DATA 607 Assignment 2

Ying Fang Lee

9/15/2024

Part 1: Build Table

I asked 5 friends: Ray, Joe, Mily, Amy, and March to rate 6 animated movies released in 2024: “IF”, “Kung Fu Panda 4”,“Despicable Me4”, “Inside Out 2”, “The Garfield Movie”, and “Thelma the Unicorn” from 1 to 5 and recorded their answers in below table saved as csv file. If they have not seen the movie yet, the rating is recorded as “NA”. File was uploaded to my github called Movie_Rating.csv (https://raw.githubusercontent.com/amily52131/DATA607/main/Assignment_2/Movie_Rating.csv).

```
library(readr)
Movie_Rate <- read_csv("https://raw.githubusercontent.com/amily52131/DATA607/main/Assignment_2/Movie_Rating.csv")
```

```
## Rows: 6 Columns: 6
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): Movie Name
## dbl (5): Ray, Joe, Mily, Amy, March
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
print(Movie_Rate)
```

```
## # A tibble: 6 × 6
##   `Movie Name`          Ray   Joe  Mily   Amy March
##   <chr>               <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 IF                      2     3     2     4    NA
## 2 Kung Fu Panda 4         5     4     5     3     2
## 3 Despicable Me 4         3     4     4     3     3
## 4 Inside Out 2            4     5     5    NA     5
## 5 The Garfield Movie     NA     3     2     1    NA
## 6 Thelma the Unicorn     NA    NA     3    NA     3
```

Part 2: Store data in SQL database

Connecting R to Database

```
library(DBI)      #Database infrastructure for R
library(RMySQL)   #Translating R and MySQL
library(tidyverse)#Organize data
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ purrr     1.0.2
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
#create a config.yml file with database information in it
#default:
#  datawarehouse:
#    driver: 'MySQL()'
#    server: 'Name_or_link_for_Database'
#    uid: 'Database_Username'
#    pwd: 'Database_Password'
#    database: 'Name_of_Database'

#Use get() from config library to get database information from config.yml
dbConfig <- config::get("datawarehouse")

#Connecting to Database with config info
mydb <- dbConnect(
  MySQL(),
  user = dbConfig$uid,
  password = dbConfig$pwd,
  dbname = dbConfig$database,
  host = dbConfig$server
)
```

Installed MySQL workbench and connected with the azure database with the information provided in the email. I manually created a new table called Movie_Rates using the *CREATE TABLE* function and then append the movie and rating using the *INSERT INTO* function.

```
CREATE TABLE Movie_Rates (
  Movie_Name CHAR(100),
  Ray INT,
  Joe INT,
  Mily INT,
  Amy INT,
  March INT
);
```

```
INSERT INTO Movie_Rates
(Movie_Name, Ray, Joe, Mily, Amy, March) VALUES
("IF", 2, 3, 2, 4, null),
("Kung Fu Panda 4", 5, 4, 4, 3, 2),
("Despicable Me 4", 3, 4, 4, 3, 3),
("Inside Out 2", 4, 5, 5, null, 5),
("The Garfield Movie", null, 3, 2, 1, null),
("Thelma the Unicorn", null, null, 3, null, 3);
```

```
SELECT * FROM Movie_Rates;
```

6 records

Movie_Name	Ray	Joe	Mily	Amy	March
IF	2	3	2	4	NA
Kung Fu Panda 4	5	4	4	3	2
Despicable Me 4	3	4	4	3	3
Inside Out 2	4	5	5	NA	5
The Garfield Movie	NA	3	2	1	NA
Thelma the Unicorn	NA	NA	3	NA	3

Part 3: Transfer Data from SQL database to R dataframe

```
#To see the tables in the database
dbListTables(mydb)
```

```
## [1] "movie_rates"      "movie_rates_test"
```

```
#Convert the information from MySQL to R data frame
MovieRates <- tbl(mydb, "movie_rates") #convert source data to table
Rates_df <- collect(MovieRates) #convert table to R data frame
Rates_df
```

```
## # A tibble: 6 × 6
##   Movie_Name           Ray   Joe  Mily   Amy March
##   <chr>              <int> <int> <int> <int> <int>
## 1 IF                     2     3     2     4    NA
## 2 Kung Fu Panda 4        5     4     4     3     2
## 3 Despicable Me 4        3     4     4     3     3
## 4 Inside Out 2           4     5     5    NA     5
## 5 The Garfield Movie    NA     3     2     1    NA
## 6 Thelma the Unicorn    NA    NA     3    NA     3
```

```
dbDisconnect(mydb) #Disconnect from database
```

```
## [1] TRUE
```

Part 4: Missing data strategy

My approach to missing data in this particular observation is to leave it in the table as “null”. I did not want to assign it a value because when calculating statistics, R can ignore null values and calculate the value based on the observed instances. For example, when calculating the average movie rating of each movie we can see that the rating goes down significantly if we count the average with all 5 people instead of based on if people have seen the movie. Movie rating should be calculated based on the rates people have observed.

```
# Calculate average movie rating by removing the null value.
Average_w_NA <- Rates_df %>%
  select(Ray,Joe, Mily, Amy, March) %>%
  rowMeans(na.rm = TRUE)

# New data frame replacing NA with 0
Rates_df_0 <- Rates_df
Rates_df_0 <- replace(Rates_df_0, is.na(Rates_df_0),0)

# Calculate average movie rating with 0 if not observed
Average_w_0 <- Rates_df_0 %>%
  select(Ray,Joe, Mily, Amy, March) %>%
  rowMeans()

Average_w_0
```

```
## [1] 2.2 3.6 3.4 3.8 1.2 1.2
```

```
Average_w_NA
```

```
## [1] 2.75 3.60 3.40 4.75 2.00 3.00
```

Bonus Challenge Questions:

Are you able to use a password without having to share the password with people who are viewing your code?

I did little research on how to approach this issue since I did not want to upload to github my username and password for the database connection. After researching on best practices (https://solutions.posit.co/connections/db/best-practices/managing-credentials/) for managing credentials in R. I chose the method of using a config file so that my credential is not visible to people who are viewing my code.