i18l3u6b7

January 23, 2025

[1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```
data = pd.read_csv(r"C:\Users\91703\Downloads\IMDb_Dataset (2).csv")
```

[3]:
```
data.head()
```

[3]:

	Title	IMDb Rating	Year	Certificates	Genre	Director	Star Cast	MetaScore	Duration (minutes)
0	Gladiator	8.5	2000	R	Action	Ridley Scott	David FranzoniJohn LoganWilliam Nicholson	67.0	155.0
1	Mission: Impossible - Dead Reckoning Part One	7.7	2023	PG-13	Action	Christopher McQuarrie	Tom CruiseHayley AtwellVing Rhames	81.0	163.0
2	Rebel Moon - Part Two: The Scargiver	5.2	2024	PG-13	Action	Zack Snyder	Zack SnyderKurt JohnstadShay Hatten	35.0	122.0
3	Inglourious Basterds	8.4	2009	R	Adventure	Quentin Tarantino	Brad PittDiane KrugerEli Roth	69.0	153.0
4	Borderlands	7.1	2024	R	Action	Eli Roth	Eli RothJoe Crombie	66.9	116.3

[4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 9 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Title               400 non-null    object
```

```
 1   IMDb Rating          400 non-null    float64
 2   Year                 400 non-null    int64
 3   Certificates         400 non-null    object
 4   Genre                400 non-null    object
 5   Director             400 non-null    object
 6   Star Cast            400 non-null    object
 7   MetaScore            400 non-null    float64
 8   Duration (minutes)   400 non-null    float64
dtypes: float64(3), int64(1), object(5)
memory usage: 28.3+ KB
```

[5]:
```
data.describe()
```

[5]:
```
       IMDb Rating         Year   MetaScore  Duration (minutes)
count   400.000000   400.000000  400.000000          400.000000
mean      7.110250  2015.150000   66.898750          116.334750
std       0.932251    13.470194   15.312006           21.369757
min       3.900000  1939.000000   26.000000           80.000000
25%       6.700000  2010.750000   57.750000          101.000000
50%       7.100000  2023.000000   66.900000          116.000000
75%       7.700000  2024.000000   77.000000          124.000000
max       9.200000  2025.000000  100.000000          206.000000
```

[6]:
```
data.isnull()
```

[6]:
```
      Title  IMDb Rating   Year  Certificates  Genre  Director  Star Cast  \
0    False        False  False         False  False     False      False
1    False        False  False         False  False     False      False
2    False        False  False         False  False     False      False
3    False        False  False         False  False     False      False
4    False        False  False         False  False     False      False
..     …          …    …           …    …       …        …
395  False        False  False         False  False     False      False
396  False        False  False         False  False     False      False
397  False        False  False         False  False     False      False
398  False        False  False         False  False     False      False
399  False        False  False         False  False     False      False

     MetaScore  Duration (minutes)
0        False               False
1        False               False
2        False               False
3        False               False
4        False               False
..         …                 …
395      False               False
396      False               False
```

```
397      False                False
398      False                False
399      False                False

[400 rows x 9 columns]
```

[7]:
```
data.isnull().sum()
```

[7]:
```
Title                 0
IMDb Rating           0
Year                  0
Certificates          0
Genre                 0
Director              0
Star Cast             0
MetaScore             0
Duration (minutes)    0
dtype: int64
```

[9]:
```
#What is the average IMDb Rating of all movies in this dataset?
average_imdb_rating = data['IMDb Rating'].mean()
print("Average IMDb Rating:", average_imdb_rating)
```

```
Average IMDb Rating: 7.11025
```

[12]:
```
#What is the total number of movies released in the year 2020 in this dataset?
total_movies_in_2020 = len(data[data['Year'] == 2024])
print(total_movies_in_2020)
```

```
143
```

[14]:
```
#What is the most common Genre in this dataset?
most_common_genre = data['Genre'].value_counts().index[0]
print("Most Common Genre:", most_common_genre)
```

```
Most Common Genre: Action
```

[15]:
```
#Which Director has directed the most movies in this dataset?
most_active_director = data['Director'].value_counts().index[0]
print("Most Active Director:", most_active_director)
```

```
Most Active Director: George Miller
```

[21]:
```
#What is the average Duration (minutes) of movies in this dataset?
average_duration = data['Duration (minutes)'].mean()
print("Average Duration:", average_duration)
```

```
Average Duration: 116.33474999999999
```

[23]:
```
#What is the highest MetaScore in this dataset?
highest_metascroe = data['MetaScore'].max()
print("Highest MetaScore:", highest_metascroe)
```

```
Highest MetaScore: 100.0
```

[25]:
```
#How many movies have a Certificate of 'R' in this dataset?
total_r_certified = len(data[data['Certificates'] == 'R'])
print("Total R Certified Movies:", total_r_certified)
```

```
Total R Certified Movies: 183
```

[27]:
```
#What is the average IMDb Rating of movies with Certificate 'PG-13'?
avg_imdb_pg13 = data[data['Certificates'] == 'PG-13']['IMDb Rating'].mean()
print("Average IMDb PG-13:", avg_imdb_pg13)
```

```
Average IMDb PG-13: 6.882291666666667
```

[30]:
```
#Which Star has acted in the most movies in this dataset?
star_list = data['Star Cast'].apply(lambda x: x.split(','))
flat_list = [item.strip() for sublist in star_list for item in sublist]
most_active_star = max(set(flat_list), key=flat_list.count)
print("Most Active Star:", most_active_star)
```

```
Most Active Star: Gil KenanJason ReitmanIvan Reitman
```

[31]:
```
#What year has the highest total Duration (minutes) of movies in this dataset?
year_duration_sum = data.groupby('Year')['Duration (minutes)'].sum()
highest_year_duration = year_duration_sum.idxmax()
print("Year with Highest Total Duration:", highest_year_duration)
```

```
Year with Highest Total Duration: 2024
```