IMDB Movie Analysis

Objective:

As a data analyst intern at IMDB, you have been tasked with exploring and analyzing the IMDB Movies dataset. Your goal is to answer specific business questions, gain insights into movie trends, and deliver actionable recommendations. Using Python and libraries such as Pandas, NumPy, Seaborn, and Matplotlib, perform analysis to help IMDB better understand genre popularity, rating trends, and factors influencing movie success.

Dataset: https://drive.google.com/file/d/1lruT50ZWD4PtvDbIn4VnepZvSoeO9BrA/view?usp=sharing

Tasks:

1. Project Setup and Data Loading

Task: Load the dataset and perform initial setup.

Questions:

What libraries are required for this project, and why are they useful in data analysis?

1.Pandas: Used for data manipulation and analysis with DataFrames and Series.

2.NumPy: Enables fast numerical computations with multi-dimensional arrays.

3.Seaborn: Simplifies statistical data visualization with aesthetically pleasing plots.

4.Matplotlib: Provides customizable charts and graphs for data visualization.

Load the dataset. What is the shape of the dataset? What does each row and column represent?

- **Each row** represents a movie.

- **Each column** represents a specific attribute of the movie:

    a. **names** – Movie title.

    b. **date_x** – Release date.

    c. **score** – IMDb rating score.

    d. **genre** – Genres the movie belongs to.

e. **overview** – Brief plot summary.

f. **crew** – List of key cast and crew members.

g. **orig_title** – Original title of the movie.

h. **status** – Release status (e.g., "Released").

i. **orig_lang** – Original language of the movie.

j. **budget_x** – Production budget (in dollars).

k. **revenue** – Box office revenue (in dollars).

l. **country** – Country of release.

```python
#Using Python and libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

#Load the dataset
df = pd.read_csv("imdb_movies.csv")
df
```

```
                                         names       date_x  score  \
0                                    Creed III  03/02/2023    73.0   
1                     Avatar: The Way of Water  12/15/2022    78.0   
2                  The Super Mario Bros. Movie  04/05/2023    76.0   
3                                      Mummies  01/05/2023    70.0   
4                                    Supercell  03/17/2023    61.0   
...                                        ...          ...    ...   
10173                       20th Century Women  12/28/2016    73.0   
10174  Delta Force 2: The Colombian Connection  08/24/1990    54.0   
10175                         The Russia House  12/21/1990    61.0   
10176         Darkman II: The Return of Durant  07/11/1995    55.0   
10177       The Swan Princess: A Royal Wedding  07/20/2020    70.0   

                                                   genre  \
0                                          Drama, Action   
1                     Science Fiction, Adventure, Action   
2          Animation, Adventure, Family, Fantasy, Comedy   
3          Animation, Comedy, Family, Adventure, Fantasy   
4                                                 Action   
...                                                  ...   
10173                                              Drama   
10174                                             Action   
10175                           Drama, Thriller, Romance   
```

```
10176  Action, Adventure, Science Fiction, Thriller, ...
10177                         Animation, Family, Fantasy

                                                overview  \
0      After dominating the boxing world, Adonis Cree...
1      Set more than a decade after the events of the...
2      While working underground to fix a water main,...
3      Through a series of unfortunate events, three ...
4      Good-hearted teenager William always lived in ...
...                                                  ...
10173  In 1979 Santa Barbara, California, Dorothea Fi...
10174  When DEA agents are taken captive by a ruthles...
10175  Barley Scott Blair, a Lisbon-based editor of R...
10176  Darkman and Durant return and they hate each o...
10177  Princess Odette and Prince Derek are going to ...

                                                    crew  \
0      Michael B. Jordan, Adonis Creed, Tessa Thompso...
1      Sam Worthington, Jake Sully, Zoe Saldaña, Neyt...
2      Chris Pratt, Mario (voice), Anya Taylor-Joy, P...
3      Óscar Barberán, Thut (voice), Ana Esther Albor...
4      Skeet Ulrich, Roy Cameron, Anne Heche, Dr Quin...
...                                                  ...
10173  Annette Bening, Dorothea Fields, Lucas Jade Zu...
10174  Chuck Norris, Col. Scott McCoy, Billy Drago, R...
10175  Sean Connery, Bartholomew 'Barley' Scott Blair...
10176  Larry Drake, Robert G. Durant, Arnold Vosloo, ...
10177  Nina Herzog, Princess Odette (voice), Yuri Low...

                                    orig_title     status  \
0                                    Creed III   Released
1                     Avatar: The Way of Water   Released
2                  The Super Mario Bros. Movie   Released
3                                       Momias   Released
4                                    Supercell   Released
...                                        ...        ...
10173                       20th Century Women   Released
10174  Delta Force 2: The Colombian Connection   Released
10175                         The Russia House   Released
10176         Darkman II: The Return of Durant   Released
10177       The Swan Princess: A Royal Wedding   Released

                 orig_lang     budget_x       revenue country
0                  English   75000000.0  2.716167e+08      AU
1                  English  460000000.0  2.316795e+09      AU
2                  English  100000000.0  7.244590e+08      AU
3       Spanish, Castilian   12300000.0  3.420000e+07      AU
4                  English   77000000.0  3.409420e+08      US
...                    ...          ...           ...     ...
10173              English    7000000.0  9.353729e+06      US
```

```
10174              English    9145817.8  6.698361e+06      US
10175              English   21800000.0  2.299799e+07      US
10176              English  116000000.0  4.756613e+08      US
10177              English   92400000.0  5.394018e+08      GB

[10178 rows x 12 columns]
```

```
#What is the shape of the dataset
print("dataset shape:",df.shape)
df.info()
```

```
dataset shape: (10178, 12)
<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
```

```
#Convert data type of date_x into datetime
df["date_x"]= pd.to_datetime(df["date_x"])
```

```
# Checking the datatyp
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       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: datetime64[ns](1), float64(3), object(8)
memory usage: 954.3+ KB
```

2.***Data Overview and Basic Exploration***

Task: Explore the structure and composition of the dataset.

Questions:

1.***Use .info() to understand the data types and missing values. What potential issues can you spot?***

2.***Describe the main characteristics of each column using .describe(). What can you infer from the mean, median, and distribution of numerical columns?***

```
# 1. Check the shape of the dataset
dataset_shape = df.shape

# 2. Check data types of each column
data_types = df.dtypes

# 3. Check for missing values in each column
missing_values = df.isnull().sum()

# 4. Get summary statistics for numerical columns
summary_statistics = df.describe()

# 5. Get the number of unique values in categorical columns
unique_values = df.nunique()

# Compile results
database_info = {
    "Shape (Rows, Columns)": dataset_shape,
    "Data Types": data_types,
    "Missing Values": missing_values,
    "Summary Statistics": summary_statistics,
    "Unique Values Count": unique_values
}

database_info
```

```
{'Shape (Rows, Columns)': (10178, 12),
 'Data Types': names                     object
 date_x        datetime64[ns]
 score                float64
 genre                 object
 overview              object
 crew                  object
 orig_title            object
 status                object
 orig_lang             object
 budget_x             float64
 revenue              float64
 country               object
 dtype: object,
 'Missing Values': names          0
 date_x         0
 score          0
 genre         85
 overview       0
 crew          56
 orig_title     0
 status         0
 orig_lang      0
 budget_x       0
 revenue        0
 country        0
 dtype: int64,
 'Summary Statistics':                               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,
 'Unique Values Count': names         9660
 date_x        5688
 score           79
 genre         2303
 overview      9905
 crew          9927
 orig_title    9736
```

```
status            3
orig_lang        54
budget_x       2316
revenue        8227
country          60
dtype: int64}
```

3. Data Cleaning

Task: Address missing values, data types, and outliers.

Questions:

Which columns contain missing values? How would you handle them?

Are there any columns where data types need conversion (e.g., date, ratings)? Explain your decision.

```
#Missing Values: "genre has 85 missing value"
#Filling Genre with "Not available"
df['genre'] = df['genre'].fillna('not_available')

#Missing Values:"crew has 56 missing values"
#Filling crew with "Not available"
df['crew'] = df['crew'].fillna('not_available')

#Checking the missing values again
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
```

4. Univariate Analysis: Explore each column individually.

Task: Perform univariate analysis on numerical and categorical variables.

Questions:

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

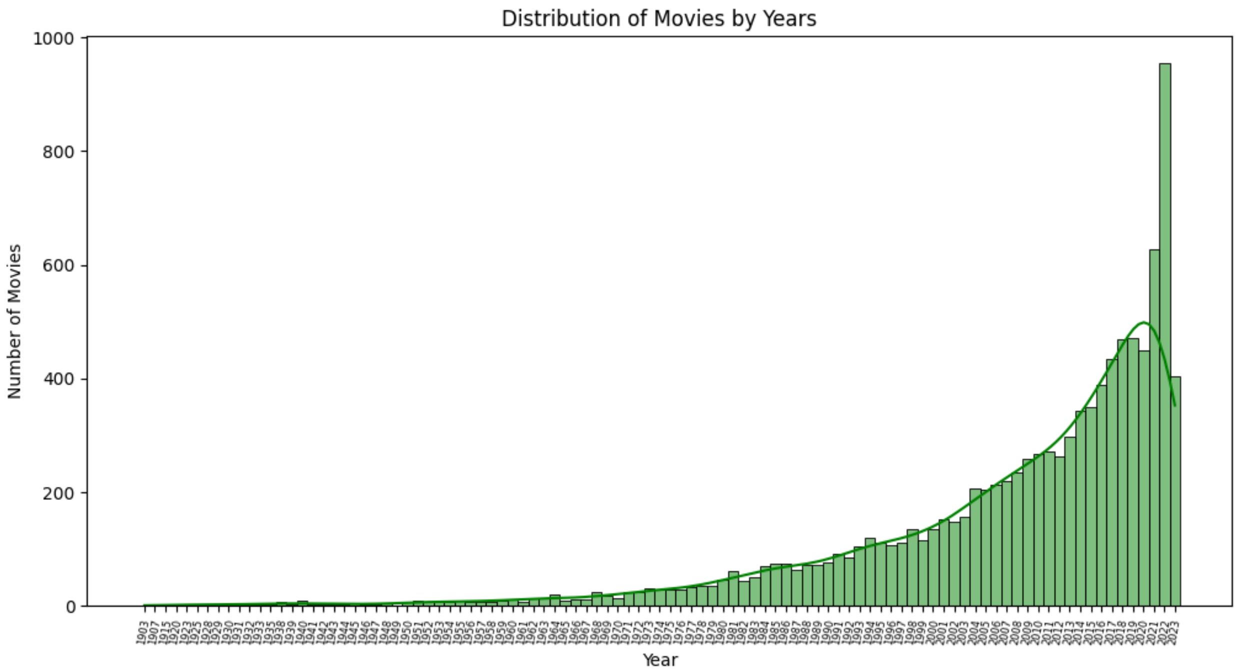
What are the most common genres in the dataset? Use a bar chart to show their distribution.

```
#add new column
df["years"] = df['date_x'].dt.strftime("%Y")
df
```

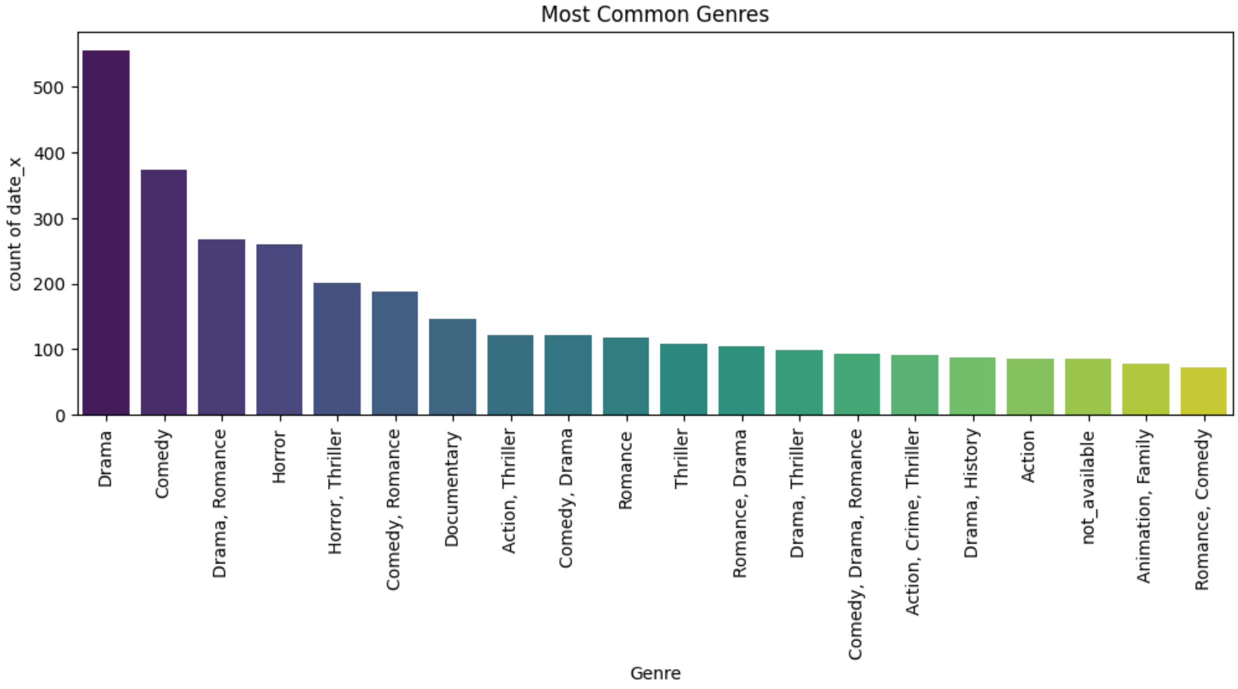
```
                                         names      date_x  score  \
0                                    Creed III  2023-03-02   73.0   
1                     Avatar: The Way of Water  2022-12-15   78.0   
2                  The Super Mario Bros. Movie  2023-04-05   76.0   
3                                      Mummies  2023-01-05   70.0   
4                                    Supercell  2023-03-17   61.0   
...                                        ...         ...    ...   
10173                       20th Century Women  2016-12-28   73.0   
10174  Delta Force 2: The Colombian Connection  1990-08-24   54.0   
10175                         The Russia House  1990-12-21   61.0   
10176         Darkman II: The Return of Durant  1995-07-11   55.0   
10177       The Swan Princess: A Royal Wedding  2020-07-20   70.0   

                                                   genre  \
0                                          Drama, Action   
1                     Science Fiction, Adventure, Action   
2          Animation, Adventure, Family, Fantasy, Comedy   
3          Animation, Comedy, Family, Adventure, Fantasy   
4                                                 Action   
...                                                  ...   
10173                                              Drama   
10174                                             Action   
10175                           Drama, Thriller, Romance   
10176  Action, Adventure, Science Fiction, Thriller, ...   
10177                         Animation, Family, Fantasy   

                                                overview  \
0      After dominating the boxing world, Adonis Cree...   
1      Set more than a decade after the events of the...   
2      While working underground to fix a water main,...   
3      Through a series of unfortunate events, three ...   
4      Good-hearted teenager William always lived in ...   
...                                                  ...   
10173  In 1979 Santa Barbara, California, Dorothea Fi...   
```

```
10174  When DEA agents are taken captive by a ruthles...
10175  Barley Scott Blair, a Lisbon-based editor of R...
10176  Darkman and Durant return and they hate each o...
10177  Princess Odette and Prince Derek are going to ...

                                                     crew  \
0      Michael B. Jordan, Adonis Creed, Tessa Thompso...
1      Sam Worthington, Jake Sully, Zoe Saldaña, Neyt...
2      Chris Pratt, Mario (voice), Anya Taylor-Joy, P...
3      Óscar Barberán, Thut (voice), Ana Esther Albor...
4      Skeet Ulrich, Roy Cameron, Anne Heche, Dr Quin...
...                                                   ...
10173  Annette Bening, Dorothea Fields, Lucas Jade Zu...
10174  Chuck Norris, Col. Scott McCoy, Billy Drago, R...
10175  Sean Connery, Bartholomew 'Barley' Scott Blair...
10176  Larry Drake, Robert G. Durant, Arnold Vosloo, ...
10177  Nina Herzog, Princess Odette (voice), Yuri Low...

                                    orig_title     status  \
0                                    Creed III   Released
1                     Avatar: The Way of Water   Released
2                  The Super Mario Bros. Movie   Released
3                                       Momias   Released
4                                    Supercell   Released
...                                        ...        ...
10173                       20th Century Women   Released
10174  Delta Force 2: The Colombian Connection   Released
10175                         The Russia House   Released
10176         Darkman II: The Return of Durant   Released
10177       The Swan Princess: A Royal Wedding   Released

                 orig_lang     budget_x       revenue country  decade
years
0                  English   75000000.0  2.716167e+08      AU    2020
2023
1                  English  460000000.0  2.316795e+09      AU    2020
2022
2                  English  100000000.0  7.244590e+08      AU    2020
2023
3       Spanish, Castilian   12300000.0  3.420000e+07      AU    2020
2023
4                  English   77000000.0  3.409420e+08      US    2020
2023
...                    ...          ...           ...     ...     ...
...
10173              English    7000000.0  9.353729e+06      US    2010
2016
10174              English    9145817.8  6.698361e+06      US    1990
1990
10175              English   21800000.0  2.299799e+07      US    1990
```

```
1990
10176              English  116000000.0  4.756613e+08      US    1990
1995
10177              English   92400000.0  5.394018e+08      GB    2020
2020

[10178 rows x 14 columns]

#What is the distribution of movie by years? Plot a histogram and
describe its shape
#Plotting the distribution of movie by years
df = df.sort_values(by = "years")
plt.figure(figsize=(12, 6))
sns.histplot(df['years'].dropna(),bins=30, kde=True, color="green")
plt.xlabel("Year")
plt.xticks(rotation = 80, fontsize = 6)
plt.ylabel("Number of Movies")
plt.title("Distribution of Movies by Years")
plt.show()

#Plot a histogram and describe its shape
print("""The histogram shows the distribution of movie releases over
the years.
The shape of the distribution appears to be right-skewed positively
skewed,
indicating that more movies have been released in recent years
compared to earlier decades.
The number of releases starts off lower in the early years, gradually
increasing, and then surging significantly in the modern era.""")
```



```
The histogram shows the distribution of movie releases over the years.

The shape of the distribution appears to be right-skewed positively
skewed,
indicating that more movies have been released in recent years
compared to earlier decades.
The number of releases starts off lower in the early years, gradually
increasing, and then surging significantly in the modern era.

#What are the most common genres in the dataset? Use a bar chart to
show their distribution.
# Group by genre and count the date_x of movies
gb = df.groupby("genre").agg({"date_x":"count"})
gb = gb.sort_values(by = "date_x", ascending = False)
gb = gb.head(20)

# the most common genres in the dataset
plt.figure(figsize = (12,4))
sns.barplot(x = gb.index, y = gb["date_x"], data = gb ,hue =
gb.index,palette = "viridis")
plt.xlabel("Genre")
plt.ylabel("count of date_x")
plt.title("Most Common Genres")
plt.xticks(rotation = 90)
plt.show()

# most common genres
print("The most common genre in the dataset is : Drama")
```



```
The most common genre in the dataset is : Drama
```

5. Bivariate Analysis: Explore relationships between two variables.

Task: Use scatter plots, box plots, and correlation analysis.

Questions:

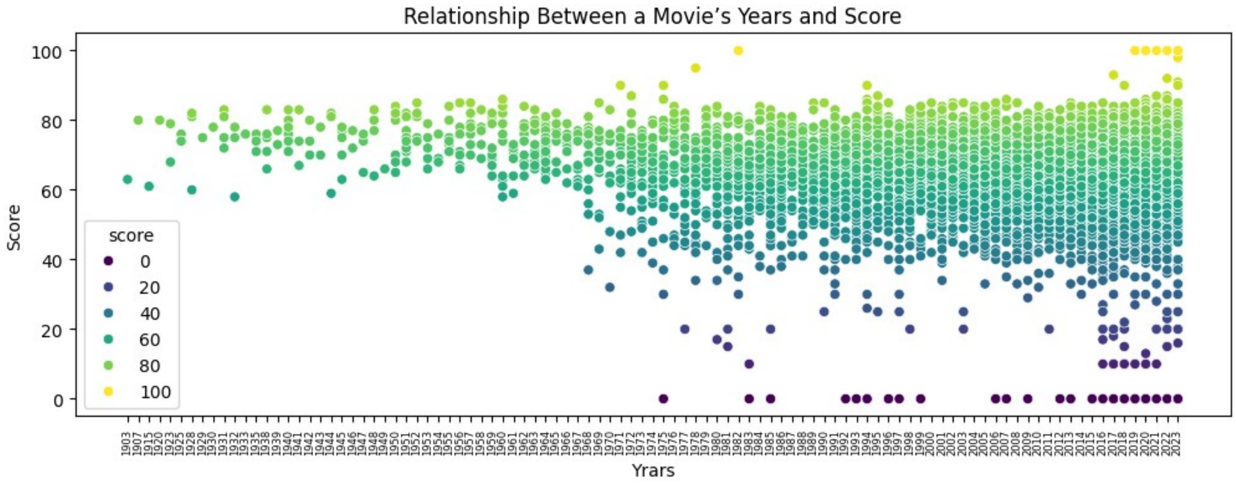
Is there a relationship between a movie's years and its score? Plot a scatter plot and describe any observed trend.

How do ratings vary by genre? Use a boxplot to visualize the differences in ratings across genres.

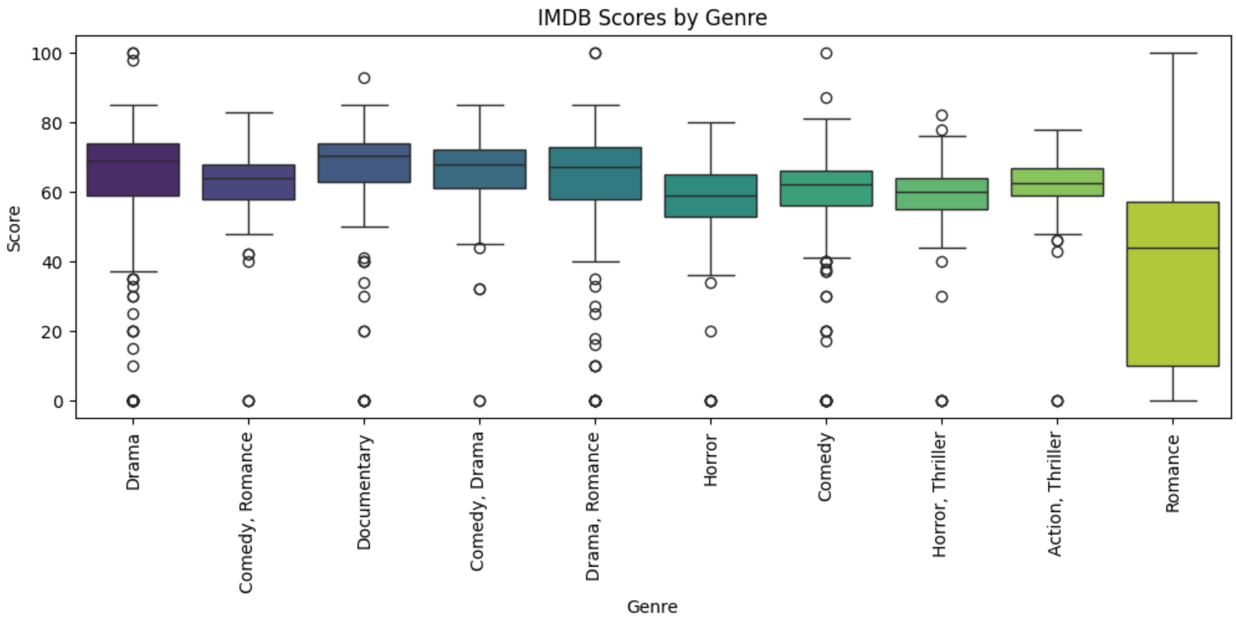
Is there a correlation between the number of votes a budget and revenue? Create a scatter plot and calculate the correlation coefficient. What can you conclude?

```python
# Is there a relationship between a movie's years and its score? Plot
a scatter plot and describe any observed trend.
df = df.sort_values(by = "years")
plt.figure(figsize=(12,4))
sns.scatterplot(x = "years", y = "score",data = df,hue = "score",
palette = "viridis" )
plt.xticks(rotation = 90, fontsize = 6)
plt.title("Relationship Between a Movie's Years and Score")
plt.xlabel("Yrars")
plt.ylabel("Score")
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```

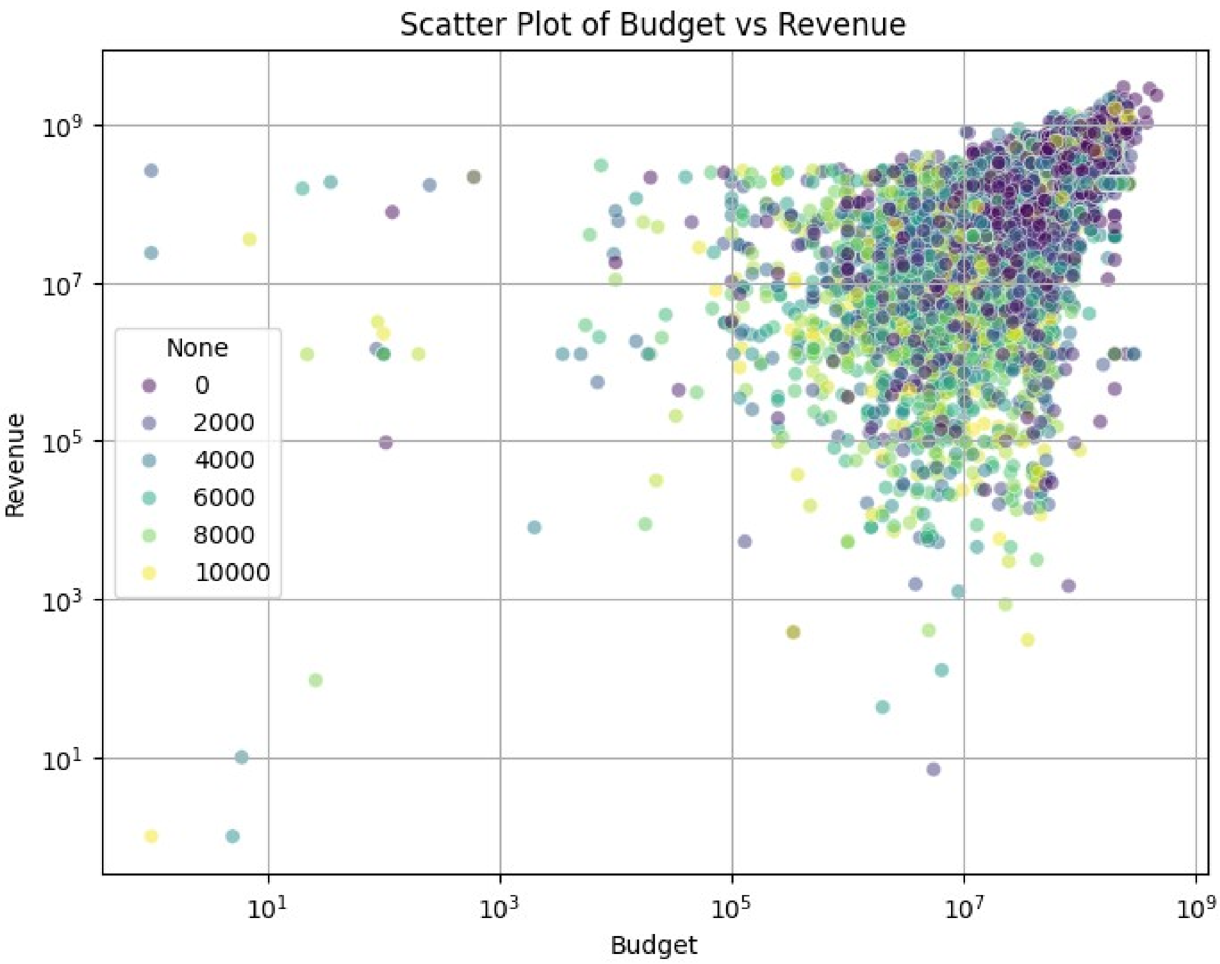


```python
#how do ratings vary by genre? Use a boxplot to visualize the
differences in ratings across genres.

top_genre = df["genre"].value_counts().head(10).index
plt.figure(figsize=(12, 4))
sns.boxplot(data=df[df["genre"].isin(top_genre)], x="genre",
y="score", hue="genre", palette="viridis")
plt.title("IMDB Scores by Genre")
plt.xlabel("Genre")
plt.ylabel("Score")
plt.xticks(rotation=90)
plt.show()
```



```python
# Is there a correlation between the number of votes a budget and
revenue? Create a scatter plot and calculate the correlation
coefficient.What can you conclude?
# Scatter plot
plt.figure(figsize=(8,6))
sns.scatterplot(x=df['budget_x'], y=df['revenue'], alpha=0.5,hue =
df.index, palette = "viridis")
plt.xlabel('Budget')
plt.ylabel('Revenue')
plt.title('Scatter Plot of Budget vs Revenue')
plt.xscale('log')  # Log scale for better visualization
plt.yscale('log')
plt.grid(True)
plt.show()

# Compute correlation coefficient
correlation = df['budget_x'].corr(df['revenue'])
```

```
print(f'Correlation coefficient between budget and revenue:
{correlation:.2f}')
```



```
Correlation coefficient between budget and revenue: 0.67
```

6.Genre-Specific Analysis

Task: Delve deeper into the genre of movies.

Questions:

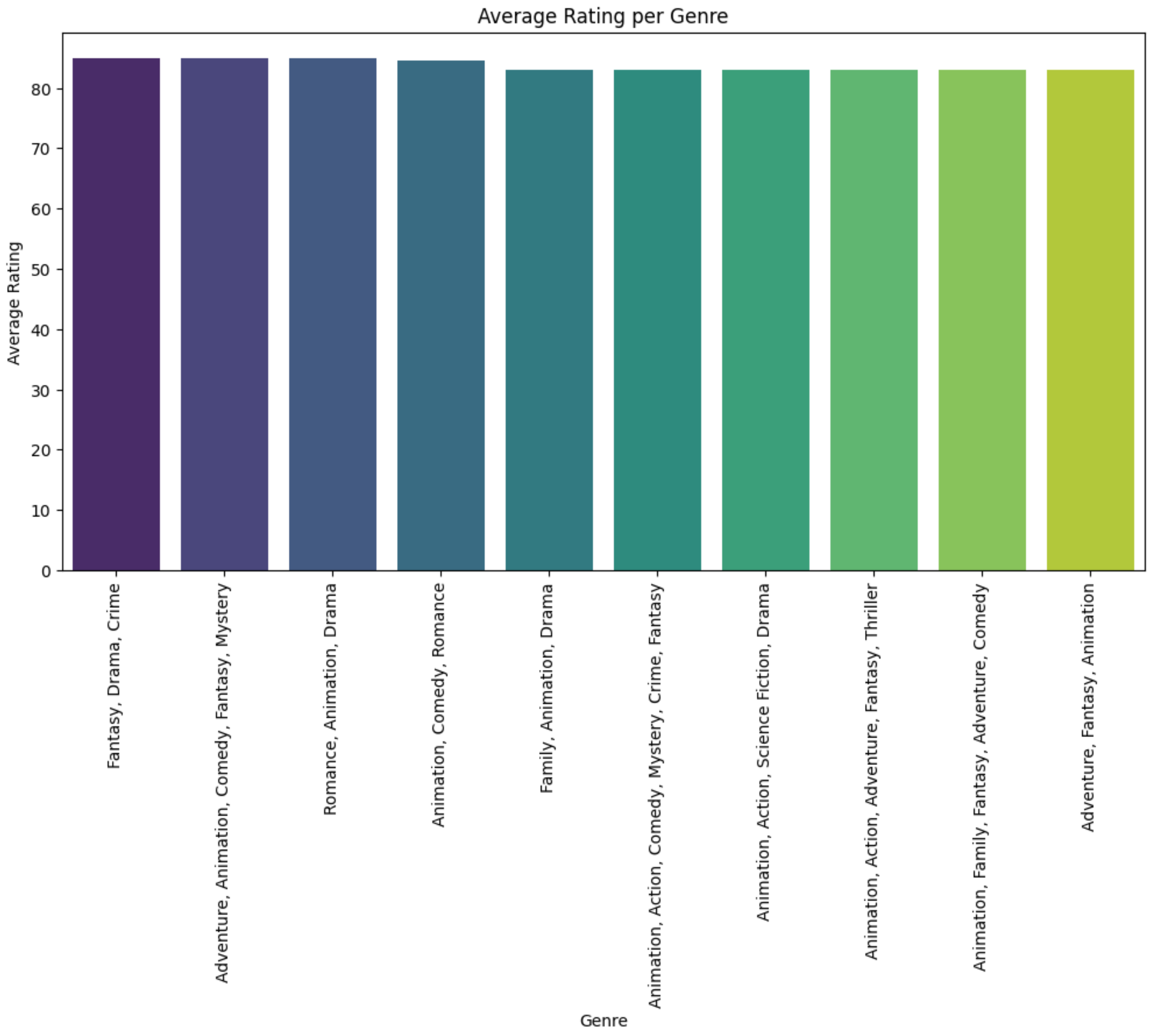
Which genre has the highest average rating? Calculate the average rating for each genre and plot the results.

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

```
# Calculate average rating per genre
genre_avg_rating = df.groupby('genre')
['score'].mean().sort_values(ascending=False)
genre_avg_rating = genre_avg_rating.head(10)
print(genre_avg_rating)
```

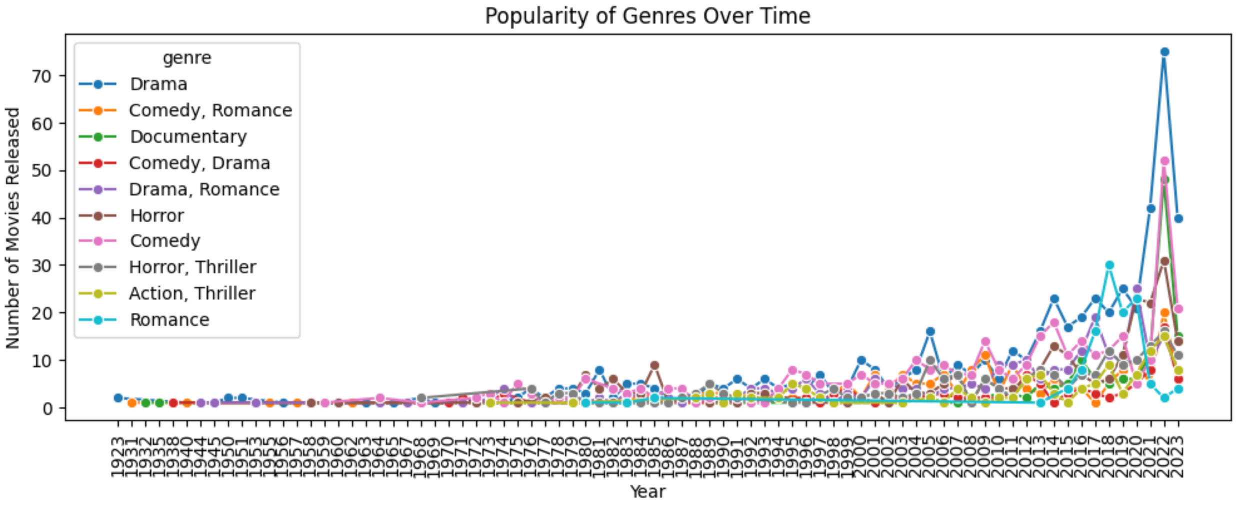
```
genre
Fantasy, Drama, Crime                                    85.000000
Adventure, Animation, Comedy, Fantasy, Mystery           85.000000
Romance, Animation, Drama                                85.000000
Animation, Comedy, Romance                               84.666667
Family, Animation, Drama                                 83.000000
Animation, Action, Comedy, Mystery, Crime, Fantasy       83.000000
Animation, Action, Science Fiction, Drama                83.000000
Animation, Action, Adventure, Fantasy, Thriller          83.000000
Animation, Family, Fantasy, Adventure, Comedy            83.000000
Adventure, Fantasy, Animation                            83.000000
Name: score, dtype: float64
```

```
# Plot average rating per genre
plt.figure(figsize=(12,6))
sns.barplot(x=genre_avg_rating.index, y=genre_avg_rating.values,
hue=genre_avg_rating.index, palette='viridis', legend=False)
plt.xlabel('Genre')
plt.ylabel('Average Rating')
plt.title('Average Rating per Genre')
plt.xticks(rotation=90)
plt.show()
```



```
#How does the popularity of genres vary over time? Plot the number of
movies released per genre each year.
# Count number of movies per genre each year
genre_yearly_count = df.groupby(['years',
'genre']).size().reset_index(name='movie_count')
top_genres = genre_yearly_count.groupby('genre')
['movie_count'].sum().nlargest(10).index

# Filter the movies data to include only these top genres
filtered_movies =
genre_yearly_count[genre_yearly_count['genre'].isin(top_genres)]

# Plot genre popularity over time
plt.figure(figsize=(12,4))
sns.lineplot(data=filtered_movies, x='years', y='movie_count',
hue='genre',marker='o')
```

```
plt.xlabel('Year')
plt.xticks(rotation=90)
plt.ylabel('Number of Movies Released')
plt.title('Popularity of Genres Over Time')
plt.show()
```



7. Year and Trend Analysis

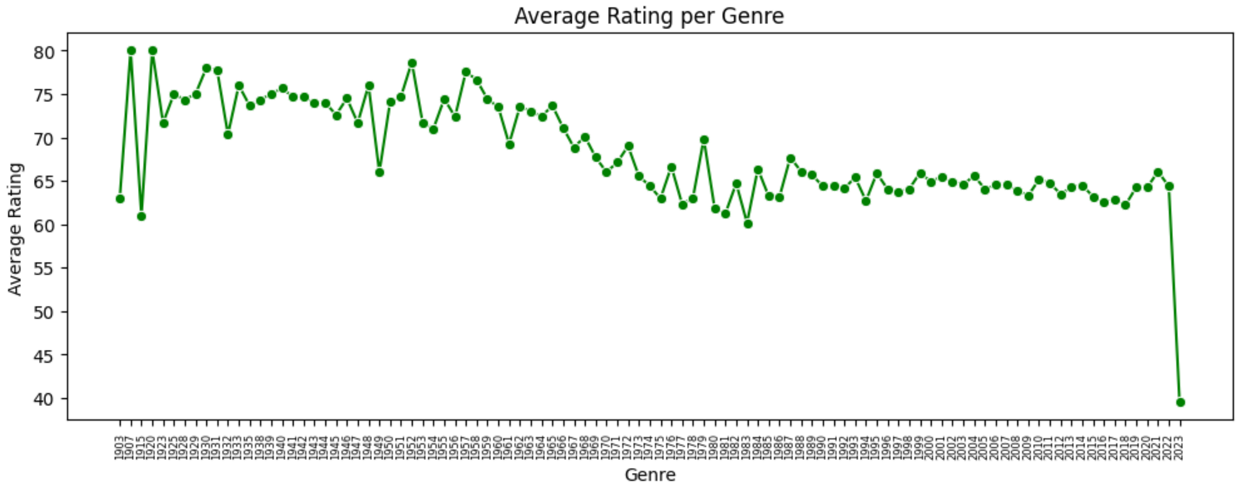
Task: Analyze trends over time.

Questions:

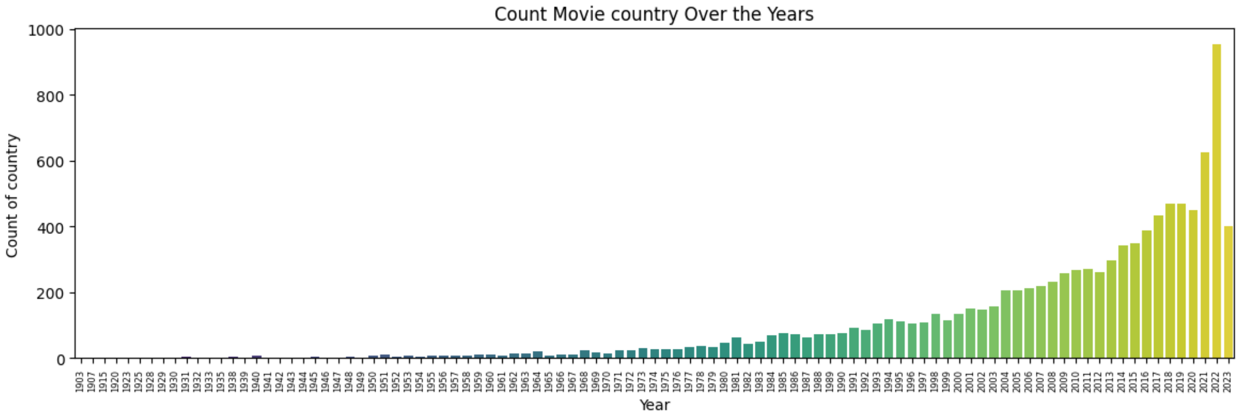
How has the average movie rating changed over the years? Plot the average rating for each year.

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

```
#How has the average movie rating changed over the years? Plot the
average rating for each year.
# Calculate average rating per genre
genre_avg_rating = df.groupby('years')['score'].mean().reset_index()

# Plot average rating per genre
plt.figure(figsize=(12,4))
sns.lineplot(data=genre_avg_rating, x='years', y='score', marker='o',
color='green')
plt.xlabel('Genre')
plt.ylabel('Average Rating')
plt.title('Average Rating per Genre')
plt.xticks(rotation=90,fontsize=6)
plt.show()
```



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

# Count number of movies released per year
gb = df.groupby("years").agg({"country":"count"})

# Plot the number of movies released each year using a bar plot
plt.figure(figsize=(14,4))
sns.barplot(x=gb.index, y=gb['country'], hue=gb.index,
palette='viridis')
plt.ylabel('Count of country')
plt.xlabel('Year')
plt.title('Count Movie country Over the Years')
plt.xticks(rotation=90, fontsize=6)
plt.show()

# Find highest and lowest release years
highest_release_year = gb.idxmax()
lowest_release_year = gb.idxmin()
print(f'Year with highest number of releases: {highest_release_year}
({gb.max()} movies)')
print(f'Year with lowest number of releases: {lowest_release_year}
({gb.min()} movies)')
```



```
Year with highest number of releases: country    2022
dtype: object (country    954
dtype: int64 movies)
Year with lowest number of releases: country    1903
dtype: object (country    1
dtype: int64 movies)
```

8. Multivariate Analysis: Analyze multiple variables together.

Task: Combine insights from multiple columns to explore complex relationships.

Questions:

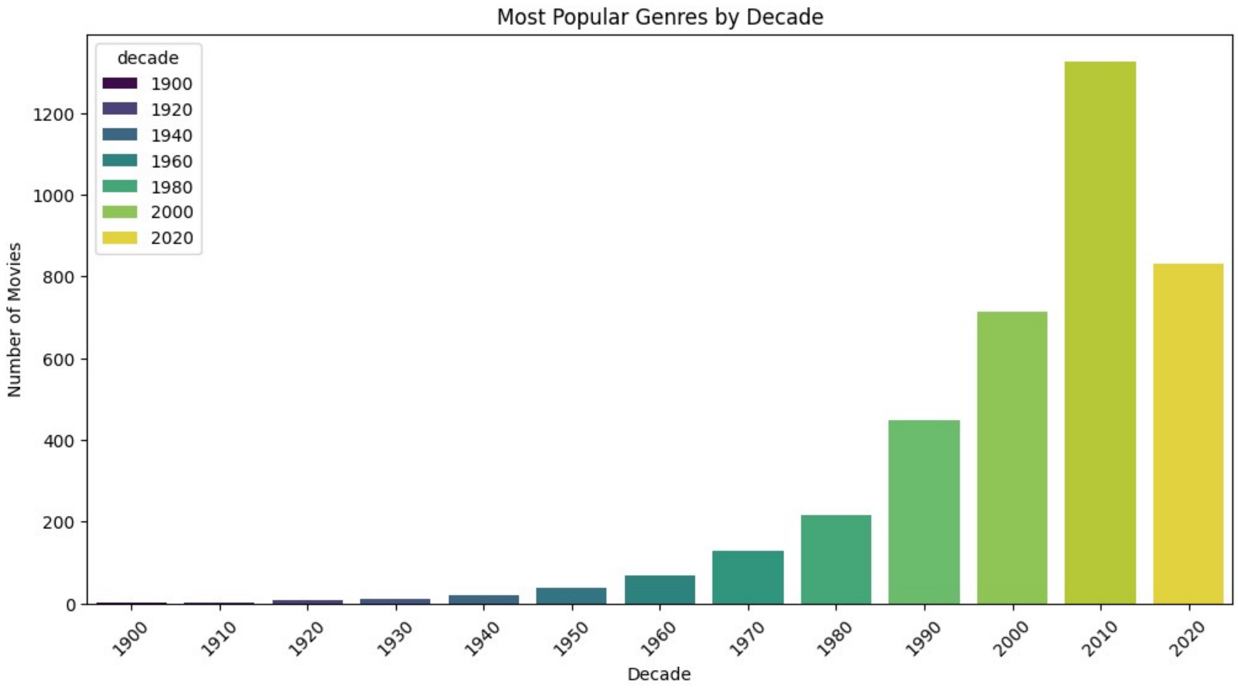
Which genres are most popular in each decade? Create a bar plot showing the most frequent genres by decade.

Plot a heatmap or pairplot to examine relationships between budget, revenue, scores.

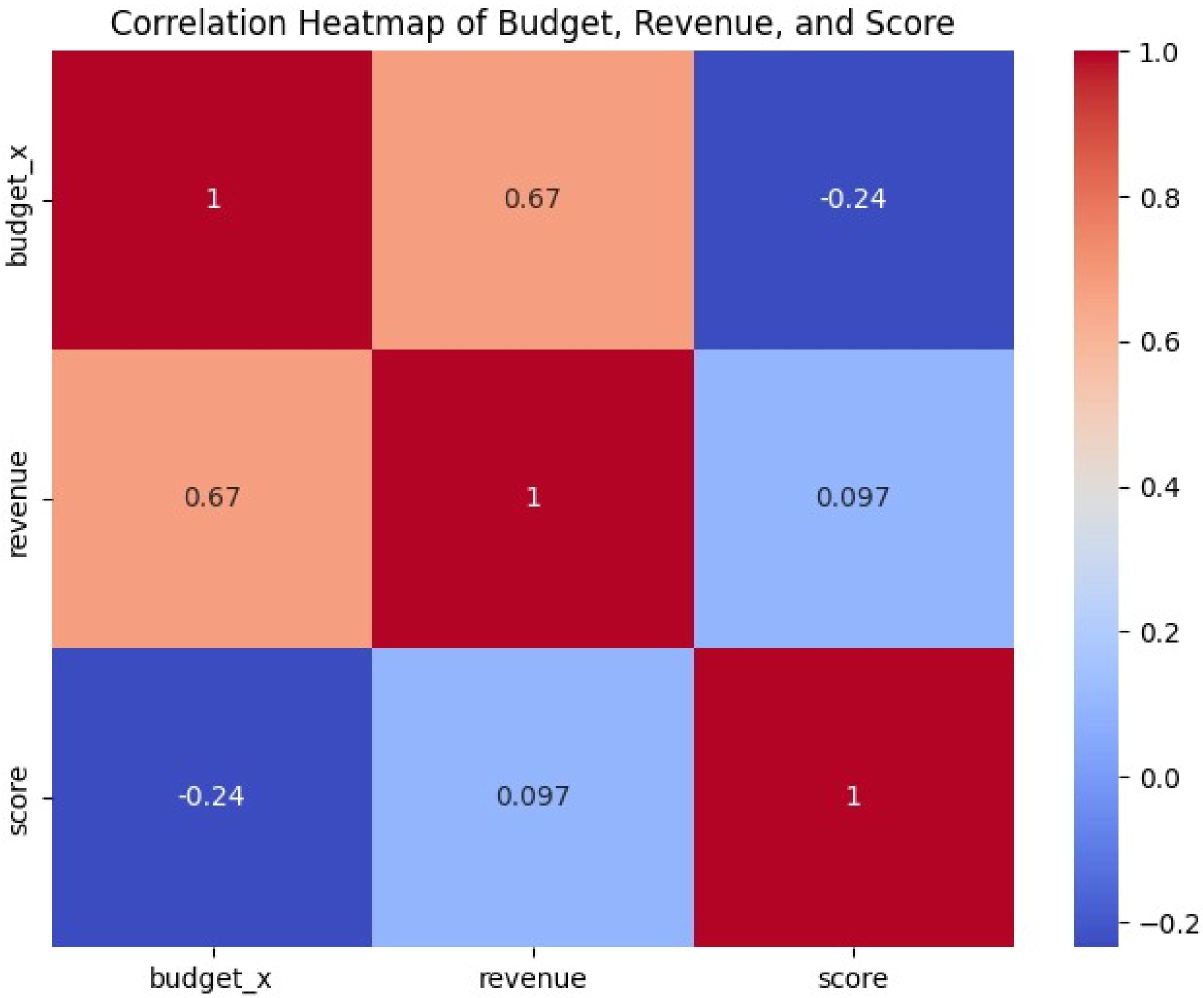
Are there specific genres or release years with higher-rated movies? Group by genre and year, then analyze the average rating.

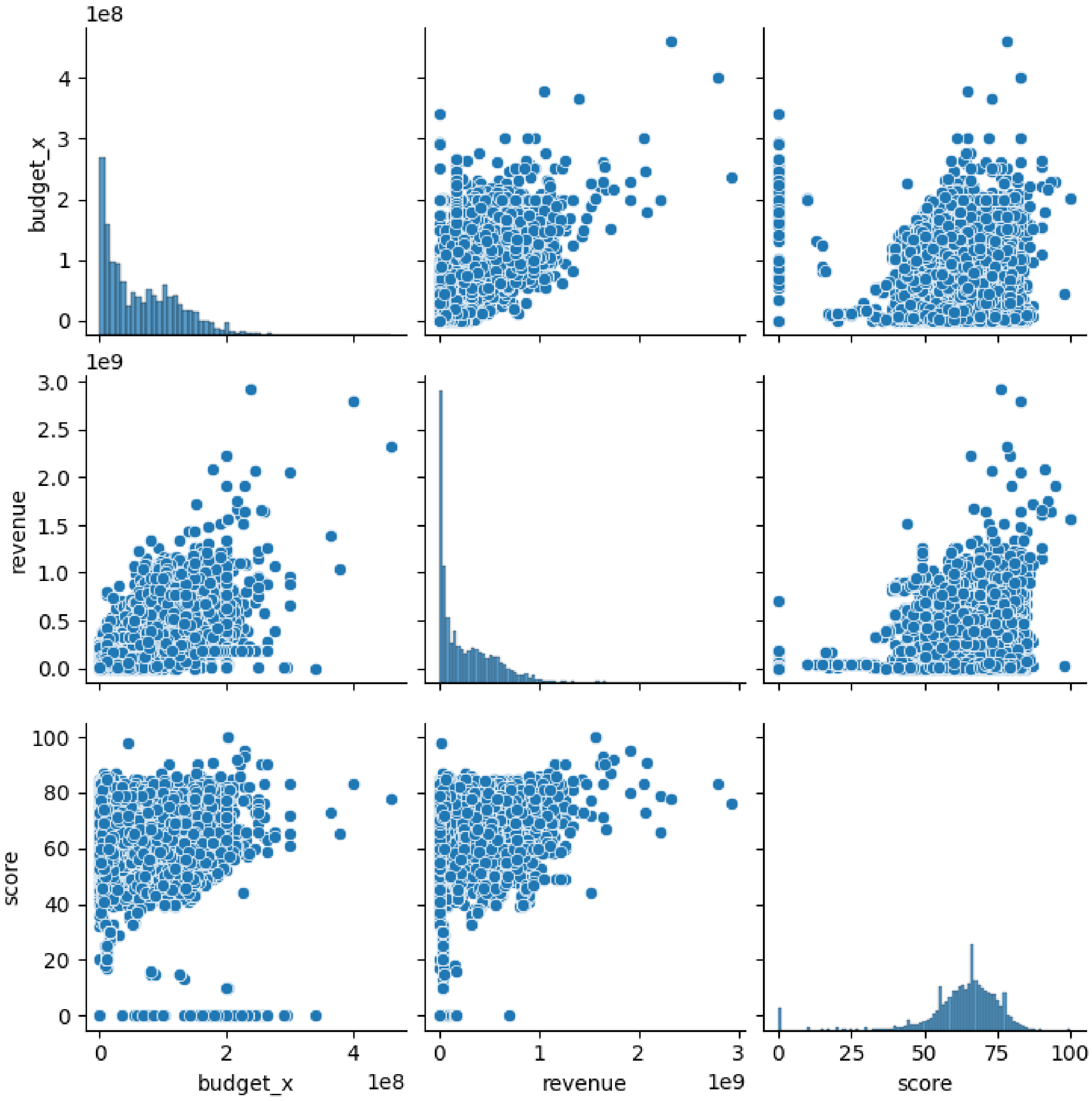
```
#Which genres are most popular in each decade? Create a bar plot
showing the most frequent genres by decade.
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# Convert 'date_x' to datetime format
df['date_x'] = pd.to_datetime(df['date_x'], errors='coerce')

# Extract decade from the year
df['decade'] = (df['date_x'].dt.year // 10) * 10
```

```
# Drop rows where decade is NaN
df = df.dropna(subset=['decade'])

# Explode the 'genre' column (split multiple genres)
df['genre'] = df['genre'].astype(str)  # Ensure it's a string before
splitting
df_exploded =
df.assign(genre=df['genre'].str.split(',')).explode('genre')

# Trim whitespace from genre names
df_exploded['genre'] = df_exploded['genre'].str.strip()

# Count genres per decade
genre_counts = df_exploded.groupby(['decade',
'genre']).size().reset_index(name='count')

# Get the most frequent genre per decade
top_genres_per_decade =
genre_counts.loc[genre_counts.groupby('decade')['count'].idxmax()]

# Plot the results
plt.figure(figsize=(12, 6))
sns.barplot(data=top_genres_per_decade, x='decade', y='count',
hue='decade', palette='viridis')
plt.xlabel("Decade")
plt.ylabel("Number of Movies")
plt.title("Most Popular Genres by Decade")
plt.xticks(rotation=45)
plt.show()
```



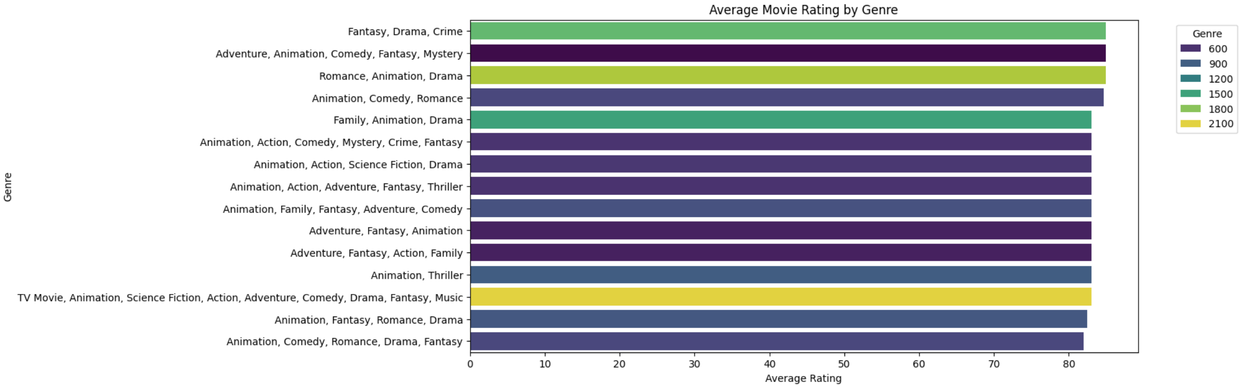
```
# Plot a heatmap or pairplot to examine relationships between budget,
revenue, scores.

#Heatmap of correlations
plt.figure(figsize=(8,6))
sns.heatmap(df[['budget_x', 'revenue', 'score']].corr(), annot=True,
cmap='coolwarm')
plt.title('Correlation Heatmap of Budget, Revenue, and Score')
plt.show()

# Pairplot to examine relationships
sns.pairplot(df[['budget_x', 'revenue', 'score']])
plt.show()
```


Correlation Heatmap of Budget, Revenue, and Score
budget_x
revenue
score
1
0.67
-0.24
0.67
1
0.097
-0.24
0.097
1
budget_x
revenue
score
1.0
0.8
0.6
0.4
0.2
0.0
−0.2



```
#Are there specific genres or release years with higher-rated movies?
Group by genre and year, then analyze the average rating.
#Group by genre and year, then analyze the average rating
plt.figure(figsize=(12, 6))
genre_avg_rating = df.groupby('genre')['score'].mean().reset_index()
genre_avg_rating = genre_avg_rating.sort_values(by='score',
ascending=False)
genre_avg_rating = genre_avg_rating.head(15)

#Plot the results
sns.barplot(data=genre_avg_rating, x='score',
y='genre',hue=genre_avg_rating.index,  palette='viridis')
```

```
plt.xlabel("Average Rating")
plt.ylabel("Genre")
plt.legend(title = 'Genre', bbox_to_anchor = (1.05, 1), loc = 'upper left')
plt.title("Average Movie Rating by Genre")
plt.show()
```



9. Insights and Summary

Task: Summarize key findings.

1. Movie Releases Trend

- Over time, the number of movies released has grown **significantly**, meaning more films are being produced now than in the past.

- The highest number of movie releases happened in **2022**, showing a peak in production.

- The distribution of releases is **right-skewed**, which means earlier years had fewer releases, and the number has increased rapidly in recent decades.

- This trend suggests that **technological advancements, streaming platforms, and increased global interest** in filmmaking have contributed to more movies being made.

2. Popular Genres & Ratings

- **Drama** is the **most common genre** across all movies, followed by **Action and Adventure**.

- However, even though Drama is the most frequent, movies in the **Fantasy, Animation, and Comedy** genres tend to receive **higher IMDB ratings (85 on average)**.

- This means that while Drama movies are made the most, **audiences and critics rate Fantasy and Animation movies higher** on average.

- One possible reason is that **animated and fantasy films often have better storytelling, visual appeal, and emotional impact**, leading to higher ratings.

---

3. Budget vs. Revenue

- Movies with **higher budgets tend to earn more money** at the box office.

- The **correlation is 0.67**, which is **a strong positive relationship**, meaning that **spending more on production usually results in higher revenue**.

- However, this isn't a **perfect correlation (1.0),** meaning some low-budget films can still perform well, and some expensive films can fail.

- Other factors like **marketing, genre, and audience engagement** also influence how much revenue a movie makes.

---

4. Trends Over Decades

- In **earlier decades, Drama was the most dominant genre**, meaning most movies focused on **serious storytelling and real-life themes**.

- In recent decades, **Action, Sci-Fi, and Fantasy genres have gained popularity**, likely due to advancements in CGI, special effects, and growing audience demand for visually engaging films.

- When looking at the **highest-rated movies**, they mostly belong to the **Fantasy, Animation, and Drama** genres.

- This suggests that **people generally enjoy imaginative and visually stunning films**, which receive better ratings compared to standard genres like Drama or Action.