5
## blackpanter        4    5
## arrival            4    4
## lalaland           2    3
```

```
movieminmax <- movieminmax[-c(1, 2),]
```

I then used the ggplot2 package to make a simple plot that shows each movie with it's minimum rating (orange dot) and maximum rating (purple dot). This isn't a rigorous analysis, but does make it easier to see that Moana has quite a spread of ratings compared to Black Panther. This graph hides the fact that some of these movies only had 1 or 2 raters.

```
library(ggplot2)
library(scales)
ggplot(movieminmax) +
  geom_point(aes(x=rownames(movieminmax), y=mins), color = "orange", size=5) +
  geom_point(aes(x=rownames(movieminmax), y=maxs), color = "purple", size=5) +
  coord_flip() +
  ylab('rating on a scale of 1-5') +
  xlab('movie title') +
  labs(title="Minimum and Maxium Ratings by Movie",
    subtitle="Note: Arrival movie only had one rating, displayed as a max here")
```



Conclusion

This process shows how, with very little code, a simple survey done on a free platform like Google Forms can be brought into MySql and then accessed by R Studio. Along the way I learned how to use the keyring package to store my passwords securely, how to display an image in R Markdown, a new way to use ggplot2, and most importantly how to access a MySQL database from R Studio.

Of course, if you wanted to expand this further, you could incorporate variables such as movie genres or main actors (Ryan Gosling is in both Blade Runner 2049 and La La Land, but they are very different genres).