# Movie_Rating with SQL

Enid Roman

2022-09-11

I chose six recent films and asked five different imaginary individuals to rate each of the movies they had seen from a scale of 1 to 5.

Data Collection

I created a database called movie_rating in a MySQL workbench to store the data in individual tables:

Table 1: movie Table 2: name Table 3: review

The SQL code for table creation:

CREATE TABLE ‘movie’ ( ‘movie_id’ int NOT NULL, ‘title’ varchar(45) NOT NULL, ‘length’ varchar(45) NOT NULL, PRIMARY KEY (‘movie_id’))

CREATE TABLE ‘name’( ‘name_id’ int NOT NULL, ‘first_name’ varchar(45) NOT NULL, ‘age’ varchar(45) NOT NULL, PRIMARY KEY (’name_id))

CREATE TABLE ‘review’ ( ‘review_id’ int NOT NULL, ‘movie_id’ varchar(45) NOT NULL, ‘name_id’ varchar(45) NOT NULL, ‘rating’ int DEFAULT NULL, ‘review’ varchar(45) DEFAULT NULL, PRIMARY KEY (‘review_id’) )

First I installed and uploaded the packages I needed.

```
# First I installed and uploaded the packages I needed.

#install.packages("RMySQL")
#install.packages("DBI")
library(RMySQL)
```

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

```
library(DBI)
library(dbplyr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.1
## v readr   2.1.2     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::ident()  masks dbplyr::ident()
## x dplyr::lag()    masks stats::lag()
## x dplyr::sql()    masks dbplyr::sql()
```

```
library(ggplot2)
```

You need to create a password to access the local database.

```
# I then connected MySQL to R to upload my database, movie_rating

mydb = dbConnect(RMySQL::MySQL(),
       dbname='movie_rating',
       host='127.0.0.1',
       port=3306,
       user='root',
       password=rstudioapi::askForPassword("database password"))
```

```
# I previewed the tables.

dbListTables(mydb)
```

```
## [1] "movie"  "name"   "review"
```

Here I see the oldest of the individual is 39 and the youngest is 7.

```
# I wrote a query to show name table.

name_table <- dbSendQuery(mydb, "SELECT * FROM name;")
dbFetch(name_table)
```

```
##   name_id first_name age
## 1       1    Anthony  27
## 2       2   Josaline  17
## 3       3      Leila   7
## 4       4      Angie  37
## 5       5   Jonathan  39
```

Here I see the longest in length time is The Good Nurse and the shortest is Texas Chainsaw Massacre.

```
# I wrote a query to show movie table.

movie_table <- dbSendQuery(mydb, "SELECT * FROM movie;")
dbFetch(movie_table)
```

```
##   movie_id                   title length
## 1        1        The Adam Project    106
## 2        2 Texas Chainsaw Massacre     81
## 3        3               Pinocchio     90
## 4        4          The Good Nurse    121
## 5        5             Tall Girl 2     97
## 6        6         Against The Ice    102
```

```
# I wrote a query to show review table.

review_table <- dbSendQuery(mydb, "SELECT * FROM review")
dbFetch(review_table)
```

```
##    review_id movie_id name_id rating      review
## 1          1        1       1      5 Great movie
## 2          2        2       1      4      Bloody
## 3          3        3       1      3 Didn’t Care
## 4          4        4       1      3      Boring
## 5          5        5       1      2         Not
## 6          6        6       1      4          Ok
## 7          7        1       2      5    Loved it
## 8          8        2       2      5      Bloody
## 9          9        3       2      5       Enjoy
## 10        10        4       2      4          Ok
## 11        11        5       2      4       Funny
## 12        12        6       2      3          Ok
## 13        13        1       3      4        Good
## 14        14        2       3      0          NA
## 15        15        3       3      5    Loved it
## 16        16        4       3      2  Don’t care
## 17        17        5       3      4       Funny
## 18        18        6       3      2      Boring
## 19        19        1       4      5       Great
## 20        20        2       4      4       Scary
## 21        21        3       4      5       Loved
## 22        22        4       4      4        Good
## 23        23        5       4      2      Stupid
## 24        24        6       4      4        Good
## 25        25        1       5      5       Great
## 26        26        2       5      5      Bloody
## 27        27        3       5      4        Good
## 28        28        4       5      4          Ok
## 29        29        5       5      2      Really
## 30        30        6       5      4        Good
```

```
# I then joined all three tables movie, name, and review to create one table called movie_rating.

movie_rating <- dbSendQuery(mydb, "SELECT
M.title As 'Title',
M.length AS 'Length',
N.first_name AS 'Name',
N.age AS 'Age',
R.rating As 'Rating',
R.review AS 'Review'
FROM movie AS M
JOIN review AS R
ON M.movie_id = R.movie_id
JOIN name AS N
ON N.name_id = R.name_id;")
#dbFetch(movie_rating)
data<-fetch(movie_rating)
print(data)
```

```
##                       Title Length     Name Age Rating      Review
## 1         The Adam Project    106  Anthony  27      5 Great movie
## 2  Texas Chainsaw Massacre     81  Anthony  27      4      Bloody
## 3                Pinocchio     90  Anthony  27      3 Didn’t Care
## 4           The Good Nurse    121  Anthony  27      3     Boring
## 5              Tall Girl 2     97  Anthony  27      2        Not
## 6          Against The Ice    102  Anthony  27      4         Ok
## 7         The Adam Project    106 Josaline  17      5   Loved it
## 8  Texas Chainsaw Massacre     81 Josaline  17      5     Bloody
## 9                Pinocchio     90 Josaline  17      5      Enjoy
## 10          The Good Nurse    121 Josaline  17      4         Ok
## 11             Tall Girl 2     97 Josaline  17      4      Funny
## 12         Against The Ice    102 Josaline  17      3         Ok
## 13        The Adam Project    106    Leila   7      4       Good
## 14 Texas Chainsaw Massacre     81    Leila   7      0         NA
## 15               Pinocchio     90    Leila   7      5   Loved it
## 16          The Good Nurse    121    Leila   7      2 Don’t care
## 17             Tall Girl 2     97    Leila   7      4      Funny
## 18         Against The Ice    102    Leila   7      2     Boring
## 19        The Adam Project    106    Angie  37      5      Great
## 20 Texas Chainsaw Massacre     81    Angie  37      4      Scary
## 21               Pinocchio     90    Angie  37      5      Loved
## 22          The Good Nurse    121    Angie  37      4       Good
## 23             Tall Girl 2     97    Angie  37      2     Stupid
## 24         Against The Ice    102    Angie  37      4       Good
## 25        The Adam Project    106 Jonathan  39      5      Great
## 26 Texas Chainsaw Massacre     81 Jonathan  39      5     Bloody
## 27               Pinocchio     90 Jonathan  39      4       Good
## 28          The Good Nurse    121 Jonathan  39      4         Ok
## 29             Tall Girl 2     97 Jonathan  39      2     Really
## 30         Against The Ice    102 Jonathan  39      4       Good
```

```r
# Checked the structure of the data. 30 rows. 6 columns.

str(data, vec.len = 1)
```

```
## 'data.frame':    30 obs. of  6 variables:
##  $ Title : chr  "The Adam Project" ...
##  $ Length: chr  "106" ...
##  $ Name  : chr  "Anthony" ...
##  $ Age   : chr  "27" ...
##  $ Rating: int  5 4 ...
##  $ Review: chr  "Great movie" ...
```

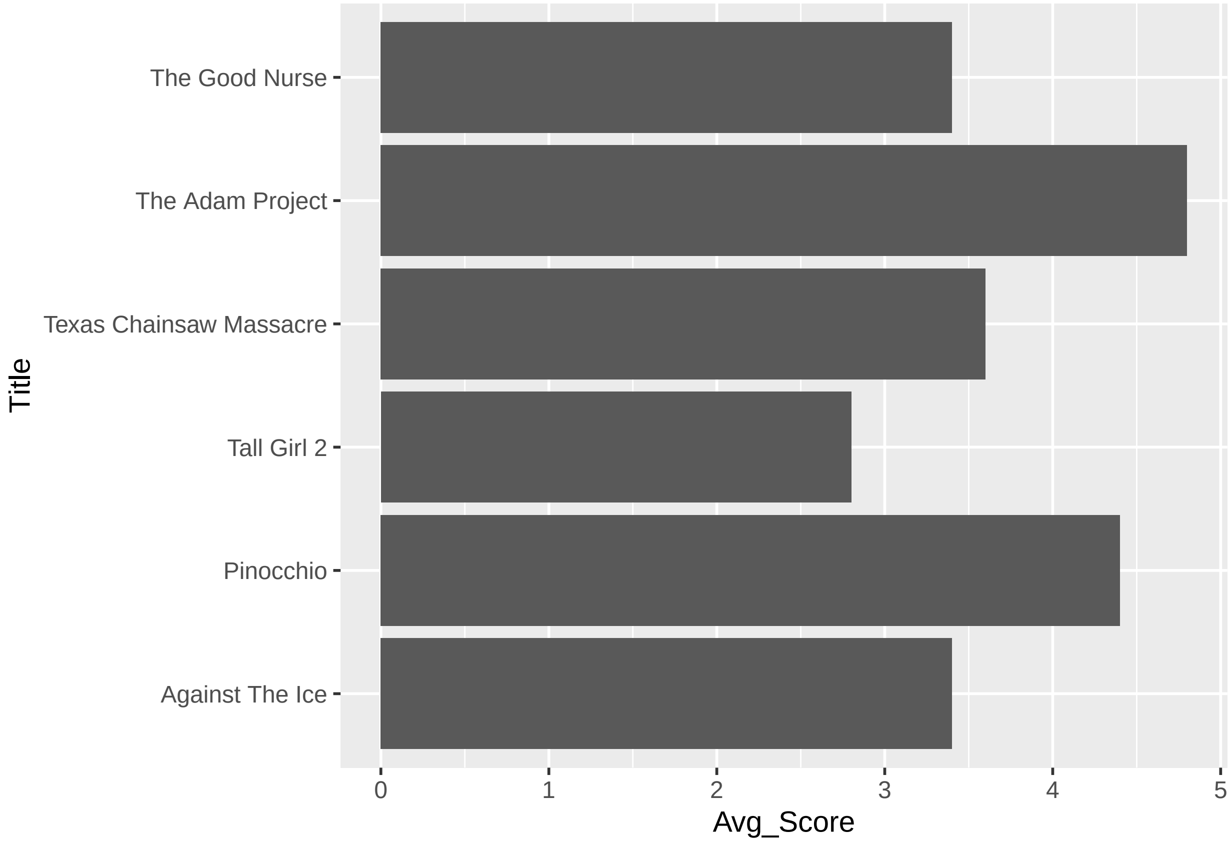
Here I see The Adam Project had the highest average rating with 4.8 and Tall Girl 2 has the lowest with 2.8.

```r
# I did a group by to see the average score for each movie rated.

new_data <- data %>%
  filter(!is.na(Rating)) %>%
   group_by(Title) %>%
    summarise(Avg_Score = mean(as.numeric(Rating))) %>%
      arrange(desc(Avg_Score))
        new_data
```

```
## # A tibble: 6 x 2
##   Title                   Avg_Score
##   <chr>                       <dbl>
## 1 The Adam Project              4.8
## 2 Pinocchio                     4.4
## 3 Texas Chainsaw Massacre       3.6
## 4 Against The Ice               3.4
## 5 The Good Nurse                3.4
## 6 Tall Girl 2                   2.8
```

Same results as the above.

```
# I did a ggplot of the Average Score for each movie for visualization purpose.

new_data %>%
ggplot +
geom_col(aes(Avg_Score, Title))
```



The Adam Project has the most 5 rating with 4 counts. Texas Mascare had 0 rating from a 7 year old who thought it was too bloody.

The Good Nurse got 1 - 2 Rating and 1 - 3 Rating and 3 - 4 Rating. That is about average rating.

The Adam Project got 1 - 3 Rating and 3 - 4 Rating. That is above average, 1st in place.

Texas Chainsaw Massacre got 1 - 0 Rating, 2 - 4 Rating, 2 - 5 Rating.

Tall Girl got 3 - 2 Rating and 2 - 4 Rating. That is below average. Ranked the lowest.

Pinocchio got 1 - 3 Rating, 1 - 4 Rating, and 3 - 5 Rating. This is above average, 2nd in place.

Against The Ice got 1 - 2 Rating, 1 - 3 Rating, and 3 - 5 Rating. This is average.

```
# I did another group by to see the count for each rating per movie.

count_data <- data
count_data %>% group_by(Title, Rating) %>% summarise(count = n())%>%
arrange(desc(Title))
```

```
## ‘summarise()‘ has grouped output by ’Title’. You can override using the
## ‘.groups‘ argument.
```

```
## # A tibble: 16 x 3
## # Groups:   Title [6]
##    Title                   Rating count
##    <chr>                    <int> <int>
##  1 The Good Nurse               2     1
##  2 The Good Nurse               3     1
##  3 The Good Nurse               4     3
##  4 The Adam Project             4     1
##  5 The Adam Project             5     4
##  6 Texas Chainsaw Massacre      0     1
##  7 Texas Chainsaw Massacre      4     2
##  8 Texas Chainsaw Massacre      5     2
##  9 Tall Girl 2                  2     3
## 10 Tall Girl 2                  4     2
## 11 Pinocchio                    3     1
## 12 Pinocchio                    4     1
## 13 Pinocchio                    5     3
## 14 Against The Ice              2     1
## 15 Against The Ice              3     1
## 16 Against The Ice              4     3
```

In conlusion more research has to be done on the ratings for these 5 movies to see what motivated these individuals to give them the rating they received. For now we can go as per the written reviews section for the reason of their ratings. For example Texas Chainsaw was bloody. Some gave it high ranking for that because the like horrow movies and some gave 0 because it was too bloody for her. We might want to look at the length of the movies. For example, The Good Nurse has a longest length of 121 minutes out of the 6 movies. Texas Chainsaw Massacre was the shortest with 81 minutes. The length of the movie could also affect the rating of the movies. Also we might want to look at the age of the individuals. The age could affect the rating of the movies. We would also need to take a take rating from a larger group to get a broader analysis.

For now The Adam Project ranks # 1 and Tall Girl ranks # 6 as per the 5 individuals.