Assignment 2

Coco Donovan

2023-02-08

Loading the Necessary Packages

```
library(RMySQL)
```

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

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(reshape)
```

```
##
## Attaching package: 'reshape'

## The following object is masked from 'package:dplyr':
##
##     rename
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following objects are masked from 'package:reshape':
##
##     colsplit, melt, recast
```

```
library(ggplot2)
```

Establishing the MySQL Connection

```
mysqlconnection = dbConnect(RMySQL::MySQL(),
                            dbname='MovieRatings',
                            host='127.0.0.1',
                            port=3306,
                            user='root',
                            password='Navonod17')
```

Inspecting the available tables in the database

```
dbListTables(mysqlconnection)
```

```
## [1] "movie_info"    "movie_ratings"
```

Converting movie_ratings databasze into a R dataframe

```
result = dbSendQuery(mysqlconnection, "select * from movie_ratings")

movie_ratings <- fetch(result)

# I added and id here, because I did not want to take an email (for the
# purposes of anonymity), but I did want a unique identifier

 colnames(movie_ratings) <- c('Timestamp',
         '[M3GAN]',
         '[The Whale]',
         '[The Menu]',
         '[Black Panther: Wakanda Forever]',
         '[Everything Everywhere All At Once]',
         '[Knives Out]')

respondent_id <- c(1:nrow(movie_ratings))

movie_ratings$respondent_id <- respondent_id
```

Reshaping the data frame

```
# I want to keep this here, because it shows that I really did work hard on this
# and that there is always room to grow!

ratings <- movie_ratings[2:ncol(movie_ratings)]
```

```
df <- data.frame(matrix(ncol = 3, nrow = ncol(ratings)*nrow(ratings)))

x <- c("Timestamp","movie_names", "movie_ratings")
colnames(df) <- x

movies_names <- c()
individual_ratings <- c()
timestamps <- c()

for (y in 1:ncol(ratings)) {
  for (x in 1: nrow(ratings)) {
    timestamps <- c(timestamps, movie_ratings$Timestamp[x])
    movies_names <- c(movies_names, colnames(ratings)[y])
    individual_ratings <- c(individual_ratings, ratings[x,y])
  }
}

df$Timestamp <- timestamps
df$movie_names <- movies_names
df$movie_ratings <- individual_ratings
```

Reshaping the data frame (condensed version)

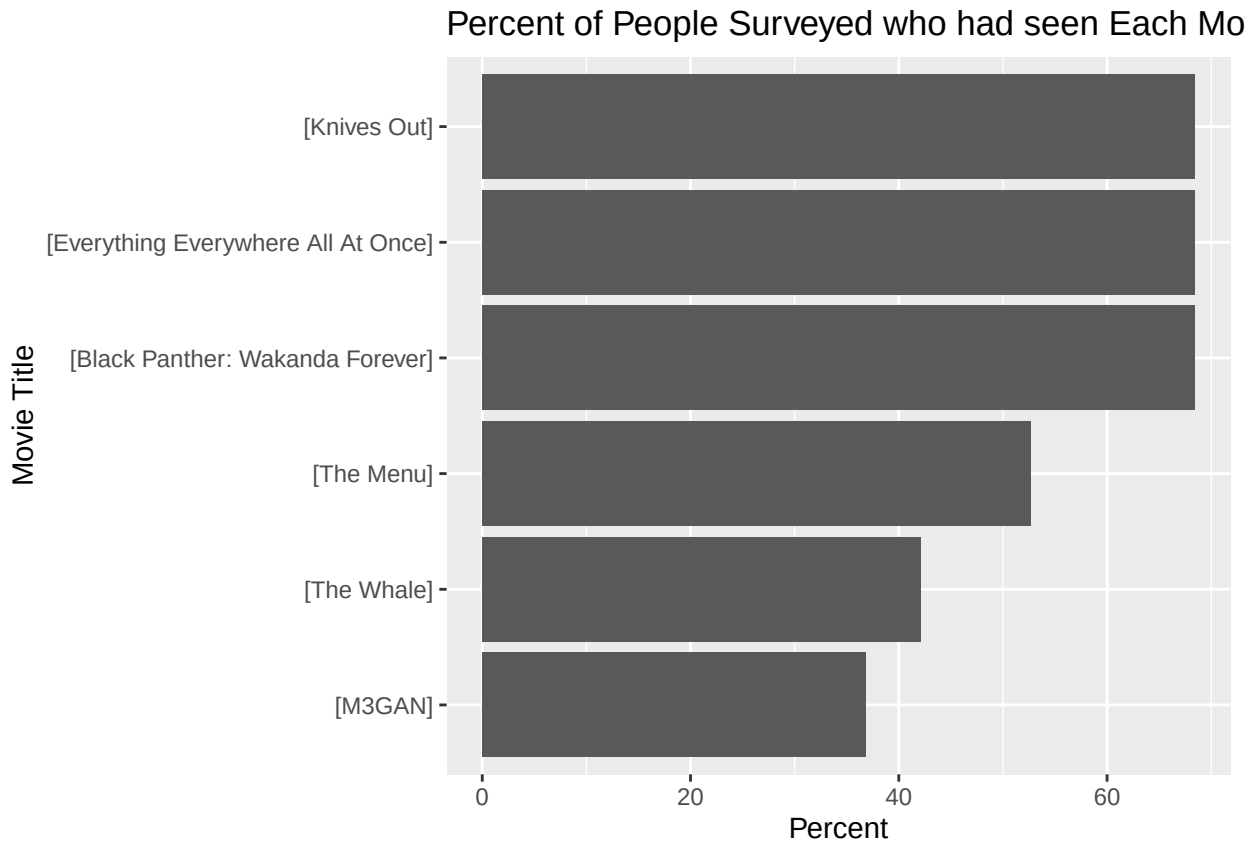
```
melted_movies <- melt(movie_ratings, Movie_Names <- c("Timestamp", "respondent_id"))

colnames(melted_movies) <- c("Timestamp", "respondent_id", "Movie_Name", "Rating")
```

Percent of total group who watched each movie

```
percent_watched <- melted_movies %>%
  group_by(Movie_Name) %>%
  summarise(percents = (100.0 * sum(Rating != "N/A"))/n_distinct(respondent_id)) %>%
  arrange(percents)

percent_watched
```

```
## # A tibble: 6 x 2
##   Movie_Name                          percents
##   <fct>                                  <dbl>
## 1 [M3GAN]                                 36.8
## 2 [The Whale]                             42.1
## 3 [The Menu]                              52.6
## 4 [Black Panther: Wakanda Forever]        68.4
## 5 [Everything Everywhere All At Once]     68.4
## 6 [Knives Out]                            68.4
```

```
percent_watched_graph <- ggplot(percent_watched, aes(x=percents, y=Movie_Name)) +
  geom_bar(stat = "identity") +
  labs(title = "Percent of People Surveyed who had seen Each Movie",
       x = "Percent",
       y = "Movie Title")

percent_watched_graph
```



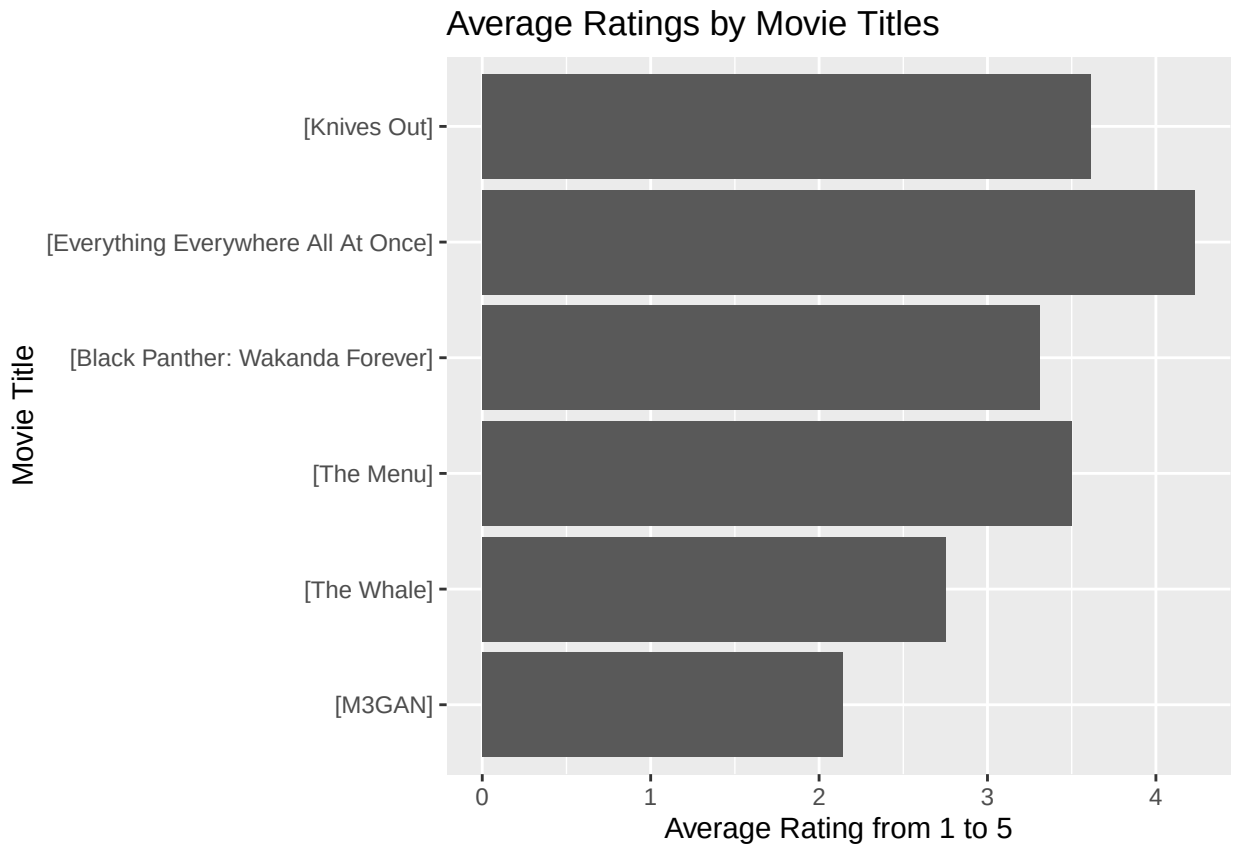
Average Rating

```
Avg_Ratings <- melted_movies %>%
  group_by(Movie_Name) %>%
  filter(Rating != "N/A") %>%
  summarise(Avgs = sum(as.double(Rating))/sum(Rating != "N/A"))

Avg_Ratings
```

```
## # A tibble: 6 x 2
##   Movie_Name                          Avgs
##   <fct>                              <dbl>
## 1 [M3GAN]                             2.14
## 2 [The Whale]                         2.75
## 3 [The Menu]                          3.5
```

```
## 4 [Black Panther: Wakanda Forever]     3.31
## 5 [Everything Everywhere All At Once]  4.23
## 6 [Knives Out]                         3.62
```

```
Avgs_graph <- ggplot(percent_watched, aes(x=Avg_Ratings$Avgs, y=Movie_Name)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Ratings by Movie Titles",
       x = "Average Rating from 1 to 5",
       y = "Movie Title")

Avgs_graph
```



Converting movie_info into a R dataframe

```
#for reference "movie_info" is a quick table I put together and added to my
#MySQL so that I could work with some outside characteristics

result = dbSendQuery(mysqlconnection, "select * from movie_info")

movie_info <- fetch(result)
```

Joining movie_ratings and movie_info for analysis

```
Avg_Ratings_and_Info <- merge(x = Avg_Ratings, y = movie_info, by = "Movie_Name",
                              all.x = TRUE)

Avg_Ratings_and_Info
```

```
##                            Movie_Name     Avgs   Genre Release_Day
## 1    [Black Panther: Wakanda Forever] 3.307692  Action          11
## 2 [Everything Everywhere All At Once] 4.230769  Comedy          25
## 3                          [Knives Out] 3.615385 Mystery          27
## 4                              [M3GAN] 2.142857  Horror           6
## 5                           [The Menu] 3.500000  Horror          18
## 6                          [The Whale] 2.750000   Drama           4
##   Release_Month Release_Year Box_Office    Budget Runtime IMDb_Rating
## 1            11         2022  842200000 250000000     161         7.0
## 2             3         2022  106000000  25000000     139         8.0
## 3            11         2019  311900000  40000000     130         7.9
## 4             1         2023  150500000  12000000     102         6.4
## 5            11         2022   79300000  30000000     106         7.2
## 6             9         2022   15900000   3000000     117         8.0
```

```
# IMDb does ratings out of 10, whereas we did rankings out of 5

Avg_Ratings_and_Info$Scaled_IMDb_Rating <- Avg_Ratings_and_Info$IMDb_Rating/2.0

Percent_Watched_and_Info <- merge(x = percent_watched, y = movie_info, by = "Movie_Name",
                                  all.x = TRUE)
```

Difference from my survey's average ratings compare to IMDb's ratings

```
Avg_Diffs <- Avg_Ratings_and_Info %>%
  summarise(Movie_Name, Difference = Avgs - Scaled_IMDb_Rating) %>%
  arrange(desc(Difference))

Avg_Diffs
```

```
##                            Movie_Name Difference
## 1 [Everything Everywhere All At Once]  0.2307692
## 2                           [The Menu] -0.1000000
## 3    [Black Panther: Wakanda Forever] -0.1923077
## 4                          [Knives Out] -0.3346154
## 5                              [M3GAN] -1.0571429
## 6                          [The Whale] -1.2500000
```

```
# M3GAN and The Whale having the largest difference from the IMDb Rating makes
# sense as they were the least watched
```

```
# Seems like the folks who answered my survey are more critical than the IMDb
# average

ggplot(Avg_Diffs, aes(x = Difference, y = Movie_Name, fill = Difference)) +
  geom_bar(stat = "identity") +
  labs(title = "The Difference between the Average Ratings for my Survey Compared with IMDb's Ratings",
       y = "Movie Name")
```