IMDB Movies Project

PROJECT SETUP AND DATA LOADING

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

```
df = pd.read_csv("imdb_movies.csv")
df
```

	names	date_x	score	genre	overview	crew	orig_title	status	orig_lang	budget_x	revenu
0	Creed III	03/02/2023	73.0	Drama, Action	After dominating the boxing world, Adonis Cree...	Michael B. Jordan, Adonis Creed, Tessa Thompso...	Creed III	Released	English	75000000.0	2.716167e+0
1	Avatar: The Way of Water	12/15/2022	78.0	Science Fiction, Adventure, Action	Set more than a decade after the events of the...	Sam Worthington, Jake Sully, Zoe Saldaña, Neyt...	Avatar: The Way of Water	Released	English	460000000.0	2.316795e+0
2	The Super Mario Bros. Movie	04/05/2023	76.0	Animation, Adventure, Family, Fantasy, Comedy	While working underground to fix a water main,...	Chris Pratt, Mario (voice), Anya Taylor-Joy, P...	The Super Mario Bros. Movie	Released	English	100000000.0	7.244590e+0
3	Mummies	01/05/2023	70.0	Animation, Comedy, Family, Adventure, Fantasy	Through a series of unfortunate events, three ...	Óscar Barberán, Thut (voice), Ana Esther Albor...	Momias	Released	Spanish, Castilian	12300000.0	3.420000e+0
4	Supercell	03/17/2023	61.0	Action	Good-hearted teenager William always lived in ...	Skeet Ulrich, Roy Cameron, Anne Heche, Dr Quin...	Supercell	Released	English	77000000.0	3.409420e+0
...	...	...	...	...	...	...	...	...	...	...	
10173	20th Century Women	12/28/2016	73.0	Drama	In 1979 Santa Barbara, California, Dorothea Fi...	Annette Bening, Dorothea Fields, Lucas Jade Zu...	20th Century Women	Released	English	7000000.0	9.353729e+0

10177	The Swan Princess: A Royal Wedding	07/20/2020	70.0	Animation, Family, Fantasy	Princess Odette and Prince Derek are going to ...	Nina Herzog, Princess Odette (voice), Yuri Low...	The Swan Princess: A Royal Wedding	Released	English	92400000.0	5.394018e+0

10178 rows × 12 columns

DATA OVERVIEW AND BASIC EXPLORATION

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10178 entries, 0 to 10177
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   names       10178 non-null  object
 1   date_x      10178 non-null  object
 2   score       10178 non-null  float64
 3   genre       10093 non-null  object
 4   overview    10178 non-null  object
 5   crew        10122 non-null  object
 6   orig_title  10178 non-null  object
 7   status      10178 non-null  object
 8   orig_lang   10178 non-null  object
 9   budget_x    10178 non-null  float64
 10  revenue     10178 non-null  float64
 11  country     10178 non-null  object
dtypes: float64(3), object(9)
memory usage: 954.3+ KB
```

```
# Display the descriptive statistics
df.describe()
```

	date_x	score	budget_x	revenue
count	10178	10178.000000	1.017800e+04	1.017800e+04
mean	2008-06-15 06:16:37.445470720	63.497052	6.488238e+07	2.531401e+08
min	1903-05-15 00:00:00	0.000000	1.000000e+00	0.000000e+00
25%	2001-12-25 06:00:00	59.000000	1.500000e+07	2.858898e+07
50%	2013-05-09 00:00:00	65.000000	5.000000e+07	1.529349e+08
75%	2019-10-17 00:00:00	71.000000	1.050000e+08	4.178021e+08
max	2023-12-31 00:00:00	100.000000	4.600000e+08	2.923706e+09
std	NaN	13.537012	5.707565e+07	2.777880e+08

DATA CLEANING

```
# Convert the date_x column from object to datetime
df["date_x"] = pd.to_datetime(df["date_x"])
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10178 entries, 0 to 10177
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   names       10178 non-null  object
 1   date_x      10178 non-null  datetime64[ns]
 2   score       10178 non-null  float64
 3   genre       10178 non-null  object
 4   overview    10178 non-null  object
 5   crew        10178 non-null  object
 6   orig_title  10178 non-null  object
 7   status      10178 non-null  object
 8   orig_lang   10178 non-null  object
 9   budget_x    10178 non-null  float64
 10  revenue     10178 non-null  float64
 11  country     10178 non-null  object
dtypes: datetime64[ns](1), float64(3), object(8)
memory usage: 954.3+ KB
```

```
# Fill the null values in 'genre' and 'crew' column with 'Unavailable'
df["genre"] = df["genre"].fillna("Unavailable")
df["crew"] = df["crew"].fillna("Unavailable")
```

```
#total null values in the dataset
df.isnull().sum()
```

```
names         0
date_x        0
score         0
genre         0
overview      0
crew          0
orig_title    0
status        0
orig_lang     0
budget_x      0
revenue       0
country       0
dtype: int64
```

UNIVARIATE ANALYSIS

1.What is the distribution of movies by years?Plot a histogram and describe its shape.

```
# Adding a YEAR Column
df["year"] = df["date_x"].dt.strftime("%Y")
```

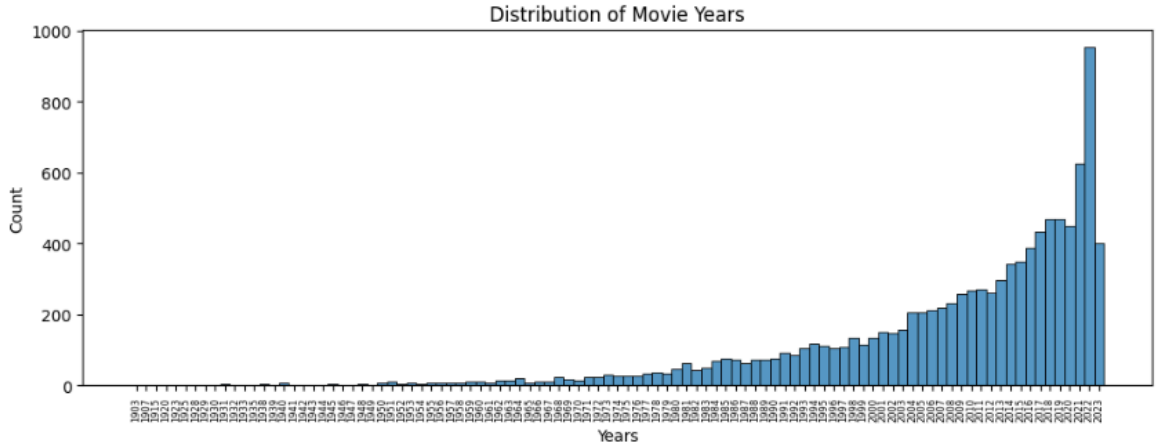
```
df
```

	names	date_x	score	genre	overview	crew	orig_title	status	orig_lang	budget_x	revenue	c
0	Creed III	2023-03-02	73.0	Drama, Action	After dominating the boxing world, Adonis Cree...	Michael B. Jordan, Adonis Creed, Tessa Thompso...	Creed III	Released	English	75000000.0	2.716167e+08	
1	Avatar: The Way of Water	2022-12-15	78.0	Science Fiction, Adventure, Action	Set more than a decade after the events of the...	Sam Worthington, Jake Sully, Zoe Saldaña, Neyt...	Avatar: The Way of Water	Released	English	460000000.0	2.316795e+09	
2	The Super Mario Bros. Movie	2023-04-05	76.0	Animation, Adventure, Family, Fantasy, Comedy	While working underground to fix a water main,...	Chris Pratt, Mario (voice), Anya Taylor-Joy, P...	The Super Mario Bros. Movie	Released	English	100000000.0	7.244590e+08	
3	Mummies	2023-01-05	70.0	Animation, Comedy, Family, Adventure, Fantasy	Through a series of unfortunate events, three ...	Óscar Barberán, Thut (voice), Ana Esther Albor...	Momias	Released	Spanish, Castilian	12300000.0	3.420000e+07	
4	Supercell	2023-03-17	61.0	Action	Good-hearted teenager William always lived in ...	Skeet Ulrich, Roy Cameron, Anne Heche, Dr Quin...	Supercell	Released	English	77000000.0	3.409420e+08	
...	...	...	...	...	...	...	...	...	...	...	...	
10173	20th Century Women	2016-12-28	73.0	Drama	In 1979 Santa Barbara, California, Dorothea Fi...	Annette Bening, Dorothea Fields, Lucas Jade Zu...	20th Century Women	Released	English	7000000.0	9.353729e+06	

10177	The Swan Princess: A Royal Wedding	2020-07-20	70.0	Animation, Family, Fantasy	Princess Odette and Prince Derek are going to ...	Nina Herzog, Princess Odette (voice), Yuri Low...	The Swan Princess: A Royal Wedding	Released	English	92400000.0	5.394018e+08

10178 rows × 13 columns

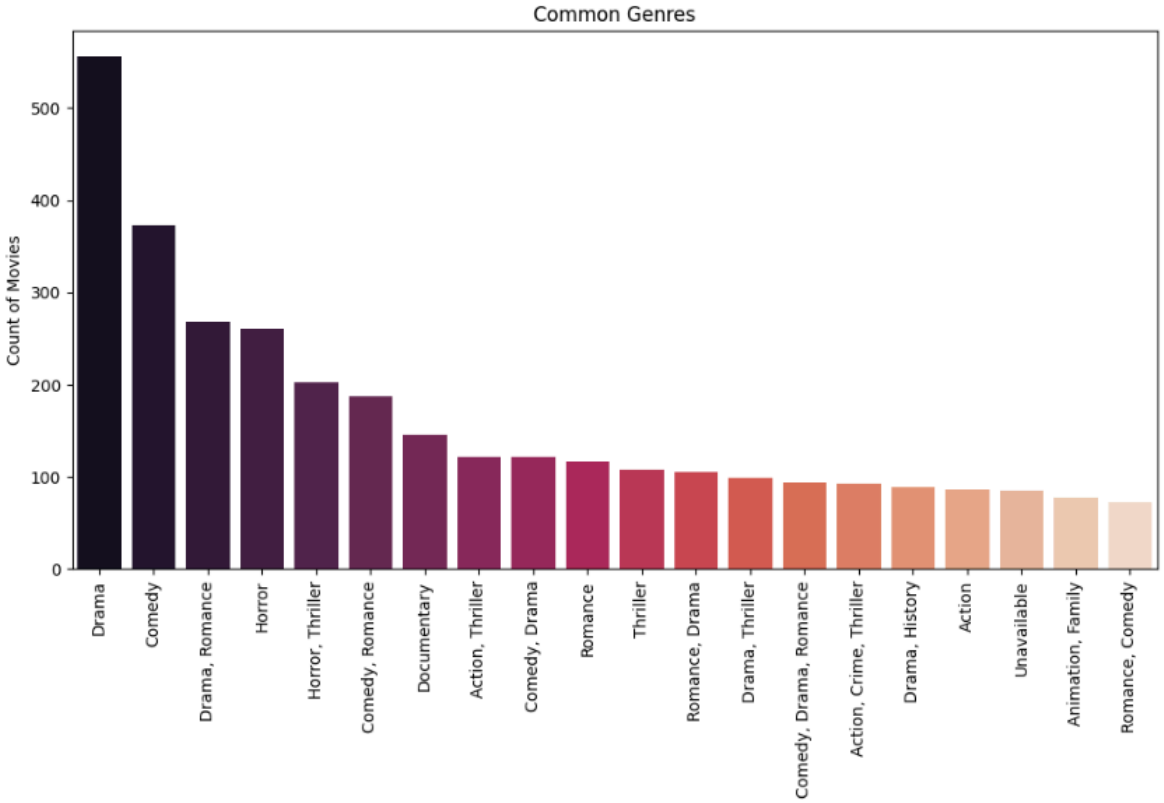
```
# Plot Distribution of Movies by Years
df = df.sort_values(by = "year")
plt.figure(figsize = (12,4) )
sns.histplot(df["year"])
plt.xticks(rotation = 90, fontsize = 6)
plt.title("Distribution of Movie Years")
plt.xlabel("Years")
plt.show()
```



2. What are the most common genres in the dataset? Use a bar chart to show their distribution.

```
# Aggregate data of genre with total count of movies
gb = df.groupby('genre').agg({'names':'count'})
gb = gb.sort_values(by = 'names', ascending = False)
gb = gb.head(20)
```

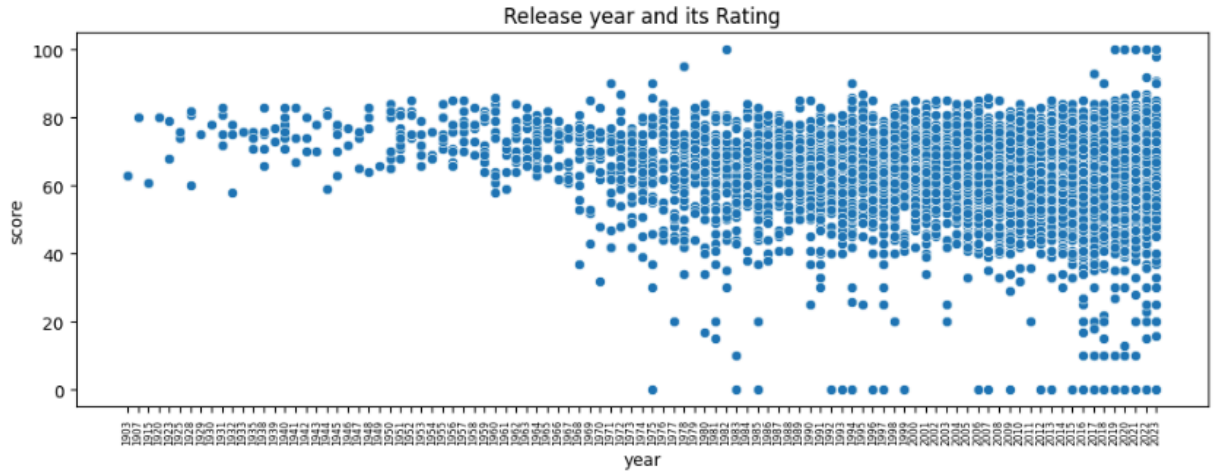
```
# Plot the distribution of Common Genres
plt.figure(figsize = (12, 6))
sns.barplot(x = gb.index, y = gb['names'], data = gb, hue = gb.index, palette = 'rocket')
plt.ylabel('Count of Movies')
plt.xlabel('Genre')
plt.title('Common Genres')
plt.xticks(rotation = 90)
plt.show()
```



BIVARIATE ANALYSIS

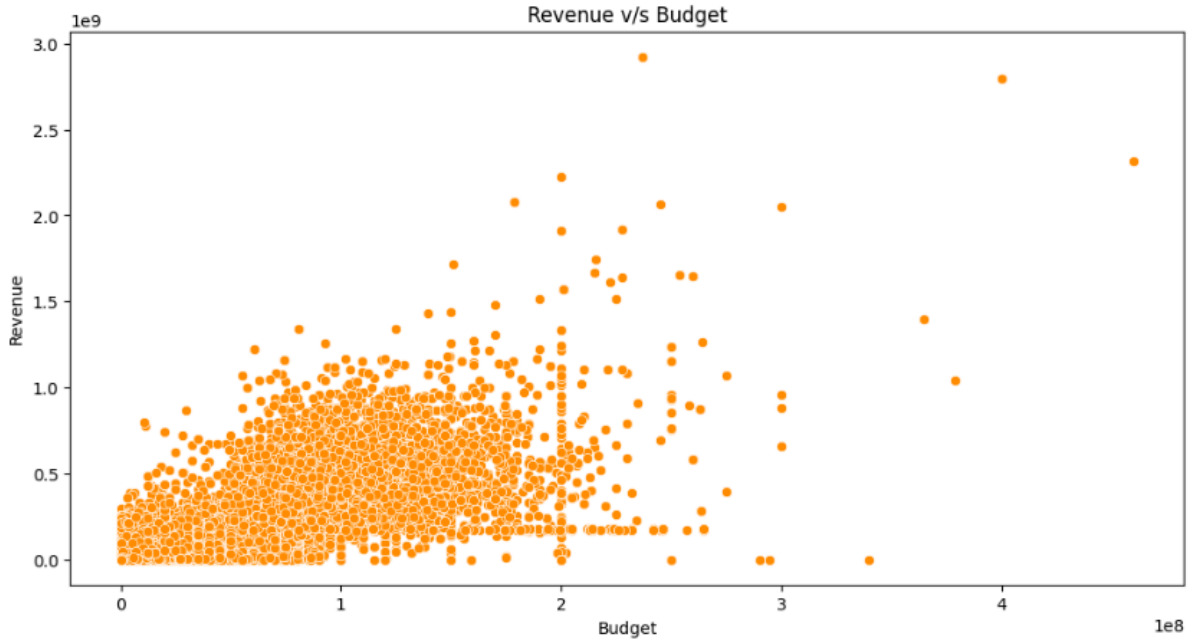
1.Is there a relationship between a movie's release year and its rating? Plot a scatter plot and describe any observed trend.

```
plt.figure(figsize = (12,4) )
plt.title("Release year and its Rating")
sns.scatterplot(x = "year", y = "score", data = df)
plt.xticks(rotation = 90, fontsize = 6)
plt.show()
```



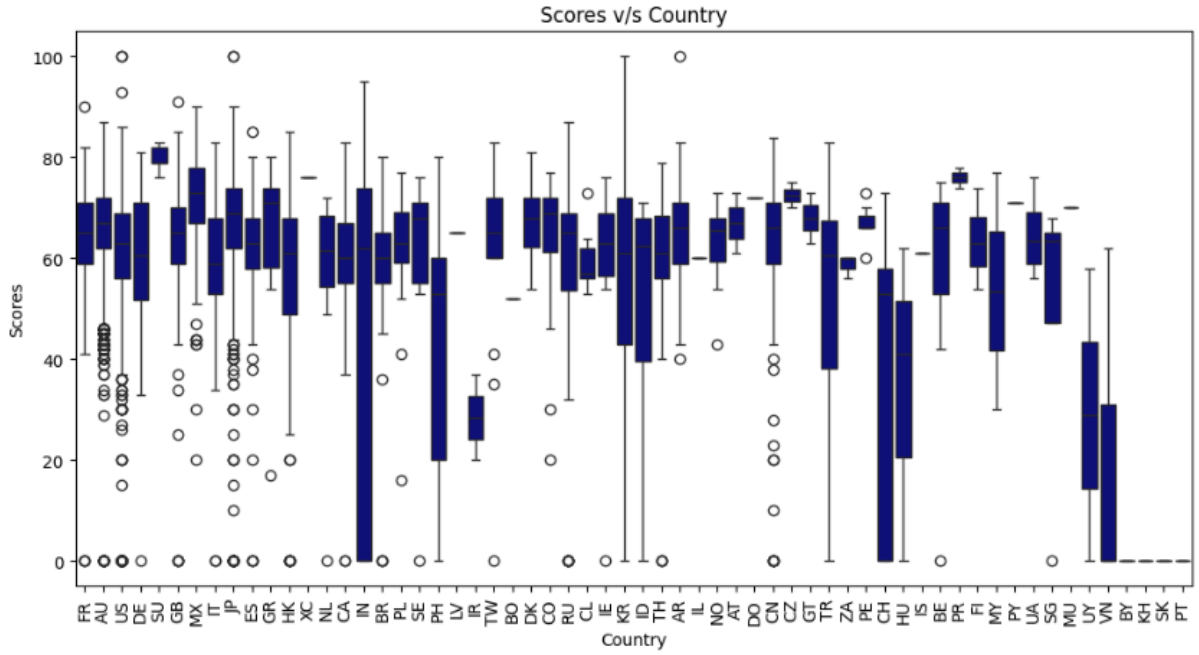
2. Is there a relationship between a movie's budget and its revenue? Plot a scatter plot and describe any observed trend.

```
# Plot the relationship between a movie's budget and its revenue
plt.figure(figsize = (12, 6))
sns.scatterplot(x = 'budget_x', y = 'revenue', data = df, color = 'darkorange')
plt.title('Revenue v/s Budget')
plt.ylabel('Revenue')
plt.xlabel('Budget')
plt.savefig("Revenue-Budget.jpg")
plt.show()
```



3. How do scores vary by country? Use a boxplot to visualize the differences in scores across country.

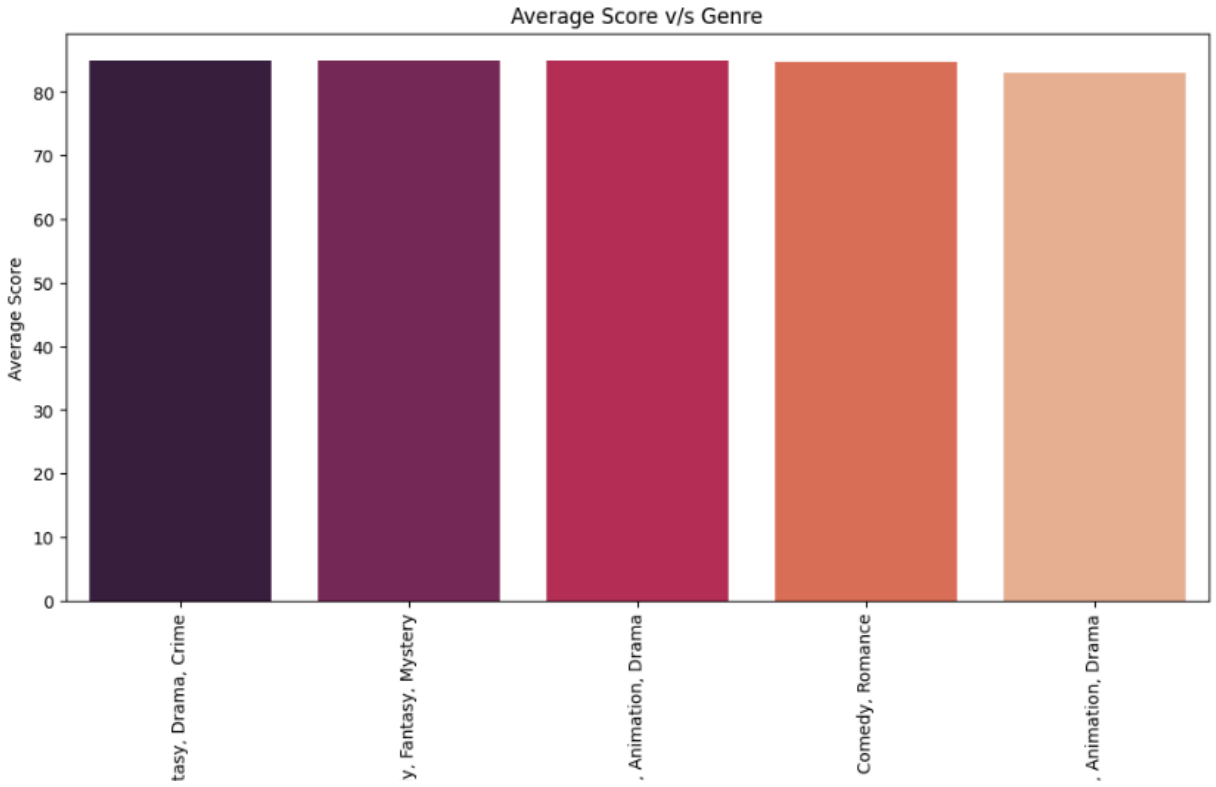
```
# Plot the difference in scores across countries
plt.figure(figsize = (12,6))
sns.boxplot(x = 'country', y = 'score', data = df, color = 'darkblue')
plt.title('Scores v/s Country')
plt.ylabel('Scores')
plt.xlabel('Country')
plt.xticks(rotation = 90)
plt.savefig('Scores-Country.jpg')
plt.show()
```



GENRE-SPECIFIC ANALYSIS:

1. Which genre has the highest average score? Calculate the average score for each genre and plot the results.

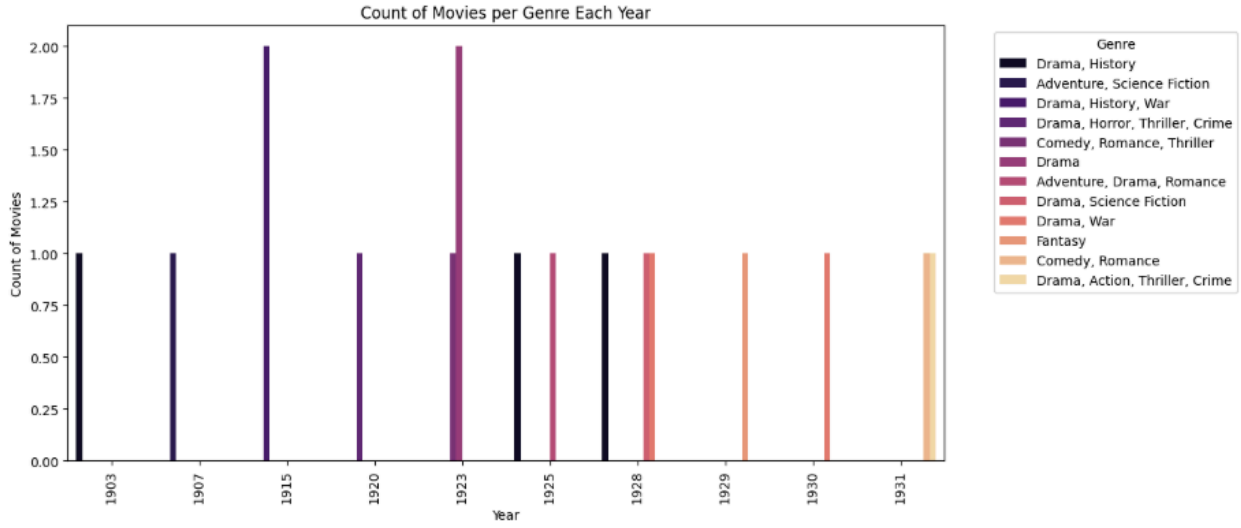
```
# Plot the graph of average score across genres
gb1 = df.groupby('genre').agg({'score':'mean'})
gb1 = gb1.sort_values(by = 'score', ascending = False)
gb1 = gb1.head(5)
plt.figure(figsize = (12, 6))
sns.barplot(x = gb1.index, y = gb1['score'], data = gb1, hue = gb1.index, palette = 'rocket')
plt.ylabel('Average Score')
plt.xlabel('Genre')
plt.title('Average Score v/s Genre')
plt.xticks(rotation = 90)
plt.savefig('Average Score-Genre.jpg')
plt.show()
```



2. How does the popularity of genres vary over time? Plot the number of movies released per genre each year.

```
# Extract year from the date_x column
df['year'] = df['date_x'].dt.strftime("%Y")
df['year'] = df['year'].astype(int)
```

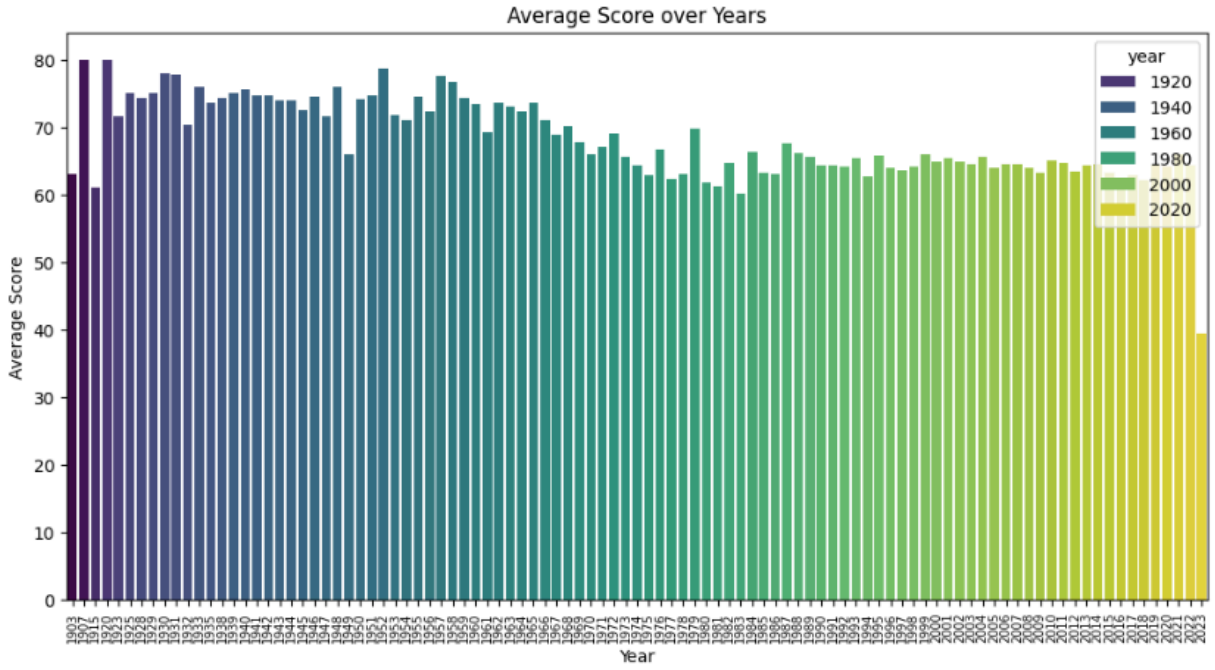
```
# Group by year and genre and count the number of movies in each group
gb2 = df.groupby(['year', 'genre']).size().reset_index(name='count')
gb2 = gb2.head(15)
# Plot the data
plt.figure(figsize=(12, 6))
sns.barplot(x = 'year', y = 'count', hue = 'genre', data = gb2, palette = 'magma')
plt.title('Count of Movies per Genre Each Year')
plt.xlabel('Year')
plt.ylabel('Count of Movies')
plt.xticks(rotation = 90)
plt.legend(title = 'Genre', bbox_to_anchor = (1.05, 1), loc = 'upper left')
plt.savefig('Count of Movies per Genre Each Year.jpg')
plt.show()
```



YEAR AND TREND ANALYSIS

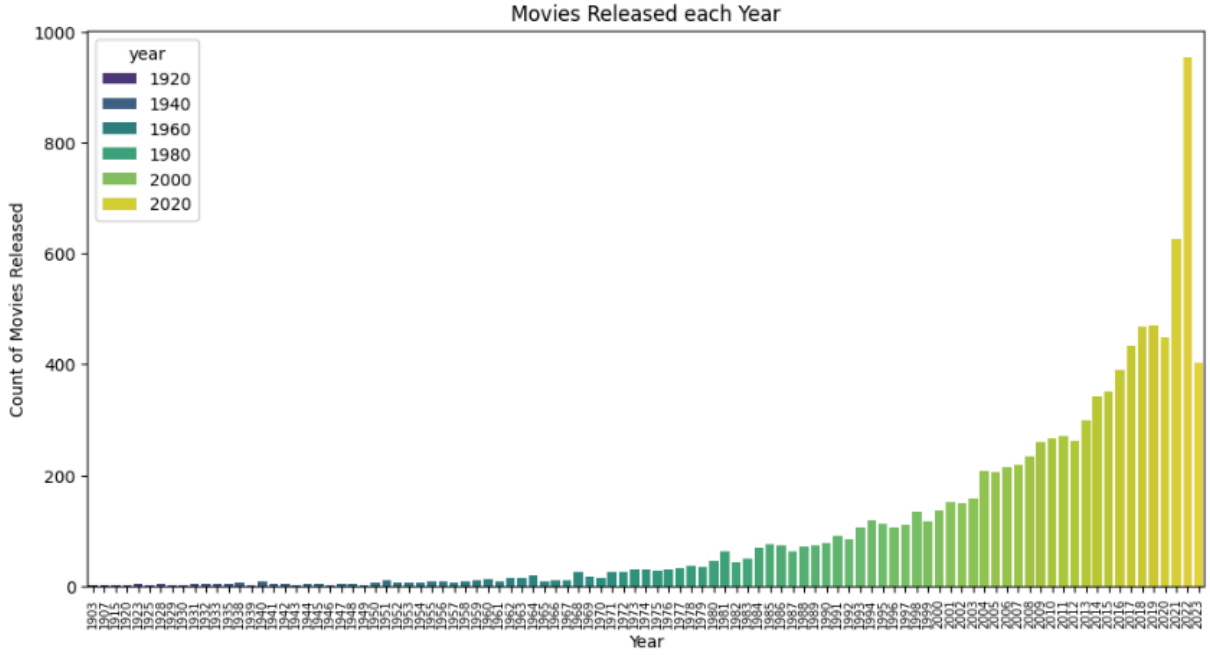
1. How has the average score changed over the years? Plot the average score for each year.

```
# Plot the average score across years
gb3 = df.groupby('year').agg({'score':'mean'})
plt.figure(figsize = (12, 6))
sns.barplot(x = gb3.index, y = gb3['score'], data = gb3, hue = gb3.index, palette = 'viridis')
plt.ylabel('Average Score')
plt.xlabel('Year')
plt.title('Average Score over Years')
plt.xticks(rotation = 90, fontsize = 7)
plt.savefig('Average Score over Years.jpg')
plt.show()
```



2. Which years had the highest and lowest number of movie releases? Plot the number of movies released each year.

```
# Plot the number of movies released each year
gb4 = df.groupby('year').agg({'names':'count'})
plt.figure(figsize = (12, 6))
sns.barplot(x = gb4.index, y = gb4['names'], data = gb4, hue = gb4.index, palette = 'viridis')
plt.ylabel('Count of Movies Released')
plt.xlabel('Year')
plt.title('Movies Released each Year')
plt.xticks(rotation = 90, fontsize = 7)
plt.savefig('Movies Released each Year.jpg')
plt.show()
```




```
# Year with highest movie releases
max_row = gb4.loc[gb4['names'] == gb4['names'].max()]
highest_releases_year = max_row.index[0]
print(highest_releases_year)
```

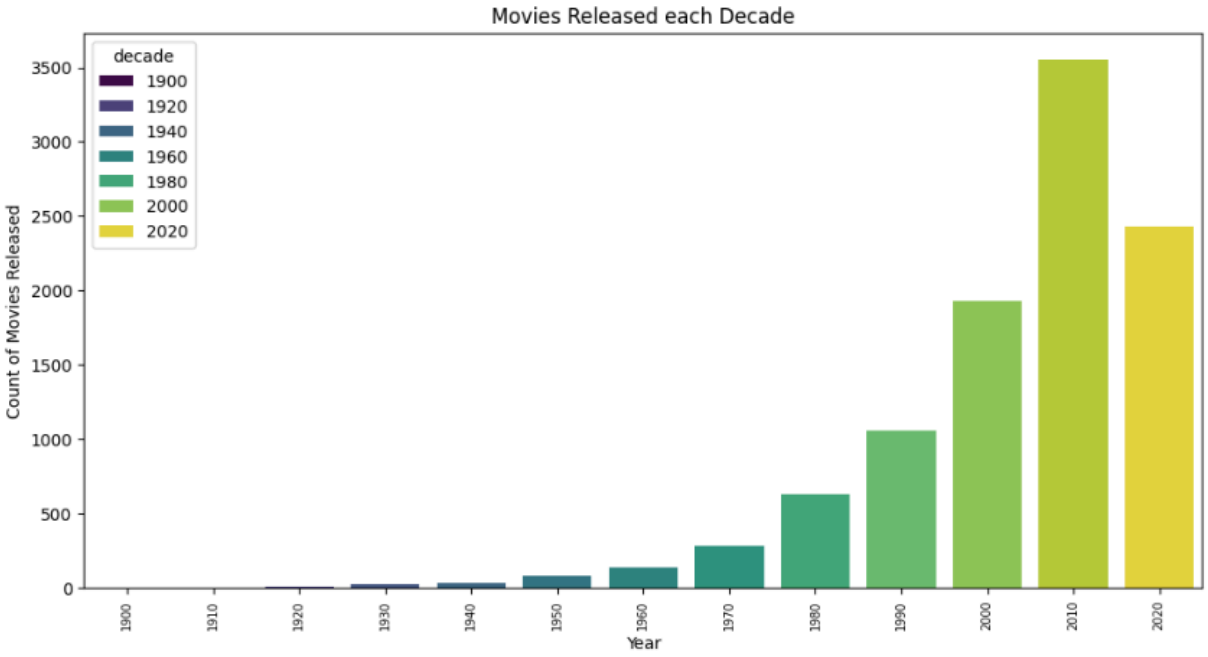
```
2022
```

```
# Year with lowest movie releases
min_row = gb4.loc[gb4['names'] == gb4['names'].min()]
lowest_releases_year = min_row.index[0]
print(lowest_releases_year)
```

```
1903
```

3. Plot the number of movies released each decade.

```
# Calculate decade from the year column
df['decade'] = df['year'] // 10 * 10
# Plot the number of movies released each decade
gb5 = df.groupby('decade').agg({'genre':'count'})
plt.figure(figsize = (12, 6))
sns.barplot(x = gb5.index, y = gb5['genre'], data = gb5, hue = gb5.index, palette = 'viridis')
plt.ylabel('Count of Movies Released')
plt.xlabel('Year')
plt.title('Movies Released each Decade')
plt.xticks(rotation = 90, fontsize = 7)
plt.savefig('Movies Released each Decade.jpg')
plt.show()
```



INSIGHTS AND SUMMARY

1. Based on your analysis, what are three major insights you learned about movie trends, popular genres, or movie scores?

Genre Popularity Trends: Over the years, action and adventure movies have consistently grown in popularity, likely due to advancements in visual effects and their strong global appeal. On the other hand, genres like westerns and musicals have declined, reflecting shifting audience preferences and cultural trends.

Budget vs. Movie Scores: High-budget films generally receive better ratings, benefiting from superior production quality, renowned directors, and A-list actors. However, some low-budget films, particularly independent dramas and thrillers, achieve critical acclaim due to strong storytelling and innovative filmmaking.

Seasonal Release Impact: Movies released during peak seasons—such as summer and the holidays—tend to perform better in terms of box office earnings and audience ratings. This suggests strategic release planning, where studios aim to maximize viewership and profitability.

2. Additional Questions for Further Exploration

1. How does a movie's revenue correlate with its audience and critic scores? Do commercially successful films generally receive better ratings?
2. What is the relationship between high IMDb scores and long-term box office success?
3. Do movies from certain countries perform better in global markets, and if so, why?
4. Which genres achieve the highest revenue-to-budget ratio, and what factors (e.g., marketing, storytelling, casting) contribute to their success?
5. How have streaming platforms affected traditional box office trends, and what impact do they have on movie ratings?