Phase 1 Project

Project Overview

For this project, you will use exploratory data analysis to generate insights for a business stakeholder.

Business Problem

Microsoft sees all the big companies creating original video content and they want to get in on the fun. They have decided to create a new movie studio, but they don’t know anything about creating movies. You are charged with exploring what types of films are currently doing the best at the box office. You must then translate those findings into actionable insights that the head of Microsoft's new movie studio can use to help decide what type of films to create.

The Data

We are going to fetch information on **movie titles**, **box office ratings** and the **box office gross** from the follwing files.

- imdb.title.basics
- imdb.title.ratings
- bom.movie_gross

Data Gathering:

Load the data from the provided CSV files into your Jupyter Notebook using pandas.

```
# Importing necessary libraries
import pandas as pd
import numpy as np
```

```
boxoffice_gross = pd.read_csv('/content/bom.movie_gross.csv')
ratings = pd.read_csv('/content/title.ratings.csv')
title = pd.read_csv('/content/title.basics.csv')
```

Let's have a look at the tables for each of our datasets below.

```
boxoffice_gross.head()
```

	title	studio	domestic_gross	foreign_gross	year
0	Toy Story 3	BV	415000000.0	652000000	2010
1	Alice in Wonderland (2010)	BV	334200000.0	691300000	2010
2	Harry Potter and the Deathly Hallows Part 1	WB	296000000.0	664300000	2010
3	Inception	WB	292600000.0	535700000	2010
4	Shrek Forever After	P/DW	238700000.0	513900000	2010

```
ratings.head()
```

	tconst	averagerating	numvotes
0	tt10356526	8.3	31
1	tt10384606	8.9	559
2	tt1042974	6.4	20
3	tt1043726	4.2	50352
4	tt1060240	6.5	21

```
title.head()
```

	tconst	primary_title	original_title	start_year	runtime_minutes	genres
0	tt0063540	Sunghursh	Sunghursh	2013	175.0	Action,Crime,Drama
1	tt0066787	One Day Before the Rainy Season	Ashad Ka Ek Din	2019	114.0	Biography,Drama
2	tt0069049	The Other Side of the Wind	The Other Side of the Wind	2018	122.0	Drama
3	tt0069204	Sabse Bada Sukh	Sabse Bada Sukh	2018	NaN	Comedy,Drama
4	tt0100275	The Wandering Soap Opera	La Telenovela Errante	2017	80.0	Comedy,Drama,Fantasy

Data Cleaning

Check for data redundancy including, duplicates, and missing values that might affect our data analysis stage. We do this for all of our datasets.

```
# Check for missing values
missing_values_bom = boxoffice_gross.isnull().sum()
print("Missing Values in bom.movie_gross.csv:")
print(missing_values_bom)

# Remove rows with missing values
missing_values_bom.dropna(inplace=True)

# Remove duplicates (if any)
boxoffice_gross.drop_duplicates(inplace=True)

# Display the cleaned dataset
```

```
print("\nCleaned bom.movie_gross.csv Dataset:")
display(boxoffice_gross.head())
```

Here's a snippet of our result:

```
Missing Values in bom.movie_gross.csv:
title                0
studio               5
domestic_gross      28
foreign_gross     1350
year                 0
dtype: int64
```

Merging Data

Here we will load other data sets first and clean before merging. Merging of relevant datasets is based on common columns (e.g., movie title or ID) to create a comprehensive dataset for analysis. Other datasets such as budgets to comapre the net profit across genres.

```
budget = pd.read_csv('/content/movie_budgets.csv')
```

```
budget.head()
```

	id	release_date	movie	production_budget	domestic_gross	worldwide_gross
0	1	Dec 18, 2009	Avatar	$425,000,000	$760,507,625	$2,776,345,279
1	2	May 20, 2011	Pirates of the Caribbean: On Stranger Tides	$410,600,000	$241,063,875	$1,045,663,875
2	3	Jun 7, 2019	Dark Phoenix	$350,000,000	$42,762,350	$149,762,350
3	4	May 1, 2015	Avengers: Age of Ultron	$330,600,000	$459,005,868	$1,403,013,963
4	5	Dec 15, 2017	Star Wars Ep. VIII: The Last Jedi	$317,000,000	$620,181,382	$1,316,721,747

We clean the data as shown in the data cleaning step.

Gross Box office revenue information table. We'll merge all data from our four tables, boxoffice_gross, title, ratings, and budget to give us a clearer picture of which movies performed well at the box office. We'll call this table **movies_dataset**.

```
# Merging relevant datasets based on common identifiers
movies_data = pd.merge(boxoffice_gross, title, how='inner', left_on='title', right_on='primary_title')
movies_data = pd.merge(movies_data, ratings, how='inner', on='tconst')

movies_data.info()


# Merging budget table to our current table
movies_dataset = pd.merge(movies_data, budget, left_on='title', right_on='movie')

movies_dataset.info()
```

We clean our merged table and print out our table using:

```
# Check for missing values
print("Missing values in movies_dataset:")
print(movies_dataset.isnull().sum())

# Remove rows with missing values
movies_dataset.dropna(inplace=True)

# Display the cleaned dataset
print("\nCleaned Movies Dataset:")
display(movies_dataset.head())
```

```
Missing values in movies_dataset:
title                 0
studio                0
domestic_gross_x      1
foreign_gross       198
year                  0
tconst                0
primary_title         0
original_title        0
start_year            0
runtime_minutes      30
genres                7
averagerating         0
numvotes              0
id                    0
release_date          0
movie                 0
production_budget     0
domestic_gross_y      0
worldwide_gross       0
dtype: int64

Cleaned Movies Dataset:
```

	title	studio	domestic_gross_x	foreign_gross	year	tconst	primary_title	original_title	start_year	runtime_minutes	
0	Toy Story 3	BV	415000000.0	652000000	2010	tt0435761	Toy Story 3	Toy Story 3	2010	103.0	Adventur
1	Inception	WB	292600000.0	535700000	2010	tt1375666	Inception	Inception	2010	148.0	Ac
2	Shrek Forever After	P/DW	238700000.0	513900000	2010	tt0892791	Shrek Forever After	Shrek Forever After	2010	93.0	Adventur
3	The Twilight Saga: Eclipse	Sum.	300500000.0	398000000	2010	tt1325004	The Twilight Saga: Eclipse	The Twilight Saga: Eclipse	2010	124.0	Adve
4	Iron Man 2	Par.	312400000.0	311500000	2010	tt1228705	Iron Man 2	Iron Man 2	2010	124.0	Ac

As we can see there are duplicate columns with similar values thus we'll remove these columns in a final data cleaning step:

```
# Create a copy of our movies_dataset
movies_dataset_copy = movies_dataset.copy()

# Identifying redundant columns in for domestic_gross
print(movies_dataset_copy[['domestic_gross_x', 'domestic_gross_y']].head())

# Drop the redundant column (domestic_gross_y')
movies_dataset_copy.drop(columns=['domestic_gross_y'], inplace=True)

# Rename the remaining column if necessary
movies_dataset_copy.rename(columns={'domestic_gross_x': 'domestic_gross'}, inplace=True)

# Dropping redundant title columns
movies_dataset_copy.drop(columns=['primary_title', 'original_title', 'movie'], inplace=True)

# Dropping redundant year columns
movies_dataset_copy.drop(columns=['start_year', 'year'], inplace=True)

# Now, you can proceed with the analysis or any further processing
print("\nRemoved redundant Columns in Movies Dataset:")
display(movies_dataset_copy.head())
```

```
   domestic_gross_x domestic_gross_y
0       415000000.0     $415,004,880
1       292600000.0     $292,576,195
2       238700000.0     $238,736,787
3       300500000.0     $300,531,751
4       312400000.0     $312,433,331

Removed redundant Columns in Movies Dataset:
```

	title	studio	domestic_gross	foreign_gross	tconst	runtime_minutes	genres	averagerating	numvotes	id	rel
0	Toy Story 3	BV	415000000.0	652000000	tt0435761	103.0	Adventure,Animation,Comedy	8.3	682218	47	Ju
1	Inception	WB	292600000.0	535700000	tt1375666	148.0	Action,Adventure,Sci-Fi	8.8	1841066	38	J
2	Shrek Forever After	P/DW	238700000.0	513900000	tt0892791	93.0	Adventure,Animation,Comedy	6.3	167532	27	Ma
3	The Twilight Saga: Eclipse	Sum.	300500000.0	398000000	tt1325004	124.0	Adventure,Drama,Fantasy	5.0	211733	53	Ju
4	Iron Man 2	Par.	312400000.0	311500000	tt1228705	124.0	Action,Adventure,Sci-Fi	7.0	657690	15	M

To preserve and demonstrate our data cleaning proccess, we created a copy of the original `movies_dataset` table.

Some of our data was inconsistent with "production_data" and "worldwide_gross" still being in string format. Here we cleaned the data so it matches the format of other numeric/float data types.

```
# Clean the 'production_budget' and 'worldwide_gross' columns
movies_dataset_copy['production_budget'] = movies_dataset_copy['production_budget'].str.replace('$', '').str.replace(',', '').astype(float)
movies_dataset_copy['worldwide_gross'] = movies_dataset_copy['worldwide_gross'].str.replace('$', '').str.replace(',', '').astype(float)
# Clean the 'foreign_gross' column, remove commas, and convert to float
movies_dataset_copy['foreign_gross'] = movies_dataset_copy['foreign_gross'].str.replace(',', '').astype(float)
# Now, you can proceed with the data analysis
```

To prepare our data for analysis step we created a new column called "total_gross" where we add the values between the "domestic_gross" and "foreign_gross". While the "Worldwide_gross" column serves a similar function, there are inconsistencies between "total_gross" and "Worldwide_gross".

```
# Create new column 'total_gross' by adding 'domestic_gross' and 'worldwide_gross'
movies_dataset_copy['total_gross'] = movies_dataset_copy['domestic_gross'] + movies_dataset_copy['foreign_gross']

# Display the first few rows to verify the new column
print(movies_dataset_copy.head())
```

```
                        title studio  domestic_gross  foreign_gross  \
0                 Toy Story 3     BV     415000000.0    652000000.0
1                   Inception     WB     292600000.0    535700000.0
2         Shrek Forever After   P/DW     238700000.0    513900000.0
3  The Twilight Saga: Eclipse   Sum.     300500000.0    398000000.0
4                  Iron Man 2   Par.     312400000.0    311500000.0

      tconst  runtime_minutes                      genres  averagerating  \
0  tt0435761            103.0  Adventure,Animation,Comedy            8.3
1  tt1375666            148.0     Action,Adventure,Sci-Fi            8.8
2  tt0892791             93.0  Adventure,Animation,Comedy            6.3
3  tt1325004            124.0     Adventure,Drama,Fantasy            5.0
4  tt1228705            124.0     Action,Adventure,Sci-Fi            7.0

   numvotes  id  release_date  production_budget  worldwide_gross  \
0    682218  47  Jun 18, 2010        200000000.0     1.068880e+09
1   1841066  38  Jul 16, 2010        160000000.0     8.355246e+08
2    167532  27  May 21, 2010        165000000.0     7.562447e+08
3    211733  53  Jun 30, 2010         68000000.0     7.061028e+08
4    657690  15   May 7, 2010        170000000.0     6.211564e+08

    total_gross
0  1.067000e+09
1  8.283000e+08
```

```
     2  7.526000e+08
     3  6.985000e+08
     4  6.239000e+08
```

Our other column is the "net_revenue" where we get the actual revenue generated after the "production_budget" of a movie has been subtracted from our "total_gross".

```
# Create new column 'net_revenue' by subtracting 'total_gross' and 'production_budget'
movies_dataset_copy['net_revenue'] = movies_dataset_copy['total_gross'] - movies_dataset_copy['production_budget']

# Display the first few rows to verify the new column
print(movies_dataset_copy.head())
```

```
                          title studio  domestic_gross  foreign_gross  \
     0                 Toy Story 3     BV     415000000.0    652000000.0
     1                   Inception     WB     292600000.0    535700000.0
     2          Shrek Forever After   P/DW     238700000.0    513900000.0
     3  The Twilight Saga: Eclipse   Sum.     300500000.0    398000000.0
     4                  Iron Man 2   Par.     312400000.0    311500000.0

           tconst  runtime_minutes                      genres  averagerating  \
     0  tt0435761            103.0  Adventure,Animation,Comedy            8.3
     1  tt1375666            148.0     Action,Adventure,Sci-Fi            8.8
     2  tt0892791             93.0  Adventure,Animation,Comedy            6.3
     3  tt1325004            124.0     Adventure,Drama,Fantasy            5.0
     4  tt1228705            124.0     Action,Adventure,Sci-Fi            7.0

        numvotes  id  release_date  production_budget  worldwide_gross  \
     0    682218  47  Jun 18, 2010        200000000.0     1.068880e+09
     1   1841066  38  Jul 16, 2010        160000000.0     8.355246e+08
     2    167532  27  May 21, 2010        165000000.0     7.562447e+08
     3    211733  53  Jun 30, 2010         68000000.0     7.061028e+08
     4    657690  15   May 7, 2010        170000000.0     6.211564e+08

         total_gross  net_revenue
     0  1.067000e+09  867000000.0
     1  8.283000e+08  668300000.0
     2  7.526000e+08  587600000.0
     3  6.985000e+08  630500000.0
     4  6.239000e+08  453900000.0
```

Data Analysis

From our merged data we'll look at the performance at the box office across genres, ratings, and estimated budgets. Our Analysis will cover the following points of interest:

- revenue generated by each movie genre.
- influence of ratings on gross earnings.
- whether release dates affect our gross earnings.

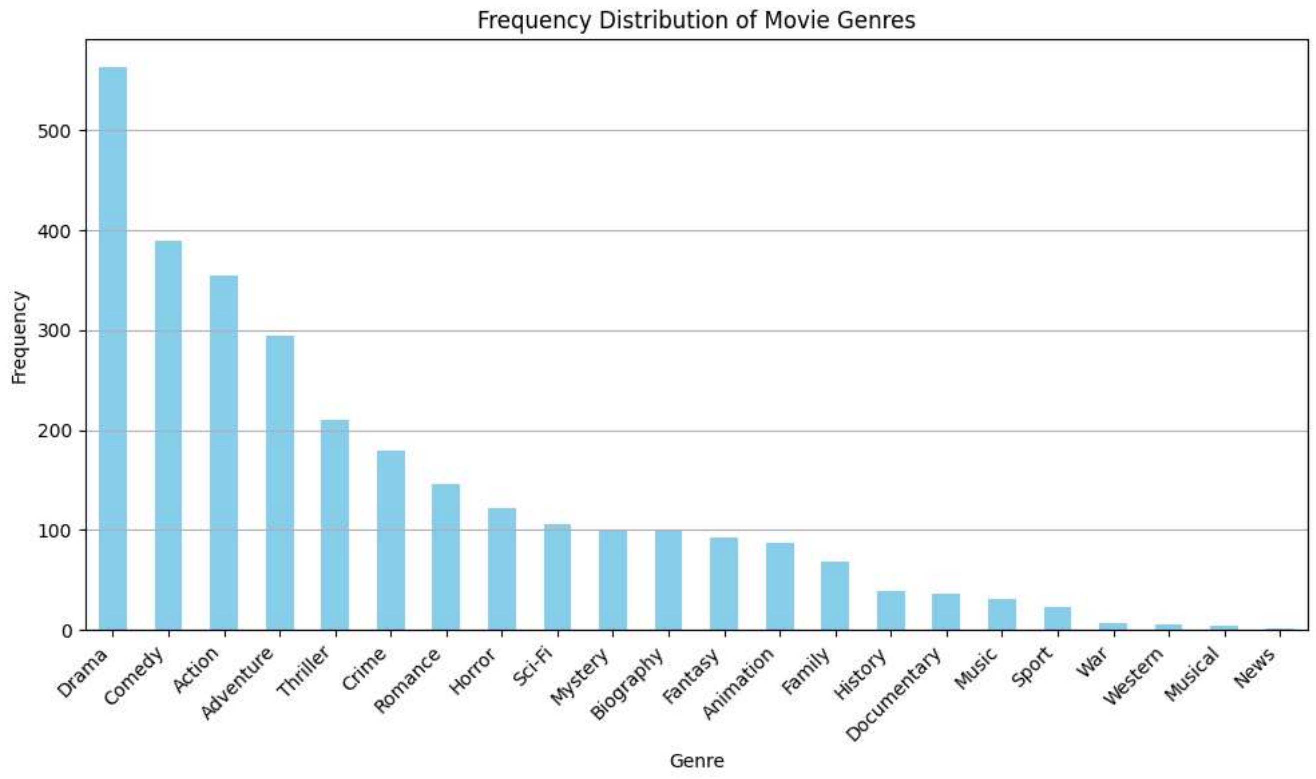
Analysis by Genre

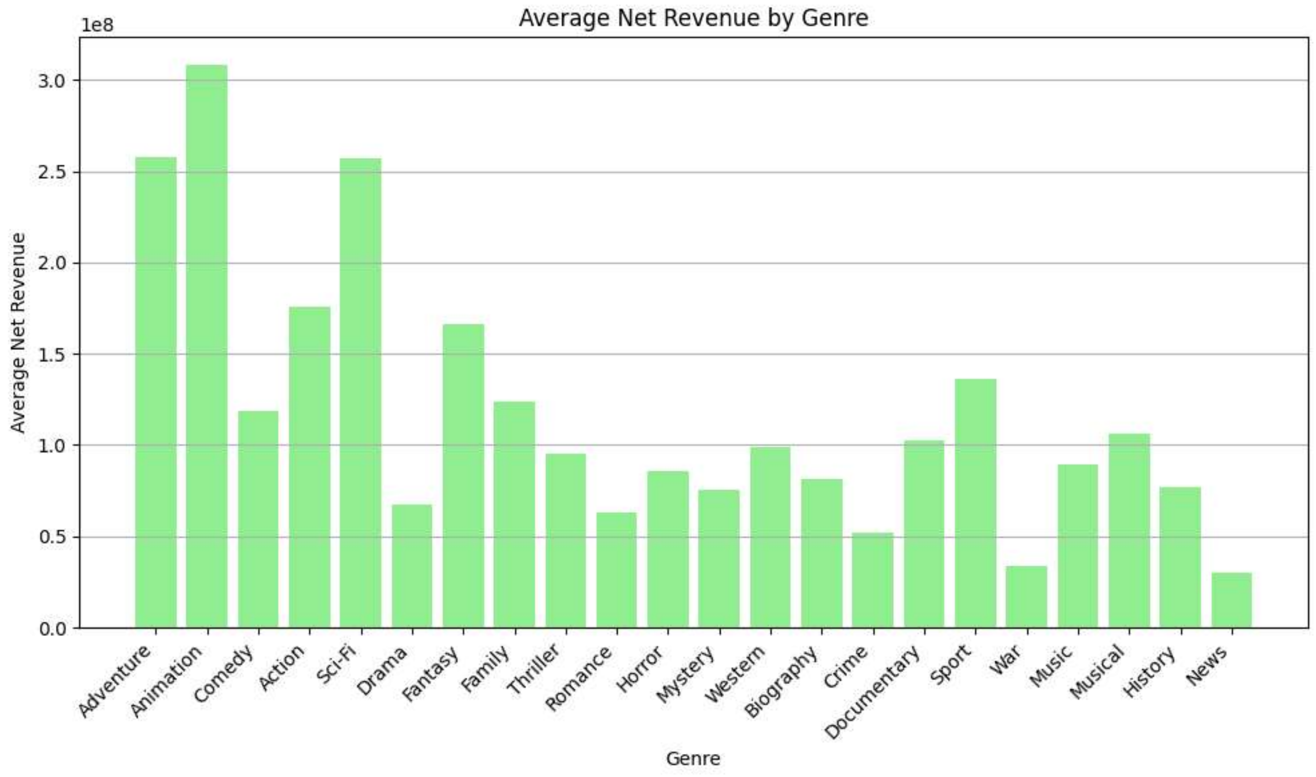
Genre Analysis: Analyze the distribution of movie genres and identify the most popular genres. You can calculate the frequency of each genre and visualize it using bar plots.

Genre Trends Over Time: Analyze how the popularity of different genres has changed over time. You can calculate the frequency of each genre for different time periods and visualize it using line plots or stacked bar plots.

Revenue by Genre: Analyze the relationship between net revenue for different genres. You can calculate average revenue and budget for each genre and visualize it using grouped bar plots or box plots.

```
import matplotlib.pyplot as plt
import seaborn as sns
```





Analysis by Rating

Ratings give us insight into how audience and critics' opinion of a movie affected it's performance in revenue.

Rating Analysis: Explore the distribution of movie ratings (averagerating) and identify any trends or patterns. You can create histograms or box plots to visualize the distribution.

Rating vs. Revenue: Analyze the relationship between movie ratings and revenue. You can calculate average revenue for different rating categories and visualize it using box plots or scatter plots.

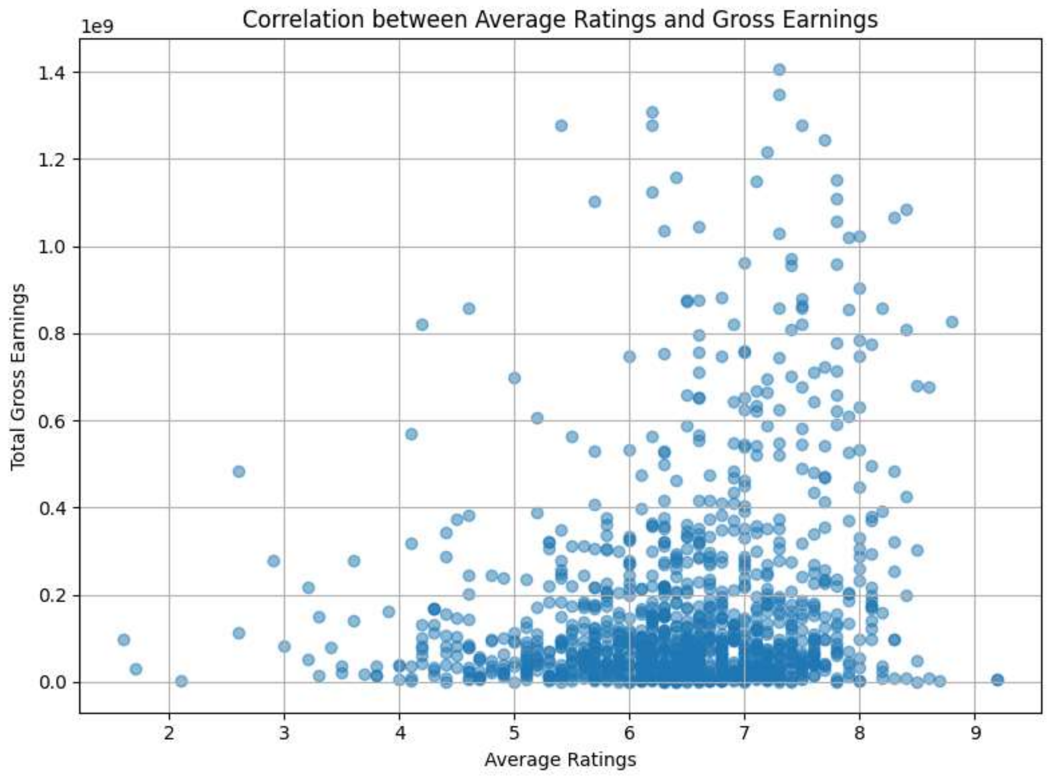
```
# Correlation analysis between ratings and gross earnings
correlation = movies_dataset_copy['averagerating'].corr(movies_dataset_copy['total_gross'])

print(f"Correlation between average ratings and gross earnings: {correlation}")
```

```
    Correlation between average ratings and gross earnings: 0.21234671196360394
```

The correlation coefficient between average ratings and gross earnings is approximately 0.21.

In simple terms, this positive correlation indicates that there is a weak to moderate relationship between the average ratings of movies and their gross earnings.

This means that the audience and critics' rating of a movie does not necessarily affect it's revenue. However, it's important to note that the correlation is not very strong, suggesting that other factors besides ratings also play a role in determining the financial success of a movie.



Analysis by Release Date

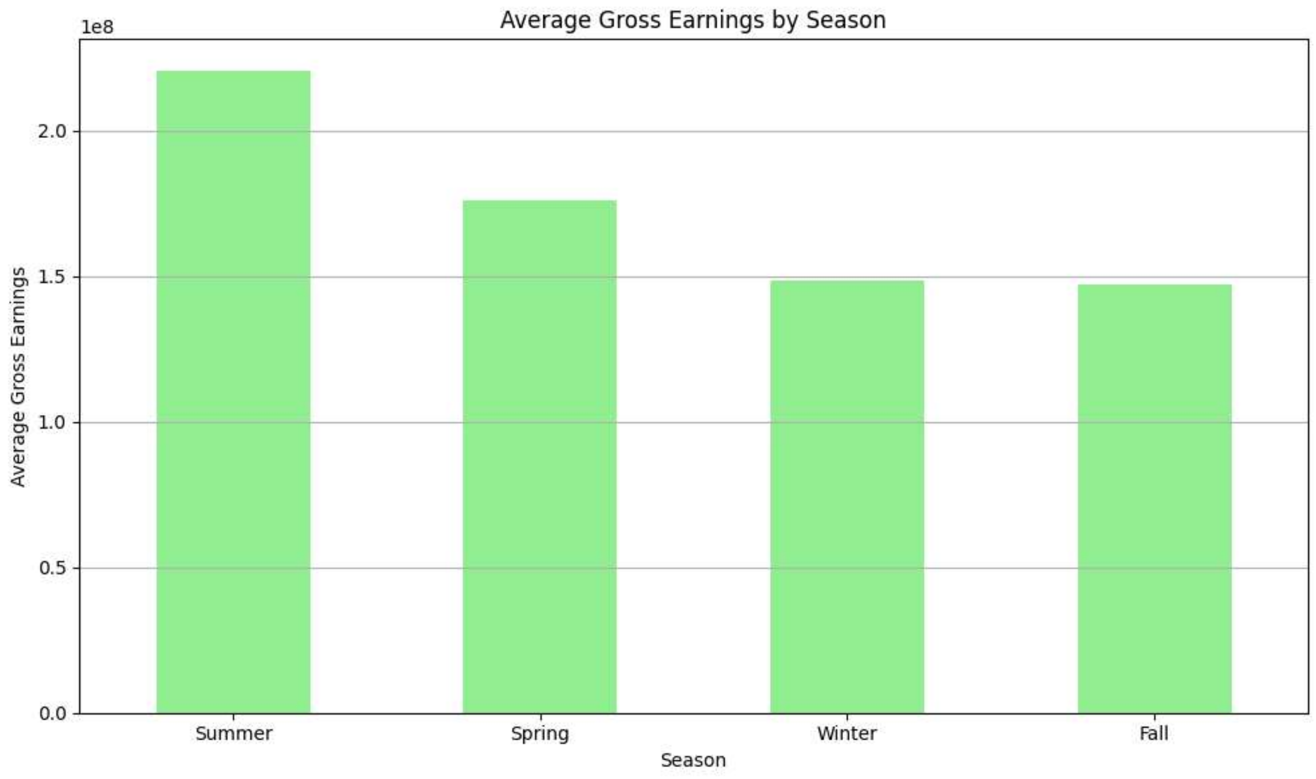
Release Date Analysis: Analyze the distribution of movie releases over time and identify any seasonal trends. You can group movies by release year or month and visualize the distribution using line plots or bar plots.

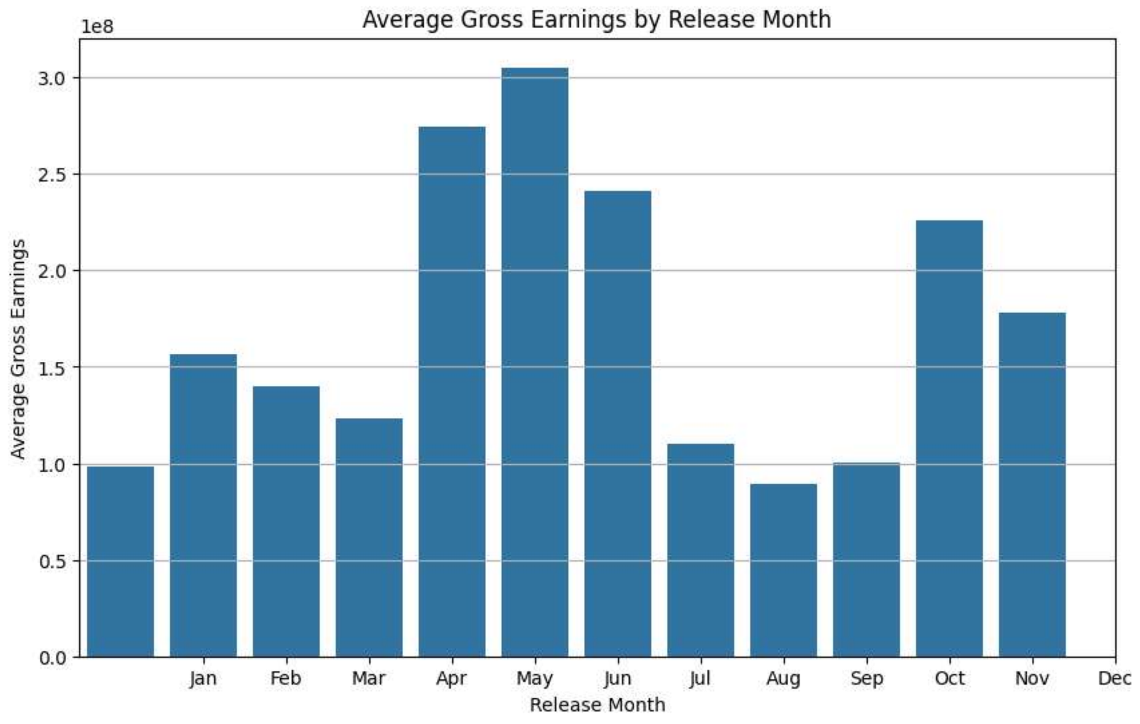
Budget vs. Revenue Trends Over Time: Analyze how the relationship between production budget and revenue has changed over time. You can calculate average budget and revenue for different time periods and visualize it using line plots.

```
# Perform ANOVA test
import scipy.stats as stats

# Group the data by release month
grouped_by_date = movies_dataset_copy.groupby('release_date')['total_gross'].apply(list)

# Perform ANOVA test
anova_result = stats.f_oneway(*grouped_by_date)

print("ANOVA Test Result:")
print(anova_result)
```

```
    ANOVA Test Result:
    F_onewayResult(statistic=1.971510556756346, pvalue=1.0469216871194916e-16)
```

This indicates that there is a statistically significant difference in gross earnings among movies released in different months. The extremely low p-value (1.05e-16) suggests that the differences in gross earnings among months are highly unlikely to be due to random chance alone. Therefore, we can conclude that release dates have a significant impact on gross earnings. However, it's important to perform further analysis or consider other factors to understand the specific patterns or trends associated with release dates and gross earnings.





The most likely months that seem to make more money are the spring and Summer months between April and June. This by no means that these months are the best times to release films as we see strong revenue generation in October and November.

Conclusions

Genre Popularity and Revenue Generation: Adventure, Animation, Action, and Sci-Fi are identified as the most popular genres based on the frequency of movie releases. These genres are also among the top revenue generators, indicating that they have a strong appeal to audiences and can translate into higher box office earnings.

Influence of Ratings on Gross Earnings: The correlation coefficient between average ratings and gross earnings is approximately 0.21, indicating a weak to moderate positive correlation. This suggests that while there is some relationship between movie ratings and gross earnings, it is not very strong. Other factors beyond ratings also contribute to a movie's financial success.

Effect of Release Dates on Gross Earnings: There is a statistically significant difference in gross earnings among movies released in different months. The analysis suggests that release dates have a significant impact on gross earnings, with certain months showing higher revenue generation than others. Spring and summer months, particularly April to June, appear to be the most profitable for movie releases, but strong revenue is also observed in October and November.

Implications and Further Analysis: Movie studios and filmmakers can use these insights to make strategic decisions regarding genre selection, release timing, and marketing efforts. While certain genres and release months show higher revenue potential, it's essential to conduct further analysis to understand specific trends and patterns comprehensively. Factors such as competition, audience demographics, marketing strategies, and the quality of the film itself can also influence box office performance and should be considered in decision-making processes.

In conclusion, while genre selection, ratings, and release timing play significant roles in determining box office success, a holistic approach considering various factors is necessary for maximizing revenue and audience satisfaction in the film industry.