

```
In [8]: import pandas as pd
import numpy as np
```

```
In [9]: csv_file_path = r'C:\Users\S1004830\Downloads\movie_dataset.csv'
```

```
In [10]: df = pd.read_csv(csv_file_path)
```

```
In [21]: df.isnull().sum().sum()
```

```
Out[21]: 192
```

```
In [22]: df.duplicated().sum()
```

```
Out[22]: 0
```

```
In [23]: highestRatedMovie = df.loc[df['Rating'].idxmax()]
```

```
In [24]: print("Highest-rated movie:")
print(highestRatedMovie)
```

```
Highest-rated movie:
Rank                                     55
Title                                The Dark Knight
Genre                                Action,Crime,Drama
Description    When the menace known as the Joker wreaks havo...
Director                                Christopher Nolan
Actors    Christian Bale, Heath Ledger, Aaron Eckhart,Mi...
Year                                     2008
Runtime (Minutes)                       152
Rating                                   9.0
Votes                                1791916
Revenue (Millions)                       533.32
Metascore                               82.0
Name: 54, dtype: object
```

```
In [28]: average_revenue = df['Revenue (Millions)'].mean()
print("Average Revenue of All Movies:", average_revenue)
```

```
Average Revenue of All Movies: 82.95637614678898
```

```
In [29]: filtered_df = df[(df['Year'] >= 2015) & (df['Year'] <= 2017)]
average_revenue_2015_2017 = filtered_df['Revenue (Millions)'].mean()

print("Average Revenue of Movies from 2015 to 2017:", average_revenue_2015_2017)
```

```
Average Revenue of Movies from 2015 to 2017: 63.099905660377345
```

```
In [30]: movies_2016 = df[df['Year'] == 2016]
count_movies_2016 = len(movies_2016)
print("Number of Movies Released in 2016:", count_movies_2016)
```

```
Number of Movies Released in 2016: 297
```

```
In [31]: nolan_movies = df[df['Director'] == 'Christopher Nolan']
count_nolan_movies = len(nolan_movies)
print("Number of Movies Directed by Christopher Nolan:", count_nolan_movies)
```

```
Number of Movies Directed by Christopher Nolan: 5
```

```
In [33]: highly_rated_movies = df[df['Rating'] >= 8.0]
count_highly_rated_movies = len(highly_rated_movies)
count_highly_rated_movies = len(highly_rated_movies)
print("Number of Movies with a Rating of at Least 8.0:", count_highly_rated_movies)
```

Number of Movies with a Rating of at Least 8.0: 78

```
In [34]: nolan_movies = df[df['Director'] == 'Christopher Nolan']
median_rating_nolan_movies = nolan_movies['Rating'].median()
print("Median Rating of Movies Directed by Christopher Nolan:", median_rating_nolan_movies)
```

Median Rating of Movies Directed by Christopher Nolan: 8.6

```
In [35]: average_rating_by_year = df.groupby('Year')['Rating'].mean()
year_highest_avg_rating = average_rating_by_year.idxmax()
highest_avg_rating = average_rating_by_year.max()
print("Year with the Highest Average Rating:", year_highest_avg_rating)
print("Highest Average Rating:", highest_avg_rating)
```

Year with the Highest Average Rating: 2007

Highest Average Rating: 7.133962264150944

```
In [37]: movies_2006 = df[df['Year'] == 2006]
movies_2016 = df[df['Year'] == 2016]
num_movies_2006 = len(movies_2006)
num_movies_2016 = len(movies_2016)
percentage_increase = ((num_movies_2016 - num_movies_2006) / num_movies_2006) * 100
print("Percentage Increase in Number of Movies (2006 to 2016):", percentage_increase)
```

Percentage Increase in Number of Movies (2006 to 2016): 575.0

```
In [38]: all_actors = df['Actors'].str.split(',').explode().str.strip()
most_common_actor = all_actors.mode().iloc[0]
print("Most Common Actor in All Movies:", most_common_actor)
```

Most Common Actor in All Movies: Mark Wahlberg

```
In [40]: all_genres = df['Genre'].str.split(',').explode().str.strip()
num_unique_genres = all_genres.nunique()
print("Number of Unique Genres:", num_unique_genres)
```

Number of Unique Genres: 20

```
In [41]: numerical_columns = ['Year', 'Runtime (Minutes)', 'Rating', 'Votes', 'Revenue (Millions)']
numerical_df = df[numerical_columns]
correlation_matrix = numerical_df.corr()
print("Correlation Matrix:")
print(correlation_matrix)
```

Correlation Matrix:

	Year	Runtime (Minutes)	Rating	Votes	\
Year	1.000000	-0.164900	-0.211219	-0.411904	
Runtime (Minutes)	-0.164900	1.000000	0.392214	0.407062	
Rating	-0.211219	0.392214	1.000000	0.511537	
Votes	-0.411904	0.407062	0.511537	1.000000	
Revenue (Millions)	-0.126790	0.267953	0.217654	0.639661	
Metascore	-0.079305	0.211978	0.631897	0.325684	

	Revenue (Millions)	Metascore
Year	-0.126790	-0.079305
Runtime (Minutes)	0.267953	0.211978
Rating	0.217654	0.631897
Votes	0.639661	0.325684
Revenue (Millions)	1.000000	0.142397
Metascore	0.142397	1.000000

In []: