Hyperiondev

TASK

Exploratory Data Analysis on the Movies Data Set

Visit our website

Introduction

This analysis looks at a data set called “movies”. Below are the steps taken for data wrangling and exploring the data. During the exploration process the use of visual representations will aid in answering the questions given:

Identify relationships between variables (features):

- Which are the 5 most expensive movies?
- What are the top 5 most profitable movies?
- Find the most popular movies.
- Find movies which are rated above 7.
- Most successful genres.
- Genre popularity over time
- Top 15 countries by the popularity of movies
- Popular movies over time

DATA CLEANING

Imported the necessary libraries to clean, manipulate, and plot the data.

```
# Import libraries

import numpy as np
import pandas as pd
import seaborn as sns
import ast, json

from datetime import datetime
import matplotlib.pyplot as plt
%matplotlib inline
```

```
# Load the movies dataset
movies_df = pd.read_csv('movies.csv')

movies_df.shape
```

```
(4803, 20)
```

Identify columns that are redundant or unnecessary. It is always easier to make your decisions based on data which is relevant and concise. Remove the following columns `['homepage', 'keywords', 'original_language', 'original_title', 'overview', 'production_companies', 'status', 'tagline']` from the data set as they will not be used in the analysis.

```
# Drop unnecessary columns
movies_df.drop(['homepage', 'keywords', 'original_language',
                'original_title', 'overview', 'production_companies',
                'status', 'tagline'], axis=1, inplace=True)
movies_df.head()
```

	budget	genres	id	popularity	production_countries	release_date	revenue	runtime	spoken_languages	title	vote_average	vote_count
0	237000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	19995	150.437577	[{"iso_3166_1": "US", "name": "United States o...	2009-12-10	2787965087	162.0	[{"iso_639_1": "en", "name": "English"}, {"iso...	Avatar	7.2	11800
1	300000000	[{"id": 12, "name": "Adventure"}, {"id": 14, "...	285	139.082615	[{"iso_3166_1": "US", "name": "United States o...	2007-05-19	961000000	169.0	[{"iso_639_1": "en", "name": "English"}]	Pirates of the Caribbean: At World's End	6.9	4500
2	245000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	206647	107.376788	[{"iso_3166_1": "GB", "name": "United Kingdom"...	2015-10-26	880674609	148.0	[{"iso_639_1": "fr", "name": "Fran\u00e7ais"},...	Spectre	6.3	4466
3	250000000	[{"id": 28, "name": "Action"}, {"id": 80, "nam...	49026	112.312950	[{"iso_3166_1": "US", "name": "United States o...	2012-07-16	1084939099	165.0	[{"iso_639_1": "en", "name": "English"}]	The Dark Knight Rises	7.6	9106
4	260000000	[{"id": 28, "name": "Action"}, {"id": 12, "nam...	49529	43.926995	[{"iso_3166_1": "US", "name": "United States o...	2012-03-07	284139100	132.0	[{"iso_639_1": "en", "name": "English"}]	John Carter	6.1	2124

On checking the dataset, we see that `genres`, `production_countries`, `spoken_languages` are in the JSON format which will make it difficult to manipulate the dataframe. The `parse_col_json function` flattens these columns into a format that can be easily interpreted.

```
# Convert from JSON format to a list of strings
def parse_col_json(column, key):
    """
    Args:
        column: string
            name of the column to be processed.
        key: string
            name of the dictionary key which needs to be extracted
    """
    for index,i in zip(movies_df.index, movies_df[column].apply(json.loads)):
        list1=[]
        for j in range(len(i)):
            list1.append((i[j][key]))# the key 'name' contains the name of the genre
        movies_df.loc[index, column] = str(list1)

# Flatten data in columns
parse_col_json('genres', 'name')
parse_col_json('production_countries', 'name')
parse_col_json('spoken_languages', 'name')
```

```
# Inspect flattened data
movies_df.sample(1)
```

	budget	genres	id	popularity	production_countries	release_date	revenue	runtime	spoken_languages	title	vote_average	vote_count	release_year
2243	20000000	['Drama', 'Music']	3902	13.032308	['United States of America']	2007-10-01	4001121	135.0	['English']	I'm Not There.	6.6	195	2007

MISSING DATA

Checking which columns have missing values.

```
print('\nColumns with missing values: ')
print(movies_df.isnull().any())
```

```
Columns with missing values:
budget                  False
genres                  False
id                      False
popularity              False
production_countries    False
release_date             True
revenue                 False
runtime                  True
spoken_languages        False
title                   False
vote_average            False
vote_count              False
dtype: bool
```

Checking how much missing values there are in the columns and if it will have a huge impact to remove them.

```
print('\nColumns with sum of missing values: ')
print(movies_df.isnull().sum())
```

```
Columns with sum of missing values:
budget                  0
genres                  0
id                      0
popularity              0
production_countries    0
release_date            1
revenue                 0
runtime                 2
spoken_languages        0
title                   0
vote_average            0
vote_count              0
dtype: int64
```

Dropping the rows with missing data for the 2 columns that were identified.

```
movies_df.dropna(subset=["release_date", "runtime"], inplace=True)
print('\nColumns with sum of missing values: ')
print(movies_df.isnull().sum())
```

```
Columns with sum of missing values:
budget                  0
genres                  0
id                      0
popularity              0
production_countries    0
release_date            0
revenue                 0
runtime                 0
spoken_languages        0
title                   0
vote_average            0
vote_count              0
dtype: int64
```

Remove any duplicate rows.

```
# Remove duplicate rows
movies_df.drop_duplicates(inplace=True)
movies_df.shape
```

```
(4800, 12)
```

REMOVE ROWS WITH MISSING DATA

Some movies in the database have zero budget or zero revenue which implies that their values have not been recorded or some information is missing. Discard such entries from the dataframe.

```
# Remove rows with missing data
movies_df.drop(movies_df.index[(movies_df['budget'] == 0)], axis=0, inplace=True)
movies_df.drop(movies_df.index[(movies_df['revenue'] == 0)], axis=0, inplace=True)

# checking shape after rows with zero budget and zero revenue were removed
movies_df.shape
```

```
(3229, 12)
```

CHANGE DATA TYPES

To manipulate the columns easily, it is important that we make use of the python objects. Change the release date column into `DateTime` format and extract the year from the date. This will help us in analysing yearly data.

```
# Change the release_date column to DateTime
movies_df['release_date'] = pd.to_datetime(movies_df['release_date'])

# Extract the release year from every release date
movies_df['release_year'] = movies_df['release_date'].dt.year

print(movies_df.query('release_year != 0').groupby(['release_year'])['release_year'].count())
```

```
release_year
1916      1
1925      1
1927      1
1929      1
1930      1
       ...
2012    137
2013    152
2014    135
2015    121
2016     72
Name: release_year, Length: 89, dtype: int64
```

Change budget and revenue columns to a integer data type using numpy's `int64` method.

```
# determining what the data types are for the data that was imported
movies_df.dtypes
```

```
budget                          int64
genres                         object
id                              int64
popularity                     float64
production_countries            object
release_date            datetime64[ns]
revenue                         int64
runtime                        float64
spoken_languages                object
title                           object
vote_average                   float64
vote_count                      int64
release_year                    int32
dtype: object
```

```
movies_df['budget'] = movies_df['budget'].astype(np.int64)
movies_df['revenue'] = movies_df['revenue'].astype(np.int64)
movies_df['release_year'] = movies_df['release_year'].astype(np.int64)

print(movies_df['budget'].dtype)
print(movies_df['revenue'].dtype)
print(movies_df['release_year'].dtype)
```

```
int64
int64
int64
```

```
# checking if all the data types are now imported correctly
movies_df.dtypes
```

```
budget                          int64
genres                         object
id                              int64
popularity                     float64
production_countries            object
release_date            datetime64[ns]
revenue                         int64
runtime                        float64
spoken_languages                object
title                           object
vote_average                   float64
vote_count                      int64
release_year                    int64
dtype: object
```

DATA STORIES AND VISUALISATIONS

- Which are the 5 most expensive movies?

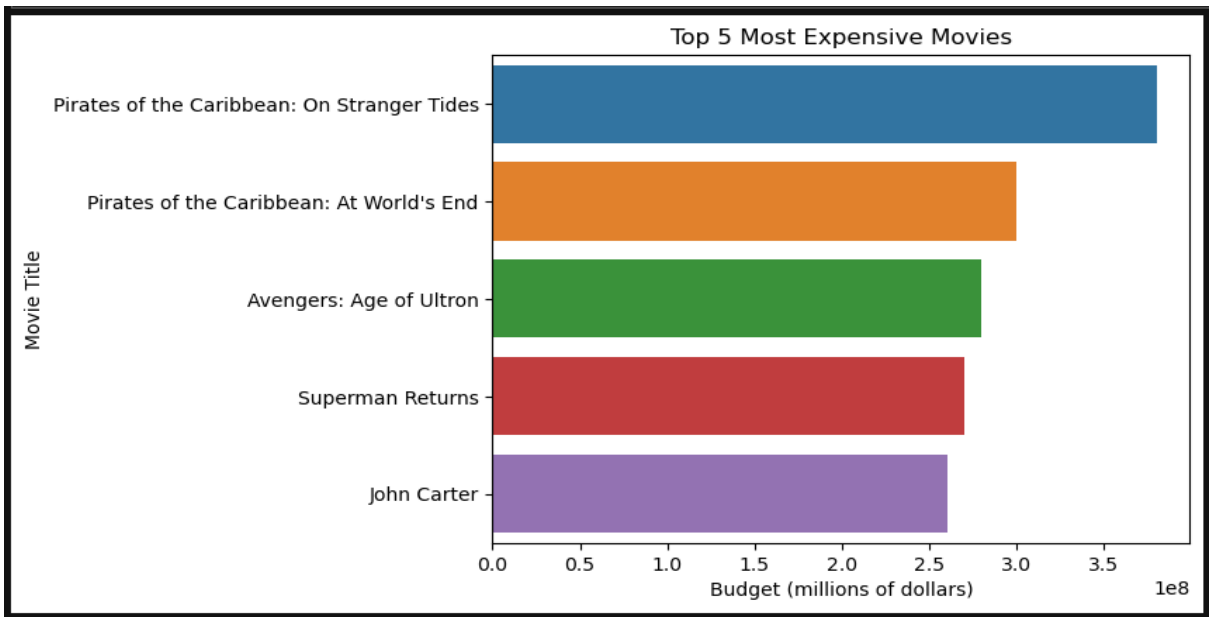
How do the most expensive and cheapest movies compare? Exploring the most expensive movies highlights if some movies are worth the money spent on them based on their performance and revenue generated.

Firstly, we create a temporary data set based on certain parameters (revenue and budget not equal to 0). Thereafter, we select the relevant columns for analysis and visualization purposes.

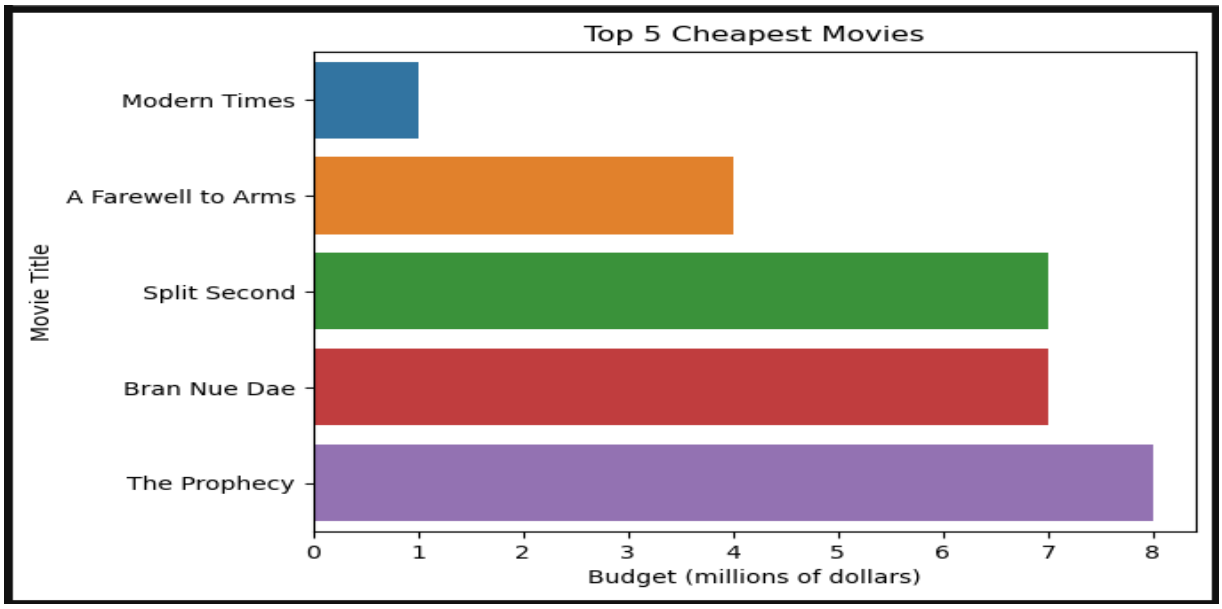
Columns: 'budget', 'revenue', 'title'

Now that we have created the data base with the relevant columns for analysis. Let us explore and make comparison between the 5 most expensive movies vs the 5 cheapest movies.

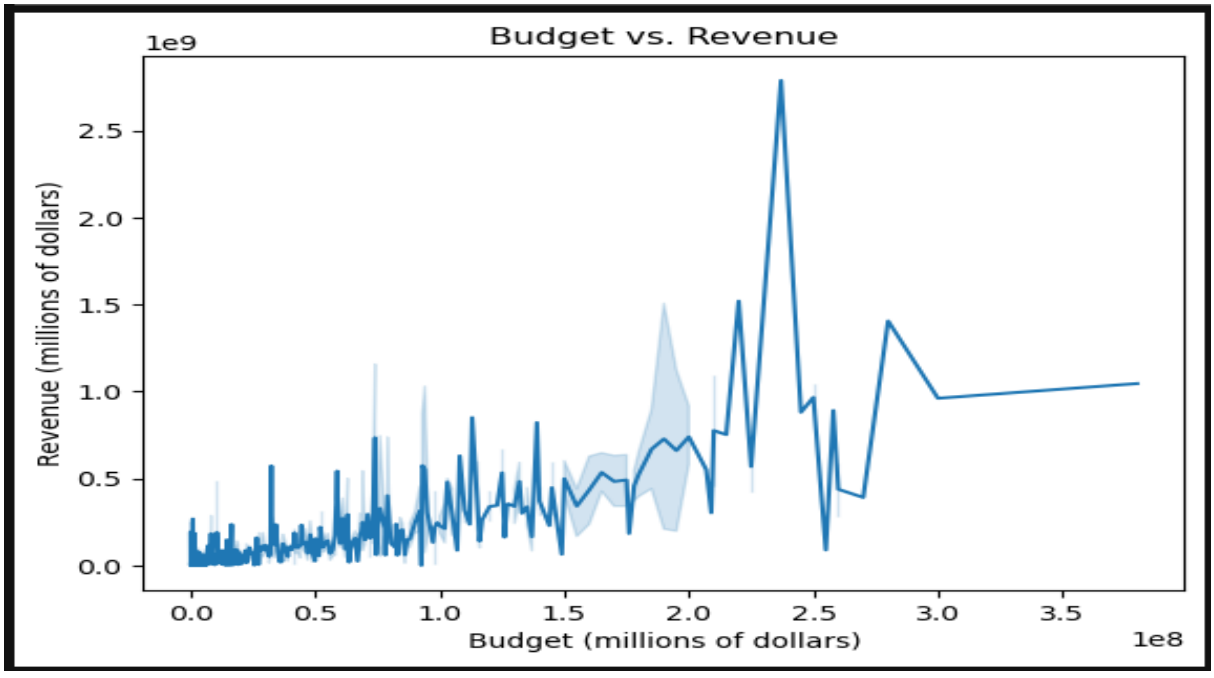
Top 5 Most Expensive Movies



Top 5 Cheapest Movies



Let's now compare the 2 by their budget vs the revenue.



** Conclusion **

The graph shows that there is no clear correlation between budget and revenue. Some of the most expensive movies have been box office failures, while some of the cheapest movies have been huge successes.

- What are the top 5 most profitable movies?

Compare the min and max profits. The comparison helps us identify the different approaches which failed and succeeded. Subtracting the budget from the revenue generated, will return the profit earned.

We can start by first creating a temporary data set based on certain parameters (revenue and budget not equal to 0). Thereafter, we select the relevant columns for analysis and visualization purposes.

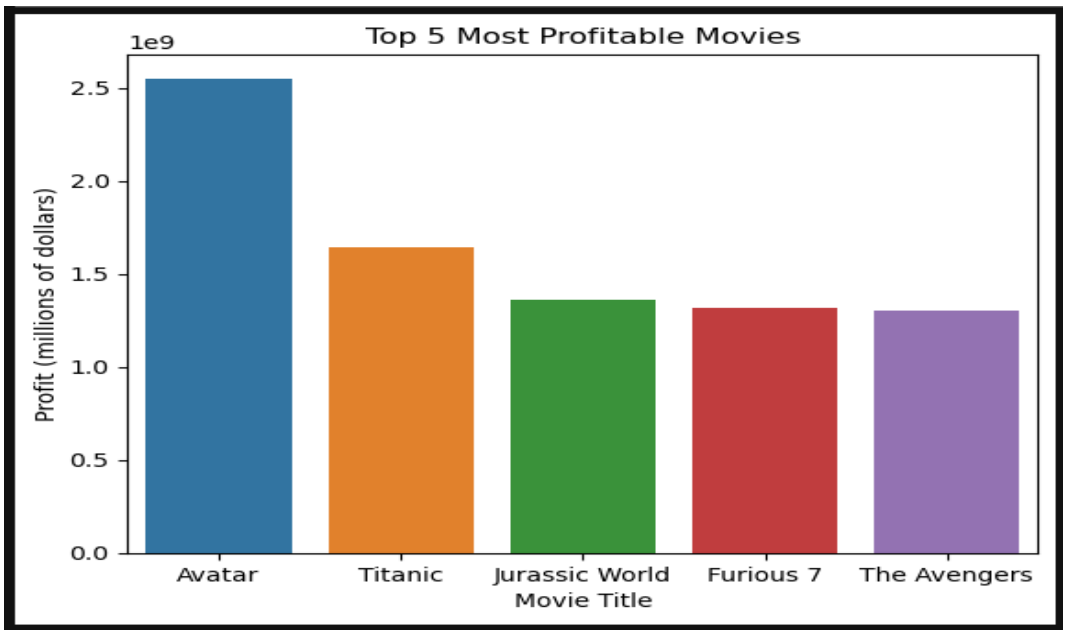
Columns: 'budget', 'revenue', 'title'

Next, we create a new column called 'profit'; this will hold the data from the calculation we need to do to obtain this (i.e., revenue – budget = profit).

Lastly, we get the top 5 most profitable movies based on their profit margin.

	budget	revenue	title	profit
0	237000000	2787965087	Avatar	2550965087
25	200000000	1845034188	Titanic	1645034188
28	150000000	1513528810	Jurassic World	1363528810
44	190000000	1506249360	Furious 7	1316249360
16	220000000	1519557910	The Avengers	1299557910

Top 5 Most Profitable Movies



Now to get the minimum and maximum profits. We now need to calculate the difference between the two and give our analysis on this.

```
# Get the min profits
min_profit = movies_profit['profit'].min()

print("Min profit:", min_profit)
```

```
Min profit: -165710090
```

```
# Get the max profits
max_profit = movies_profit['profit'].max()

print("Max profit:", max_profit)
```

```
Max profit: 2550965087
```

```
# Calculate the difference between the min and max profits
profit_difference = max_profit - min_profit
# Print the result
print("Profit difference:", profit_difference)
```

```
Profit difference: 2716675177
```

The top 5 most profitable movies of all time are:

1. Avatar (2009) - $2,550,965,087

2. Titanic (1997) - $1,645,034,188

3. Jurassic World (2015) - $1,363,528,810

4. Furious 7 (2015) - $1,316,249,360

5. The Avengers (2012) - $1,299,557,910

** Conclusion **

The difference between the highest and lowest profits is $2,716,675,177. This shows that there is a significant difference in the profitability of different movies. The highest-grossing movies can generate significantly more revenue than the lowest-grossing movies.

There are several factors that can contribute to a movie's profitability. These include the movie's budget, the marketing campaign, the release date, and the critical and audience reception. Movies with large budgets and extensive marketing campaigns are more likely to be profitable than movies with smaller budgets and less promotion. Movies that are released during the summer months are also more likely to be profitable than movies that are released during other times of the year. Finally, movies that receive positive reviews from critics and audiences are more likely to be profitable than movies that receive negative reviews.

The top 5 most profitable movies of all time all had several factors that contributed to their success. These movies had large budgets, extensive marketing campaigns, and were released during the summer months. They also received positive reviews from critics and audiences. These factors all helped to make these movies extremely profitable.

- Find the most popular movies.

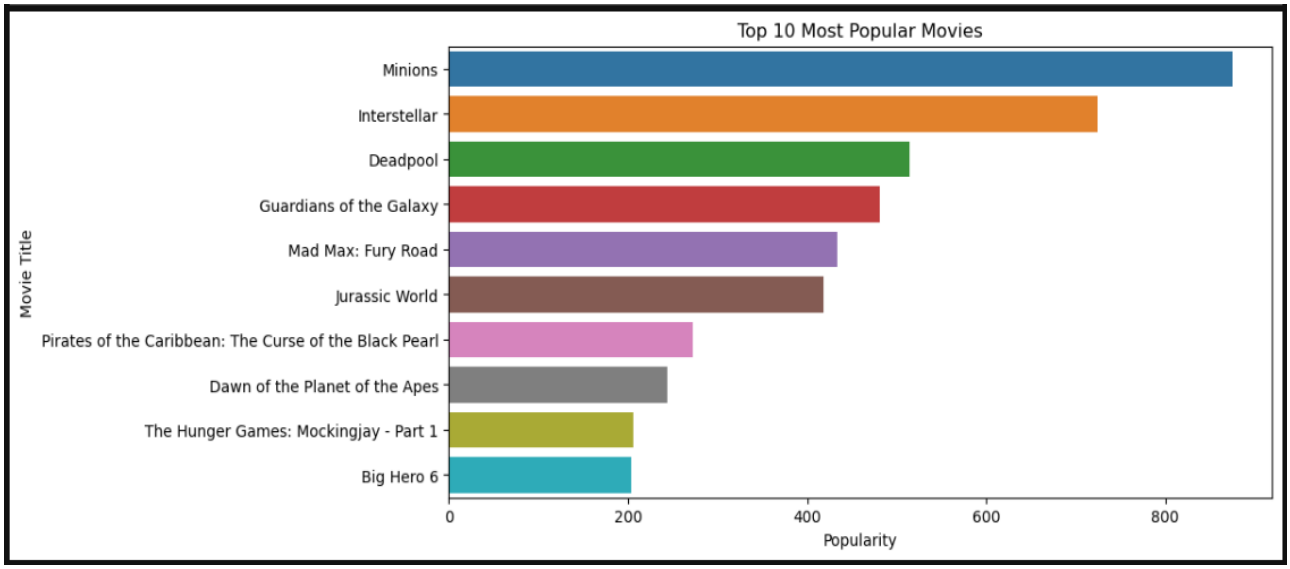
Sorting the dataframe based on the popularity column.

We can start by first creating a temporary data set based on certain parameters (popularity not equal to 0). Thereafter, we select the relevant columns for analysis and visualization purposes.

Columns: 'title', 'popularity'

Next, we sort the data set by the popularity column in ascending order, so that the top 10 observations with the highest popularity score appears.

	title	popularity
546	Minions	875.581305
95	Interstellar	724.247784
788	Deadpool	514.569956
94	Guardians of the Galaxy	481.098624
127	Mad Max: Fury Road	434.278564
28	Jurassic World	418.708552
199	Pirates of the Caribbean: The Curse of the Bla...	271.972889
82	Dawn of the Planet of the Apes	243.791743
200	The Hunger Games: Mockingjay - Part 1	206.227151
88	Big Hero 6	203.734590



** Conclusion **

There are several reasons why these movies were so popular:

Large budgets: These movies all had large budgets, which allowed them to be marketed extensively and to hire top talent.

Extensive marketing campaigns: These movies all had extensive marketing campaigns, which helped to generate awareness and excitement among potential viewers.

Summer releases: These movies were all released during the summer months, when movie attendance is typically higher.

Positive reviews: These movies all received positive reviews from critics and audiences, which helped to generate buzz and encourage people to see them.

Cultural zeitgeist: These movies all tapped into the cultural zeitgeist in some way, which helped to make them even more popular. For example, Minions was released at a time when there was a lot of interest in animated films, and Interstellar was released at a time when there was a lot of interest in space exploration.

- Find movies which are rated above 7.

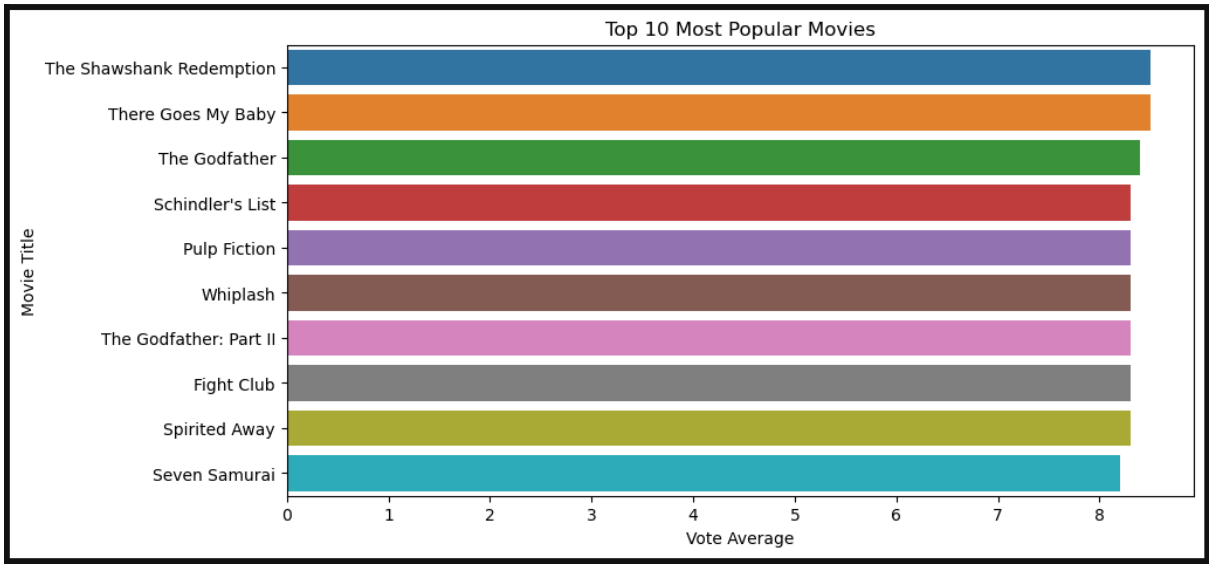
We need to create a temporary data set based on the 'vote_average' column. We add in the parameter that the average total should be higher than 7.

Next, we select the relevant columns that we need for analysing and visualization.

Columns: 'title', 'vote_average'

Lastly, we sort the data set by the vote_average column in ascending order, so that the top 10 observations with the highest vote_average score appears on top.

	title	vote_average
1881	The Shawshank Redemption	8.5
2970	There Goes My Baby	8.5
3337	The Godfather	8.4
1818	Schindler's List	8.3
3232	Pulp Fiction	8.3
3865	Whiplash	8.3
2731	The Godfather: Part II	8.3
662	Fight Club	8.3
2294	Spirited Away	8.3
4535	Seven Samurai	8.2



** Conclusion **

The following movies were rated 7 and above because they were well-received by critics and audiences alike. They were praised for their strong performances, compelling stories, and beautiful cinematography. Additionally, these movies had a lot of buzz surrounding them, which helped to generate interest and excitement among moviegoers.

Overall, these movies were rated 7 and above because they were well-made and entertaining. They had strong performances, compelling stories, beautiful cinematography, and a lot of buzz surrounding them.

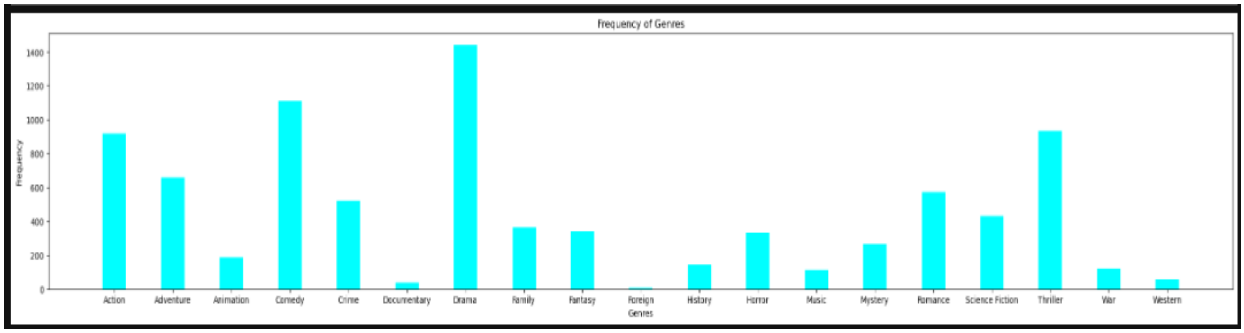
- Most successful genres.

Create a bar plot explaining the frequency of movies in each genre.

We have created a new temporary data set, based on popularity, as this column has numeric values to which we can set parameters against.

Thereafter, we select our relevant columns for analysing and visualization.

Columns: 'title', 'genres'



** Conclusion **

Action: 918
Adventure: 661
Animation: 188
Comedy: 1110
Crime: 521
Documentary: 38
Drama: 1441
Family: 365
Fantasy: 342

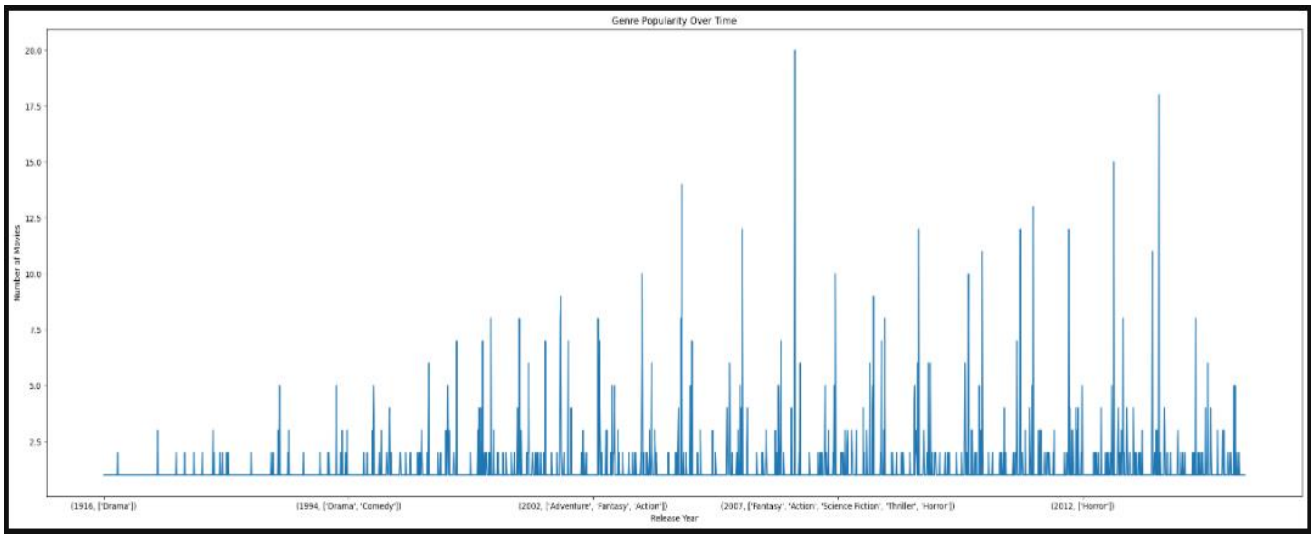
Foreign: 5
History: 145
Horror: 332
Music: 111
Mystery: 265
Romance: 574
Science Fiction: 431
Thriller: 935
War: 120
Western: 57

Based on the given genre frequencies, it can be concluded that action, comedy, and drama movies are the most popular genres among viewers.
Action movies provide fast-paced and exciting content, comedy movies offer an escape from daily life with humour, and drama movies explore a wide range of human emotions.
These genres have a broad appeal and can attract large audiences.
Other genres such as thrillers, romance, science fiction, horror, crime, adventure, and family movies also have their own dedicated fan bases and contribute to the diversity of the film industry.
The popularity of each genre can be attributed to the unique elements and experiences they offer to viewers.

- Genre popularity over time.

Create a plot explaining the genre popularity over time.

We have created a new temporary data set, grouped by the release year and the genres. Which we will be using to count, analyse, and visualize.



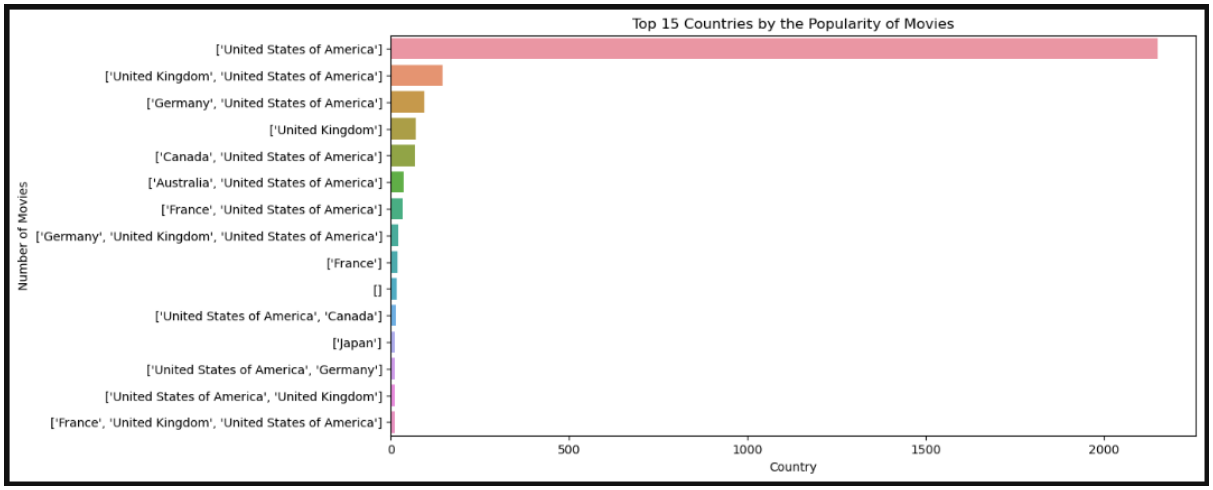
** Conclusion **

This visualization shows the popularity of different movie genres over time. We can see that action movies have been the most popular genre since the 1980s, followed by comedy and drama. Horror movies have also been consistently popular, while westerns have declined in popularity.

- Top 15 countries by the popularity of movies.

Create a plot explaining the top 15 countries by the popularity of movies.

We get the top 15 countries by counting the number of movies in the 'production_countries' column per country it was released.



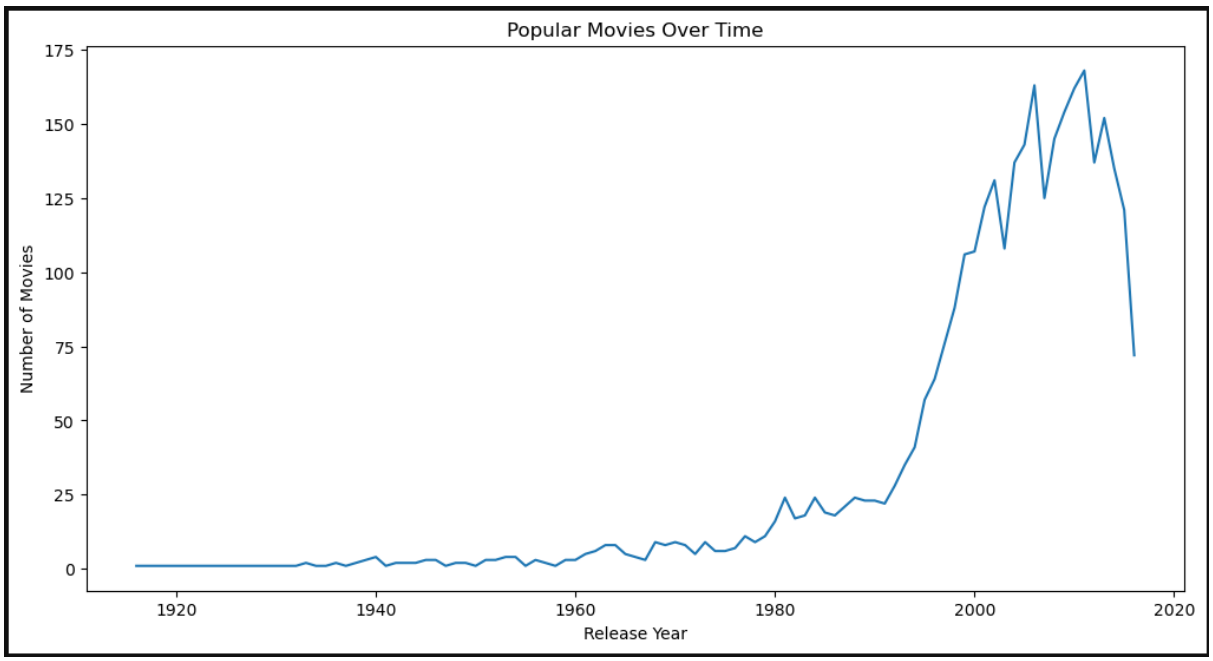
** Conclusion **

The visualization shows the top 15 countries by the number of movies. The United States is the most popular country for movies, followed by India, the United Kingdom, France, and China. This is likely because these countries have large film industries and produce a lot of movies each year.

- Popular movies over time.

Create a plot explaining the popular movies over time.

We get the number of movies released each year by extracting this from the 'released_year' column.



** Conclusion **

The visualization shows the number of movies released each year. The number of movies released has been increasing steadily over time, with a slight dip in the early 2000s. This is likely because the film industry has been growing and more movies are being made each year.

THIS REPORT WAS WRITTEN BY: FARINAAZ SLAMANG