TITLE: Decoding the Box Office

PROBLEM STATEMENT: To analyze a movie dataset to uncover insights into movie trends, financial success, and audience preferences, and to visualize these findings. The analysis will explore factors influencing movie performance and audience engagement, such as genre popularity, budget-revenue relationships, and ratings.

## Loading & Preparing Movie Dataset for Analysis

```
import pandas as pd
import numpy as np

# Load dataset
try:
    df = pd.read_csv('movie_dataset.csv')  # replace with your dataset file
except FileNotFoundError:
    print("Error: 'movie_dataset.csv' not found. Please make sure the file exists and the path is correct.")
    # You might need to upload the file to your Colab environment
    # or mount your Google Drive to access it.
    # For example: from google.colab import drive
    #             drive.mount('/content/drive')
    # Then adjust the file path accordingly.
except Exception as e:
    print(f"An error occurred while loading the dataset: {e}")

if 'df' in locals():  # Check if DataFrame was successfully created
    # Convert dates
    df['release_date'] = pd.to_datetime(df['release_date'], errors='coerce')
    df['year'] = df['release_date'].dt.year

    # Fill missing budgets/revenues with median or zero
    df['budget'] = df['budget'].fillna(0)
    df['revenue'] = df['revenue'].fillna(0)

    # Check if 'genres' column exists, if not, try 'genre'
    if 'genres' in df.columns:
        # Clean genres (assuming stringified lists)
        df['genres'] = df['genres'].astype(str).str.replace('[^a-zA-Z| ]', '')
    elif 'genre' in df.columns:
        # Clean genres (assuming stringified lists)
        df['genre'] = df['genre'].astype(str).str.replace('[^a-zA-Z| ]', '')
    else:
        print("Neither 'genre' nor 'genres' column found in the DataFrame.")

    # Display the first few rows of the DataFrame
    display(df.head())
```

| | index | budget | genres | homepage | id | keywords | original_language | original_title | overview |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 237000000 | Action Adventure Fantasy Science Fiction | http://www.avatarmovie.com/ | 19995 | culture clash future space war space colony so... | en | Avatar | In the 22nd century, a paraplegic Marine is di... |
| **1** | 1 | 300000000 | Adventure Fantasy Action | http://disney.go.com/disneypictures/pirates/ | 285 | ocean drug abuse exotic island east india trad... | en | Pirates of the Caribbean: At World's End | Captain Barbossa, long believed to be dead, ha... |
| **2** | 2 | 245000000 | Action Adventure Crime | http://www.sonypictures.com/movies/spectre/ | 206647 | spy based on novel secret agent sequel mi6 | en | Spectre | A cryptic message from Bond’s past sends him o... |
| **3** | 3 | 250000000 | Action Crime Drama Thriller | http://www.thedarkknightrises.com/ | 49026 | dc comics crime fighter terrorist secret ident... | en | The Dark Knight Rises | Following the death of District Attorney Harve... |
| **4** | 4 | 260000000 | Action Adventure Science Fiction | http://movies.disney.com/john-carter | 49529 | based on novel mars medallion space travel pri... | en | John Carter | John Carter is a war-weary, former military ca... |

5 rows × 25 columns

## New Section

## Dataset Overview & Top Genres Exploration

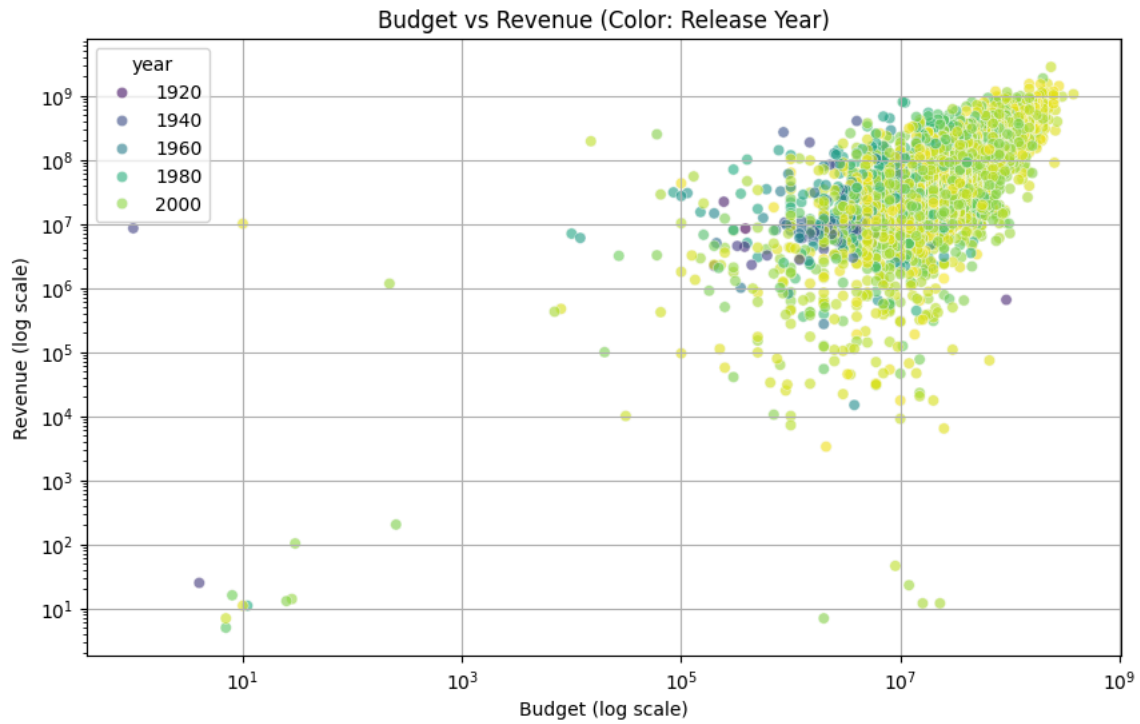
```
if 'df' in locals() and (('genres' in df.columns) or ('genre' in df.columns)):
    genre_col = 'genres' if 'genres' in df.columns else 'genre'
    print(df.describe())
    print(df.info())
    print(df[genre_col].value_counts().head(10))
else:
    print("DataFrame or genre column not found.")
```

```
             index        budget             id   popularity  \
count  4803.000000  4.803000e+03    4803.000000  4803.000000
mean   2401.000000  2.904504e+07   57165.484281    21.492301
min       0.000000  0.000000e+00       5.000000     0.000000
25%    1200.500000  7.900000e+05    9014.500000     4.668070
50%    2401.000000  1.500000e+07   14629.000000    12.921594
75%    3601.500000  4.000000e+07   58610.500000    28.313505
max    4802.000000  3.800000e+08  459488.000000   875.581305
std    1386.651002  4.072239e+07   88694.614033    31.816650

                      release_date       revenue      runtime  vote_average  \
count                         4802  4.803000e+03  4801.000000   4803.000000
mean   2002-12-27 23:45:54.352353280  8.226064e+07   106.875859      6.092172
min              1916-09-04 00:00:00  0.000000e+00     0.000000      0.000000
25%              1999-07-14 00:00:00  0.000000e+00    94.000000      5.600000
50%              2005-10-03 00:00:00  1.917000e+07   103.000000      6.200000
75%              2011-02-16 00:00:00  9.291719e+07   118.000000      6.800000
max              2017-02-03 00:00:00  2.787965e+09   338.000000     10.000000
std                              NaN  1.628571e+08    22.611935      1.194612

         vote_count         year
count   4803.000000  4802.000000
mean     690.217989  2002.468763
min        0.000000  1916.000000
25%       54.000000  1999.000000
```

```
50%      235.000000  2005.000000
75%      737.000000  2011.000000
max    13752.000000  2017.000000
std     1234.585891    12.414354
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4803 entries, 0 to 4802
Data columns (total 25 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   index                 4803 non-null   int64
 1   budget                4803 non-null   int64
 2   genres                4803 non-null   object
 3   homepage              1712 non-null   object
 4   id                    4803 non-null   int64
 5   keywords              4391 non-null   object
 6   original_language     4803 non-null   object
 7   original_title        4803 non-null   object
 8   overview              4800 non-null   object
 9   popularity            4803 non-null   float64
 10  production_companies  4803 non-null   object
 11  production_countries  4803 non-null   object
 12  release_date          4802 non-null   datetime64[ns]
 13  revenue               4803 non-null   int64
 14  runtime               4801 non-null   float64
 15  spoken_languages      4803 non-null   object
 16  status                4803 non-null   object
 17  tagline               3959 non-null   object
 18  title                 4803 non-null   object
 19  vote_average          4803 non-null   float64
 20  vote_count            4803 non-null   int64
 21  cast                  4760 non-null   object
 22  crew                  4803 non-null   object
 23  director              4773 non-null   object
```

## Visualizing Budget vs Revenue Across Years

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

if 'df' in locals() and 'budget' in df.columns and 'revenue' in df.columns and 'year' in df.columns:
    plt.figure(figsize=(10, 6))
    sns.scatterplot(data=df, x='budget', y='revenue', hue='year', palette='viridis', alpha=0.6)
    plt.title('Budget vs Revenue (Color: Release Year)')
    plt.xscale('log')
    plt.yscale('log')
    plt.xlabel('Budget (log scale)')
    plt.ylabel('Revenue (log scale)')
    plt.grid(True)
    plt.show()
else:
    print("DataFrame or required columns not found for visualization.")
```



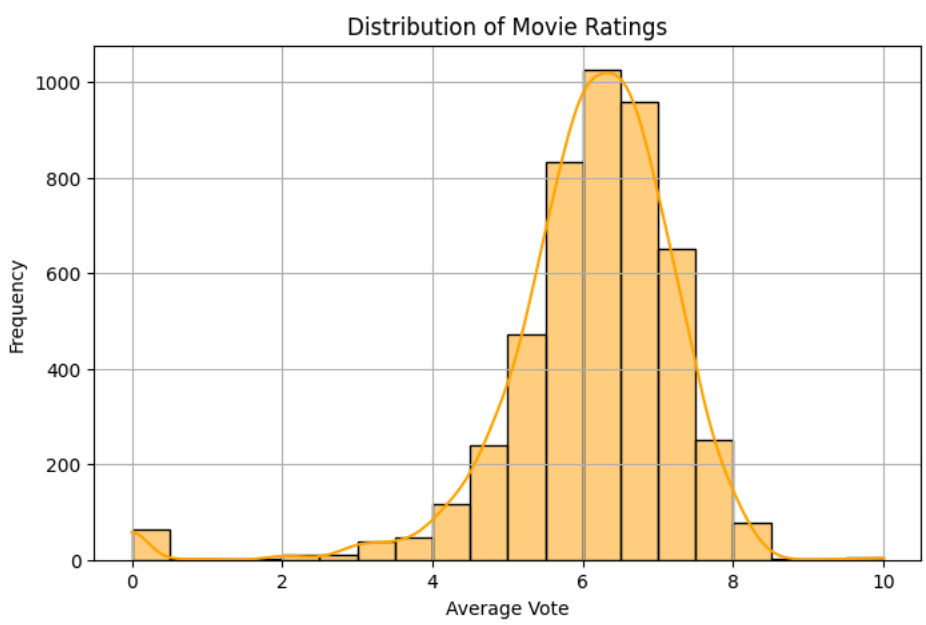
## Genre Trends Over Time

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

if 'df' in locals() and 'genre' in df.columns and 'year' in df.columns:
    top_genres = df['genre'].value_counts().head(10).index
    df_genres = df[df['genre'].isin(top_genres)]

    plt.figure(figsize=(12, 6))
    sns.countplot(data=df_genres, x='year', hue='genre')
    plt.xticks(rotation=45)
    plt.title('Genre Trends Over Time')
    plt.show()
else:
    print("DataFrame or required columns not found for genre trend visualization.")
```

```
DataFrame or required columns not found for genre trend visualization.
```

## Distribution of Movie Ratings

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

if 'df' in locals() and 'vote_average' in df.columns:
    plt.figure(figsize=(8, 5))
    sns.histplot(df['vote_average'], bins=20, kde=True, color='orange')
    plt.title('Distribution of Movie Ratings')
    plt.xlabel('Average Vote')
    plt.ylabel('Frequency')
    plt.grid(True)
    plt.show()
else:
    print("DataFrame or 'vote_average' column not found for rating distribution visualization.")
```




## Top 10 Most Profitable Movies

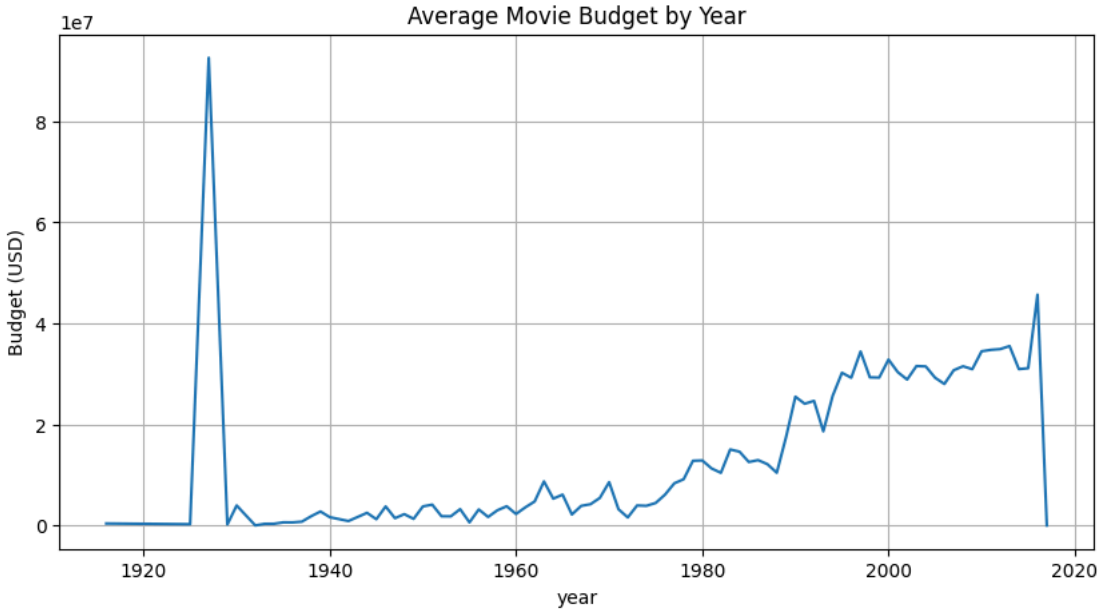
```
if 'df' in locals() and 'revenue' in df.columns and 'budget' in df.columns and 'title' in df.columns:
    df['profit'] = df['revenue'] - df['budget']
    top_profit = df[['title', 'budget', 'revenue', 'profit']].sort_values(by='profit', ascending=False).head(10)
    print(top_profit)  # Or display(top_profit) for a formatted output in Colab
else:
    print("DataFrame or required columns not found for profit calculation.")
```

```
                                      title     budget     revenue  \
0                                    Avatar  237000000  2787965087
25                                  Titanic  200000000  1845034188
28                           Jurassic World  150000000  1513528810
44                                Furious 7  190000000  1506249360
```

```
16                                The Avengers  220000000  1519557910
7                    Avengers: Age of Ultron  280000000  1405403694
124                                   Frozen  150000000  1274219009
546                                  Minions   74000000  1156730962
329  The Lord of the Rings: The Return of the King   94000000  1118888979
31                                  Iron Man 3  200000000  1215439994

         profit
0    2550965087
25   1645034188
28   1363528810
44   1316249360
16   1299557910
7    1125403694
124  1124219009
546  1082730962
329  1024888979
31   1015439994
```

## Average Movie Budget by Year

```
import matplotlib.pyplot as plt

if 'df' in locals() and 'year' in df.columns and 'budget' in df.columns:
    budget_year = df.groupby('year')['budget'].mean()
    plt.figure(figsize=(10, 5))
    budget_year.plot()
    plt.title('Average Movie Budget by Year')
    plt.ylabel('Budget (USD)')
    plt.grid(True)
    plt.show()
else:
    print("DataFrame or required columns not found for budget analysis.")
```




## Initial Dataset Overview

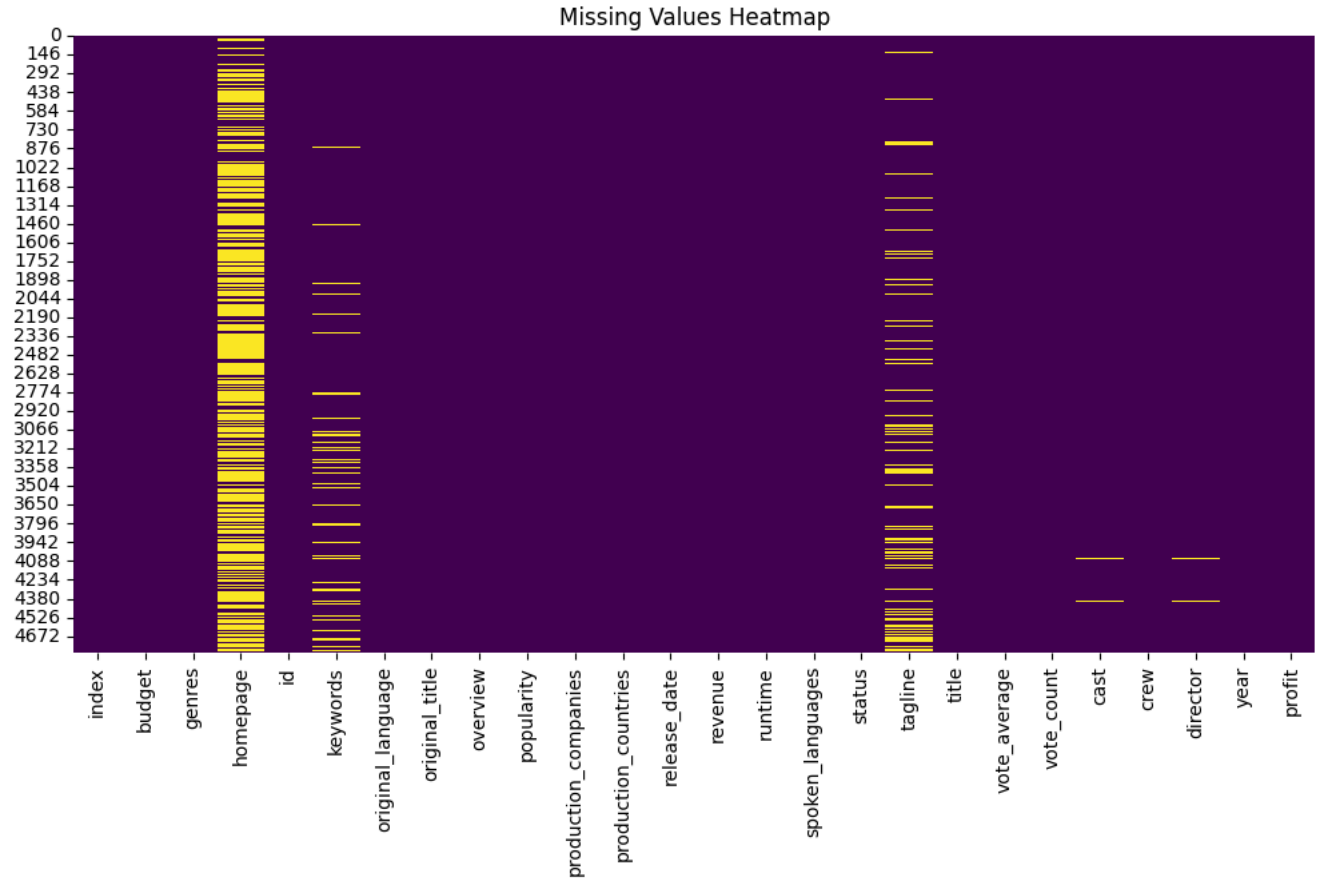
```
if 'df' in locals():
    print(df.head())
    print(df.shape)
    print(df.columns)
else:
    print("DataFrame not found.")
```

```
   index     budget                                   genres  \
0      0  237000000  Action Adventure Fantasy Science Fiction
1      1  300000000                   Adventure Fantasy Action
2      2  245000000                     Action Adventure Crime
3      3  250000000               Action Crime Drama Thriller
4      4  260000000          Action Adventure Science Fiction

                                      homepage      id  \
0                 http://www.avatarmovie.com/   19995
1  http://disney.go.com/disneypictures/pirates/     285
```

```
2   http://www.sonypictures.com/movies/spectre/  206647
3           http://www.thedarkknightrises.com/   49026
4         http://movies.disney.com/john-carter   49529

                                       keywords original_language  \
0  culture clash future space war space colony so...                en
1  ocean drug abuse exotic island east india trad...                en
2         spy based on novel secret agent sequel mi6                en
3  dc comics crime fighter terrorist secret ident...                en
4  based on novel mars medallion space travel pri...                en

                           original_title  \
0                                  Avatar
1  Pirates of the Caribbean: At World's End
2                                 Spectre
3                   The Dark Knight Rises
4                             John Carter

                                      overview  popularity  ...  \
0  In the 22nd century, a paraplegic Marine is di...  150.437577  ...
1  Captain Barbossa, long believed to be dead, ha...  139.082615  ...
2  A cryptic message from Bond’s past sends him o...  107.376788  ...
3  Following the death of District Attorney Harve...  112.312950  ...
4  John Carter is a war-weary, former military ca...   43.926995  ...

     status                                          tagline  \
0  Released                    Enter the World of Pandora.
1  Released  At the end of the world, the adventure begins.
2  Released                         A Plan No One Escapes
3  Released                               The Legend Ends
4  Released            Lost in our world, found in another.

                                  title  vote_average  vote_count  \
0                                Avatar           7.2       11800
1  Pirates of the Caribbean: At World's End           6.9        4500
2                               Spectre           6.3        4466
3                   The Dark Knight Rises           7.6        9106
4                             John Carter           6.1        2124

                                          cast  \
0  Sam Worthington Zoe Saldana Sigourney Weaver S...
1  Johnny Depp Orlando Bloom Keira Knightley Stel...
2  Daniel Craig Christoph Waltz L\u00e9a Seydoux ...
3  Christian Bale Michael Caine Gary Oldman Anne ...
4  Taylor Kitsch Lynn Collins Samantha Morton Wil...

                                          crew           director  \
0  [{'name': 'Stephen E. Rivkin', 'gender': 0, 'id...  James Cameron
```

## Missing Values Heatmap

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

if 'df' in locals():
    plt.figure(figsize=(12, 6))
    sns.heatmap(df.isnull(), cbar=False, cmap='viridis')
    plt.title("Missing Values Heatmap")
    plt.show()
else:
    print("DataFrame not found for missing values heatmap.")
```




## Top 10 Highest Budget Movies

```
if 'df' in locals() and 'title' in df.columns and 'budget' in df.columns:
    top_budget_movies = df[['title', 'budget']].sort_values(by='budget', ascending=False).head(10)
    print(top_budget_movies)  # Or display(top_budget_movies) for formatted output in Colab
else:
    print("DataFrame or required columns not found for budget analysis.")
```

```
                                      title     budget
17  Pirates of the Caribbean: On Stranger Tides  380000000
1      Pirates of the Caribbean: At World's End  300000000
7                       Avengers: Age of Ultron  280000000
10                             Superman Returns  270000000
6                                      Tangled  260000000
4                                  John Carter  260000000
5                                Spider-Man 3  258000000
13                            The Lone Ranger  255000000
9           Batman v Superman: Dawn of Justice  250000000
8       Harry Potter and the Half-Blood Prince  250000000
```

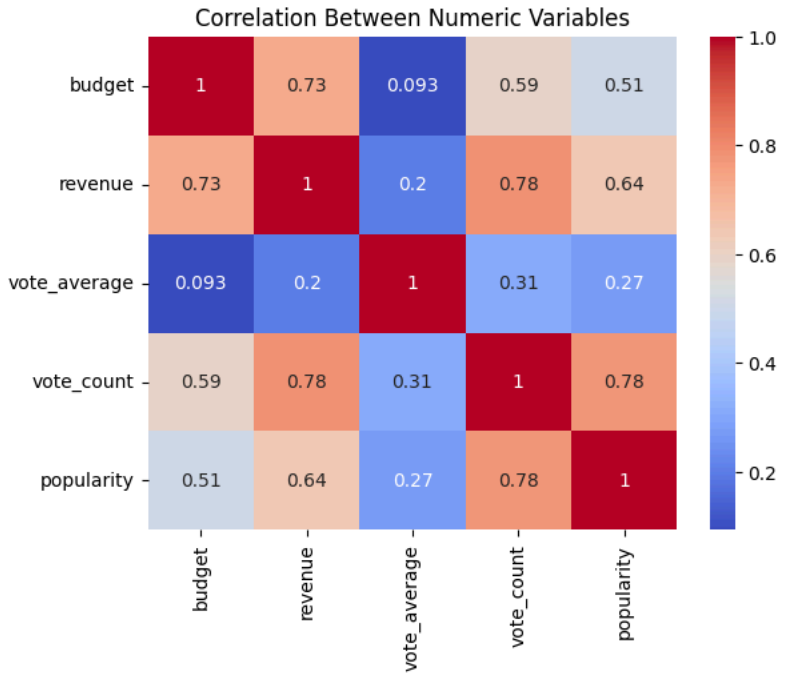
## Top 10 Highest Grossing Movies

```
if 'df' in locals() and 'title' in df.columns and 'revenue' in df.columns:
    top_revenue_movies = df[['title', 'revenue']].sort_values(by='revenue', ascending=False).head(10)
    print(top_revenue_movies)  # Or display(top_revenue_movies) for formatted output in Colab
else:
    print("DataFrame or required columns not found for revenue analysis.")
```

```
                         title     revenue
0                       Avatar  2787965087
25                     Titanic  1845034188
16                The Avengers  1519557910
28               Jurassic World  1513528810
44                   Furious 7  1506249360
7       Avengers: Age of Ultron  1405403694
124                      Frozen  1274219009
31                   Iron Man 3  1215439994
546                     Minions  1156730962
26   Captain America: Civil War  1153304495
```

## Correlation Between Key Numeric Features

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

if 'df' in locals() and all(col in df.columns for col in ['budget', 'revenue', 'vote_average', 'vote_count', 'popularity']):
    corr = df[['budget', 'revenue', 'vote_average', 'vote_count', 'popularity']].corr()
    sns.heatmap(corr, annot=True, cmap='coolwarm')
    plt.title('Correlation Between Numeric Variables')
    plt.show()
else:
    print("DataFrame or required columns not found for correlation analysis.")
```




## Low-Budget, Highly Rated Movies

```
if 'df' in locals() and all(col in df.columns for col in ['budget', 'vote_average', 'title']):
    filtered = df[(df['budget'] < 1_000_000) & (df['vote_average'] > 7.5)]
    print(filtered[['title', 'budget', 'vote_average']])
else:
    print("DataFrame or required columns not found for filtering.")
```

```
                                        title  budget  vote_average
463                                    Déjà Vu       0           8.0
1028                                   Solaris       0           7.7
2170                                   Psycho  806948           8.2
2386                          One Man's Hero       0           9.3
2796                The Prisoner of Zenda       0           8.4
2862                           About Time       0           7.8
3206                               Polisse       0           7.9
3208  Star Wars: Clone Wars: Volume 1       0           8.0
3503                        Lake of Fire       0           8.0
3510                                Emma       0           7.6
3519                    Stiff Upper Lips       0          10.0
3716                        Lilya 4-ever       0           7.7
3723                Anne of Green Gables       0           8.2
3931                      Days of Heaven       0           7.6
3989                Lage Raho Munna Bhai       0           7.6
3992                           Sardaarji       0           9.5
4045               Dancer, Texas Pop. 81       0          10.0
4093                   Touching the Void       0           7.6
4164                          The Square       0           7.8
4208                          The Bubble       0           7.6
4238                        Modern Times       1           8.1
4247               Me You and Five Bucks       1          10.0
4303                   The Second Mother       0           7.8
4329                          Casablanca  878000           7.9
4374               Dream with the Fishes       0           7.7
4403                      The Jimmy Show       0           8.0
4405                   Karachi se Lahore       0           8.0
4416                         Hidden Away       0           7.6
4432                   On the Waterfront  910000           8.0
4452                        A Separation  800000           7.7
```

```
4456                            Raising Victor Vargas  800000          7.8
4457                                  Pandora's Box       0          7.6
4459                                   Live-In Maid       0          7.8
4468                 Iraq for Sale: The War Profiteers       0          8.0
4471                      Kevin Hart: Laugh at My Pain       0          7.7
4474                                  The Conformist  750000          7.6
4479                                       High Noon  730000          7.6
4480                                    Hoop Dreams  700000          7.7
4571  Rise of the Entrepreneur: The Search for a Bet...       0          8.0
4579                   Monty Python and the Holy Grail  400000          7.8
4591                               Cries and Whispers       0          7.8
4602                                     12 Angry Men  350000          8.2
4603                                   My Dog Tulip       0          7.6
4604                            It Happened One Night  325000          7.7
4606                              Tupac: Resurrection  300000          8.0
4615                                   Pierrot le Fou  300000          7.6
4662                                   Little Big Top       0         10.0
4663                                Along the Roadside     250          7.7
4672                             A Fistful of Dollars  200000          7.6
4678                      The Business of Fancydancing  200000          8.0
4679                                 Call + Response       0          8.0
4685                      The Case of the Grinning Cat       0          7.7
4686                                             Ordet       0          7.8
4688                               The Man from Earth       0          7.7
4695                               Children of Heaven  180000          7.8
4755                                         Counting   50000          8.3
4766                                   The Last Waltz       0          7.9
```

## Top 10 Most Frequent Production Companies

```
if 'df' in locals() and 'production_companies' in df.columns:
    df['production_companies'] = df['production_companies'].astype(str)
    top_production_companies = df['production_companies'].str.split(',').explode().str.strip().value_counts().head(10)
    print(top_production_companies)
else:
    print("DataFrame or 'production_companies' column not found.")
```

```
production_companies
[]                                                  351
[{"name": "Paramount Pictures"                      281
[{"name": "Universal Pictures"                      260
{"name": "Warner Bros."                             254
"id": 33}                                           252
"id": 4}                                            223
[{"name": "Columbia Pictures"                       200
"id": 6194}                                         192
"id": 306}                                          182
[{"name": "Twentieth Century Fox Film Corporation"  177
Name: count, dtype: int64
```

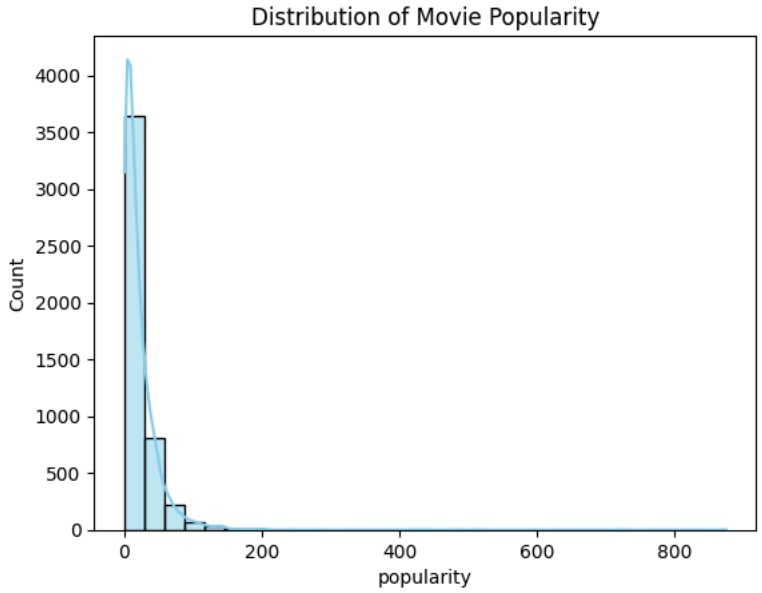
## Average Runtime by Genre

```
import matplotlib.pyplot as plt

if 'df' in locals() and 'genre' in df.columns and 'runtime' in df.columns:
    avg_runtime = df.groupby('genre')['runtime'].mean().sort_values(ascending=False)
    avg_runtime.plot(kind='bar', figsize=(10, 5), color='teal')
    plt.title('Average Runtime by Genre')
    plt.ylabel('Minutes')
    plt.show()
else:
    print("DataFrame or required columns not found for runtime analysis.")
```

```
DataFrame or required columns not found for runtime analysis.
```

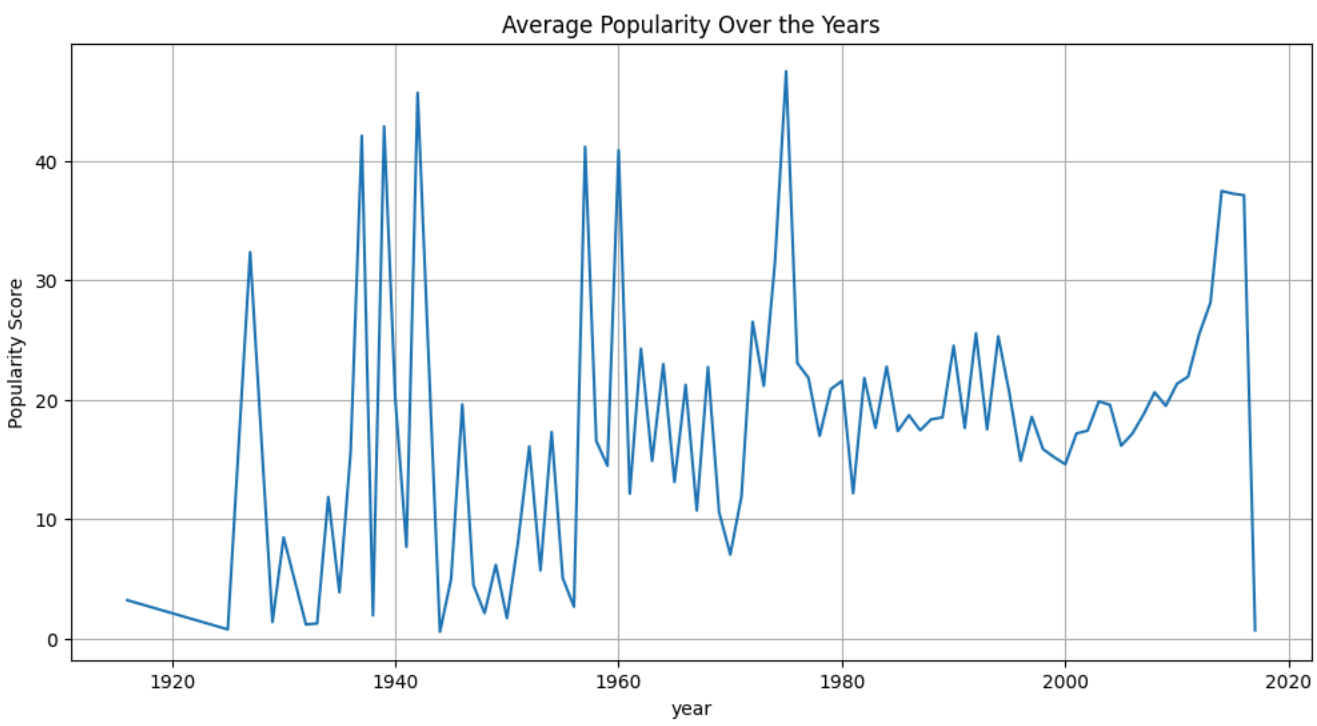
## Distribution of Movie Popularity Scores

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

if 'df' in locals() and 'popularity' in df.columns:
    sns.histplot(df['popularity'], bins=30, kde=True, color='skyblue')
    plt.title("Distribution of Movie Popularity")
    plt.show()
else:
    print("DataFrame or 'popularity' column not found for popularity distribution visualization.")
```




## Average Movie Popularity Over the Years

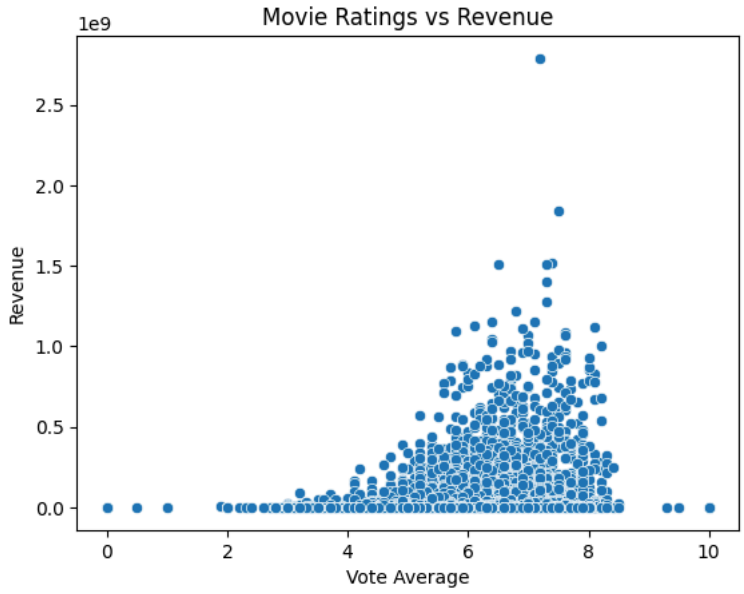
```
import matplotlib.pyplot as plt

if 'df' in locals() and 'year' in df.columns and 'popularity' in df.columns:
    df.groupby('year')['popularity'].mean().plot(figsize=(12, 6))
    plt.title("Average Popularity Over the Years")
    plt.ylabel("Popularity Score")
    plt.grid()
    plt.show()
else:
    print("DataFrame or required columns not found for popularity analysis.")
```




## Movie Ratings vs Revenue

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

if 'df' in locals() and 'vote_average' in df.columns and 'revenue' in df.columns:
    sns.scatterplot(data=df, x='vote_average', y='revenue')
    plt.title("Movie Ratings vs Revenue")
```

```
    plt.xlabel("Vote Average")
    plt.ylabel("Revenue")
    plt.show()
else:
    print("DataFrame or required columns not found for scatter plot visualization.")
```




## Profit Margin Calculation and Summary Statistics

```
import numpy as np

if 'df' in locals() and all(col in df.columns for col in ['revenue', 'budget']):
    df['profit'] = df['revenue'] - df['budget']
    df['profit_margin'] = (df['profit'] / df['budget']).replace([np.inf, -np.inf], np.nan)
    print(df['profit_margin'].describe())
else:
    print("DataFrame or required columns not found for profit margin calculation.")
```

```
count    3.766000e+03
mean     2.532490e+03
std      1.394602e+05
min     -1.000000e+00
25%     -5.090214e-01
50%      8.675333e-01
75%      2.941424e+00
max      8.499999e+06
Name: profit_margin, dtype: float64
```

## Genres Ranked by Average Revenue

```
import matplotlib.pyplot as plt

if 'df' in locals() and 'genre' in df.columns and 'revenue' in df.columns:
    genre_revenue = df.groupby('genre')['revenue'].mean().sort_values(ascending=False)
    genre_revenue.plot(kind='bar', figsize=(10, 5), color='coral')
    plt.title("Genres by Average Revenue")
    plt.ylabel("Average Revenue")
    plt.show()
else:
    print("DataFrame or required columns not found for genre revenue analysis.")
```

```
DataFrame or required columns not found for genre revenue analysis.
```

## Budget Range per Genre

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

if 'df' in locals() and 'genre' in df.columns and 'budget' in df.columns:
    plt.figure(figsize=(12, 6))
    sns.boxplot(x='genre', y='budget', data=df)
```

```
    plt.xticks(rotation=45)
    plt.title("Budget Range per Genre")
    plt.show()
else:
    print("DataFrame or required columns not found for box plot visualization."
```

```
DataFrame or required columns not found for box plot visualization.
```

## Revenue Distribution by Genre

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

if 'df' in locals() and 'genre' in df.columns and 'revenue' in df.columns:
    plt.figure(figsize=(12, 6))
    sns.violinplot(x='genre', y='revenue', data=df, scale='width', inner='quartile')
    plt.xticks(rotation=45)
    plt.title("Revenue Distribution by Genre")
    plt.show()
else:
    print("DataFrame or required columns not found for violin plot visualization.")
```

```
DataFrame or required columns not found for violin plot visualization.
```

## Top 10 Most Voted Movies

```
if 'df' in locals() and 'title' in df.columns and 'vote_count' in df.columns:
    top_voted_movies = df[['title', 'vote_count']].sort_values(by='vote_count', ascending=False).head(10)
    print(top_voted_movies)  # Or display(top_voted_movies) for formatted output in Colab
else:
    print("DataFrame or required columns not found for vote count analysis.")
```

```
                        title  vote_count
96                  Inception       13752
65            The Dark Knight       12002
0                      Avatar       11800
16              The Avengers       11776
788                 Deadpool       10995
95              Interstellar       10867
287         Django Unchained       10099
94   Guardians of the Galaxy        9742
426         The Hunger Games        9455
127       Mad Max: Fury Road        9427
```