DA5020.A6.Nithya.Sarabudla

nithyasarabudla

2023-10-21

Importing libraries

```
library(RSQLite)
library(ggplot2)
```

Bonus part 2

What is the average runtime for the thriller movie genre.

```
db_path <- "/Users/nithyasarabudla/Downloads/SQlite/imdb.db"  # Replace with the actual file path
con <- dbConnect(RSQLite::SQLite(), dbname = db_path)
query <- "SELECT AVG(Runtime)
              FROM movie_info
              WHERE Genre LIKE '%Thriller%'"

result <- dbGetQuery(con, query)
result
```

```
##   AVG(Runtime)
## 1     119.6423
```

Question 1

Part 1

Create a table named director_info using SQLite; the columns are: Director_ID, and Director_Name. The Director_ID should be the primary key.

```
CREATE TABLE director_info (
    Director_ID INTEGER PRIMARY KEY,
    Director_Name TEXT
)
```

Part 2

Import the entire data from the CSV file into the director_info table using the SQLite .import command (see helpful resources below). Verify that your data was imported correctly

```
.import "/Users/nithyasarabudla/Downloads/SQlite/directors.csv" director_info"
```

Question 2

Conneting to Database

```
db_path <- "/Users/nithyasarabudla/Downloads/SQlite/imdb.db"  # Replace with the actual file path
con <- dbConnect(RSQLite::SQLite(), dbname = db_path)
```

Part 1

Count the number of rows in the movie_info and director_info tables.

```
# Execute a SELECT query
result1 <- dbGetQuery(con, "SELECT count(*) as movie_count FROM movie_info")
result2 <- dbGetQuery(con, "SELECT count(*) as director_info FROM director_info")

# Print the result
print(result1)
```

```
##   movie_count
## 1        1000
```

```
print(result2)
```

```
##   director_info
## 1           548
```

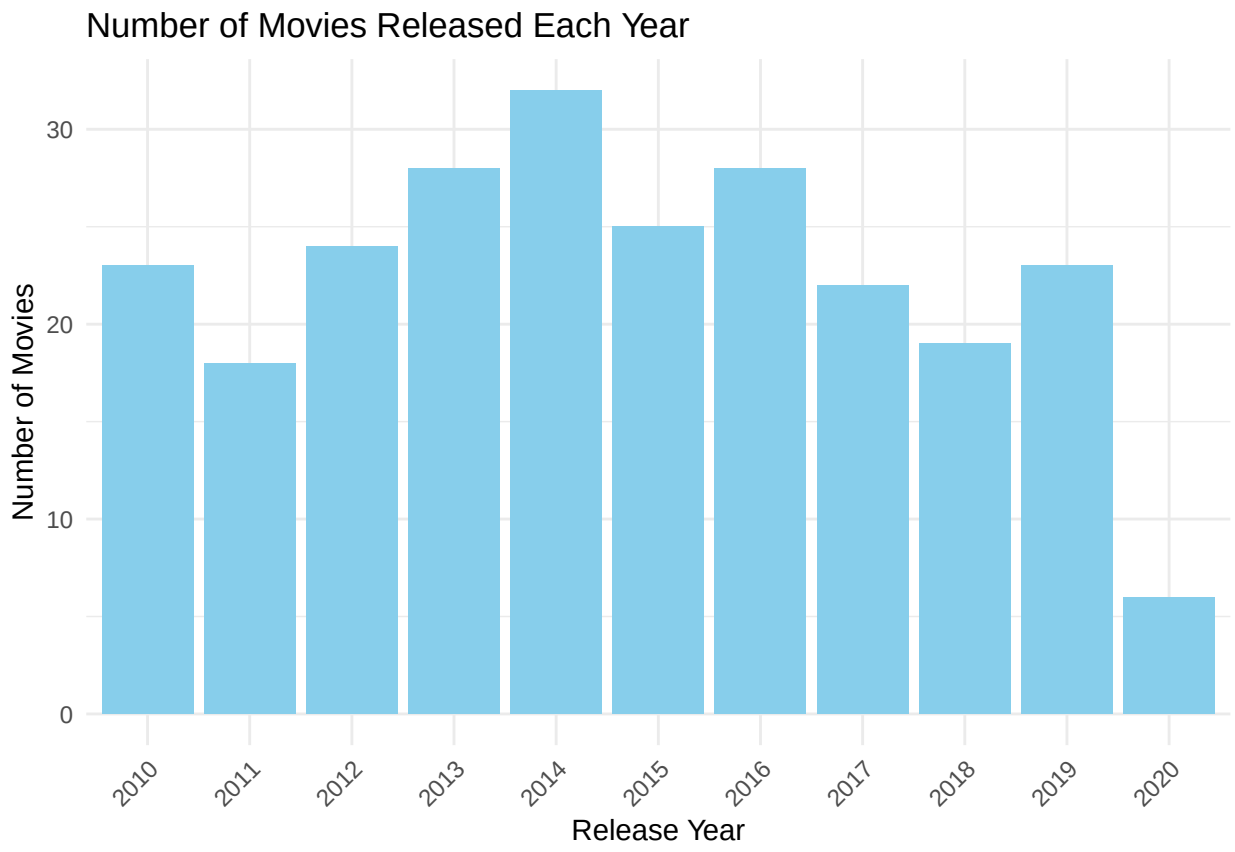
Part 2

How many movies were released between 2010 and 2020 (inclusive)? Visualize the results.

```
# Execute a SELECT query
query <- "SELECT Release_Year, COUNT(*) as num_movies FROM movie_info where Release_Year between 2010 A
result <- dbGetQuery(con, query)
result
```

```
##    Release_Year num_movies
## 1          2010         23
## 2          2011         18
## 3          2012         24
## 4          2013         28
## 5          2014         32
## 6          2015         25
## 7          2016         28
## 8          2017         22
## 9          2018         19
## 10         2019         23
## 11         2020          6
```

```r
ggplot(result, aes(x = Release_Year, y = num_movies)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Number of Movies Released Each Year", x = "Release Year", y = "Number of Movies") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Part 3

What is the minimum, average and maximum ratings for "Action" movies. Ensure that you query the genre using wild cards.

```r
min_rating <- dbGetQuery(con, "SELECT MIN(IMDB_Rating) as min_rating
FROM movie_info
WHERE upper(Genre) LIKE '%ACTION%'")

max_rating <- dbGetQuery(con, "Select max(IMDB_Rating) from movie_info where upper(Genre) LIKE '%ACTION

avg_rating <- dbGetQuery(con, "Select AVG(IMDB_Rating) from movie_info where upper(Genre) LIKE '%ACTION


print(min_rating)
```

```
##   min_rating
## 1        7.6
```

```
print(max_rating)
```

```
##   max(IMDB_Rating)
## 1                9
```

```
print(avg_rating)
```

```
##   AVG(IMDB_Rating)
## 1         7.948677
```

Part 4

What are the 25 highest-grossing movies within the dataset? Display the title, genre and gros

```
query <- "SELECT Series_Title, Genre, Gross
FROM movie_info
WHERE Gross IS NOT 'NA'
ORDER BY Gross DESC
LIMIT 25"

result <- dbGetQuery(con,query)

print(result)
```

```
##                                    Series_Title                         Genre
## 1     Star Wars: Episode VII - The Force Awakens    Action, Adventure, Sci-Fi
## 2                                Avengers: Endgame     Action, Adventure, Drama
## 3                                        Avatar   Action, Adventure, Fantasy
## 4                          Avengers: Infinity War    Action, Adventure, Sci-Fi
## 5                                       Titanic               Drama, Romance
## 6                                   The Avengers    Action, Adventure, Sci-Fi
## 7                                  Incredibles 2 Animation, Action, Adventure
## 8                                The Dark Knight         Action, Crime, Drama
## 9                                      Rogue One    Action, Adventure, Sci-Fi
## 10                         The Dark Knight Rises            Action, Adventure
## 11                     E.T. the Extra-Terrestrial               Family, Sci-Fi
## 12                                   Toy Story 4 Animation, Adventure, Comedy
## 13                                 The Lion King  Animation, Adventure, Drama
## 14                                   Toy Story 3 Animation, Adventure, Comedy
## 15                     Captain America: Civil War    Action, Adventure, Sci-Fi
## 16                                 Jurassic Park    Action, Adventure, Sci-Fi
## 17                 Guardians of the Galaxy Vol. 2    Action, Adventure, Comedy
## 18  Harry Potter and the Deathly Hallows: Part 2    Adventure, Drama, Fantasy
## 19                                  Finding Nemo Animation, Adventure, Comedy
## 20 The Lord of the Rings: The Return of the King     Action, Adventure, Drama
## 21                                     Deadpool    Action, Adventure, Comedy
## 22                                   Inside Out Animation, Adventure, Comedy
## 23         The Lord of the Rings: The Two Towers     Action, Adventure, Drama
## 24                                   Zootopia Animation, Adventure, Comedy
## 25                                      Joker       Crime, Drama, Thriller
##        Gross
```

```
## 1  936662225
## 2  858373000
## 3  760507625
## 4  678815482
## 5  659325379
## 6  623279547
## 7  608581744
## 8  534858444
## 9  532177324
## 10 448139099
## 11 435110554
## 12 434038008
## 13 422783777
## 14 415004880
## 15 408084349
## 16 402453882
## 17 389813101
## 18 381011219
## 19 380843261
## 20 377845905
## 21 363070709
## 22 356461711
## 23 342551365
## 24 341268248
## 25 335451311
```

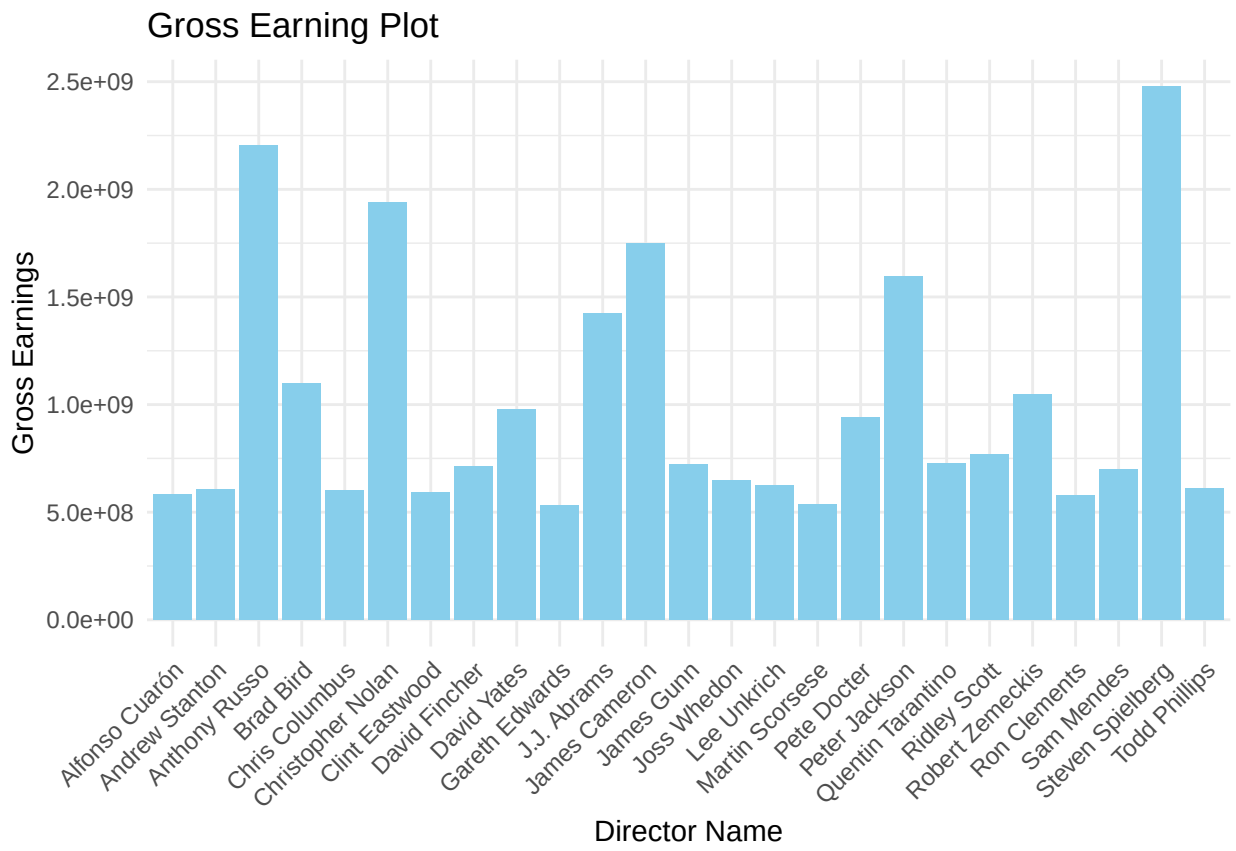
Part 5

Which directors have the highest-grossing movies. Display the director name and the total gross. Ensure that you join the necessary tables. Visualize the results using a Bar chart

```
query <- "SELECT d.Director_Name, SUM(m.Gross) as Total_Gross
FROM movie_info m
JOIN director_info d ON m.Director_ID = d.Director_ID
GROUP BY d.Director_Name ORDER BY Total_Gross DESC
LIMIT 25 ";

result <- dbGetQuery(con,query)

ggplot(result, aes(x = Director_Name, y = Total_Gross)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Gross Earning Plot", x = "Director Name", y = "Gross Earnings") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
print(result)
```

```
##        Director_Name Total_Gross
## 1   Steven Spielberg  2478133165
## 2      Anthony Russo  2205039403
## 3  Christopher Nolan  1937454106
## 4      James Cameron  1748236602
## 5      Peter Jackson  1597312443
## 6        J.J. Abrams  1423170905
## 7          Brad Bird  1099627795
## 8    Robert Zemeckis  1049446456
## 9        David Yates   978953721
## 10       Pete Docter   939382131
## 11      Ridley Scott   766711423
## 12 Quentin Tarantino   727034316
## 13        James Gunn   722989701
## 14     David Fincher   713314479
## 15        Sam Mendes   698139284
## 16       Joss Whedon   648794064
## 17       Lee Unkrich   624730895
## 18     Todd Phillips   612773814
## 19    Andrew Stanton   604651425
## 20    Chris Columbus   603336793
## 21    Clint Eastwood   592692644
## 22    Alfonso Cuarón   582645455
```

```
## 23      Ron Clements   577650742
## 24   Martin Scorsese   538319198
## 25    Gareth Edwards   532177324
```

Part 6

Create a function called verifyDirector() that takes a director name as its argument, and queries the database to check if the director exists. Your function should display a message to notify the user if the director was found or not.

```
verifyDir <- function(director_name) {

  # Prepare the SQL query to check if the director exists
  query <- sprintf("SELECT COUNT(*) as count FROM director_info WHERE Director_Name = '%s'", director_na

  # Execute the query
  result <- dbGetQuery(con, query)


  # Check if the director exists (count > 0)
  if (result$count > 0) {
    print("DIRECTOR FOUND IN DATABSE !")
  } else {
    print("DIRECTOR NOT FOUND IN DATABSE !")
  }
}

# Director present - check
verifyDir("Anthony Russo")
```

```
## [1] "DIRECTOR FOUND IN DATABSE !"
```

```
#Director not present - check
verifyDir("Nithya S")
```

```
## [1] "DIRECTOR NOT FOUND IN DATABSE !"
```