Scraping IMDB movie data using requests and beatifulsoup

1. Importing libraries/modules

In [48]:

```
import numpy as np
import pandas as pd
import requests
from bs4 import BeautifulSoup
import openpyxl
import matplotlib.pyplot as pp
import seaborn as sb
%matplotlib inline
```

2. Scraping the data and saving it as an excel file

In [18]:

```
excel = openpyxl.Workbook() #creatng an excel workbook where the data will be saved
sheet = excel.active
sheet.title = "Top Rated Movies" #naming the sheet
sheet.append(['Rank', 'Name', 'Year of Release', 'IMDB Rating']) #naming the column headers


try:
    source = requests.get("https://www.imdb.com/chart/top/")
    source.raise_for_status() #checking if the url is a valid one, and if it is not, it will return an exception

    soup = BeautifulSoup(source.text,'html.parser') #this collect the html content of the url above and then par

    movies = soup.find('tbody',class_="lister-list").find_all('tr') #finds all the tags with tr. Each tr tag is

    for movie in movies: #creating a loop. for each movie in the movies body, the program should return thefollo

        name = movie.find('td',class_="titleColumn").a.text #this extracts the title of movies into the name var

        rank = movie.find('td',class_="titleColumn").get_text(strip=True).split('.')[0] #first value after the '

        year = movie.find('td',class_="titleColumn").span.text.strip('()') #strip() basically removes anything y

        rating = movie.find('td',class_="ratingColumn imdbRating").strong.text

        sheet.append([rank, name, year, rating]) #saves the values into the excel file

except Exception as e:
    print(e)

excel.save('IMDB Movie Ratings.xlsx')
```

3. Importing the dataset

In [120]:
```
data=pd.read_excel('IMDB Movie Ratings.xlsx')
data.head()
```

Out[120]:

	Rank	Name	Year of Release	IMDB Rating
0	1	The Shawshank Redemption	1994	9.2
1	2	The Godfather	1972	9.2
2	3	The Dark Knight	2008	9.0
3	4	The Godfather Part II	1974	9.0
4	5	12 Angry Men	1957	8.9

4. Inspecting the data

In [121]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 250 entries, 0 to 249
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Rank             250 non-null    int64  
 1   Name             250 non-null    object 
 2   Year of Release  250 non-null    int64  
 3   IMDB Rating      250 non-null    float64
dtypes: float64(1), int64(2), object(1)
memory usage: 7.9+ KB
```

Notes:

```
    * This data contains records of the top 250 movies on the IMDB website.
    * It has 4 columns and 250 rows
    * It has 2 coulumns of type (int64), 2 coulumns each of types (float64) and (object)
    * There are no missing values in the data
```

5. Performing EDA (Exploratory Data Analysis)

In [160]:
```
#total number of release years
data["Year of Release"].nunique()
```

Out[160]:
```
86
```

In [161]:
```
#return the number of movies released in each year
movies_per_releaseYear = pd.DataFrame(data.groupby(["Year of Release"])["Name"].count())
movies_per_releaseYear.reset_index(inplace=True)
movies_per_releaseYear.head()
```

Out[161]:

	Year of Release	Name
0	1921	1
1	1924	1
2	1925	1
3	1926	1
4	1927	1

In [124]:
```
#changing the data type of the year column to string
data["Year of Release"]=data["Year of Release"].astype(str)
movies_per_releaseYear["Year of Release"]=movies_per_releaseYear["Year of Release"].astype(str)
```

In [125]:
```
#what is the maximum number of movies released?
movies_per_releaseYear["Name"].max()
```

Out[125]:
```
8
```

In [126]:
```
#distinct number of movies released
movies_per_releaseYear["Name"].unique()
```

Out[126]:
```
array([1, 2, 3, 4, 6, 5, 8, 7], dtype=int64)
```

In [127]:

```python
#in what year was the maximum number of movies released?
movies_per_releaseYear[movies_per_releaseYear["Name"]==8]
```

Out[127]:

	Year of Release	Name
58	1995	8

In [158]:

```python
#filtering top 250 movies by release year
data[data["Year of Release"]=="1995"]
```

Out[158]:

	Rank	Name	Year of Release	IMDB Rating
18	19	Se7en	1995	8.6
39	40	The Usual Suspects	1995	8.5
73	74	Braveheart	1995	8.3
74	75	Toy Story	1995	8.3
110	111	Heat	1995	8.2
136	137	Casino	1995	8.2
179	180	Before Sunrise	1995	8.1
237	238	La haine	1995	8.0

In [150]:

```python
#get the min and max rating values
print("Min Rating: ", data["IMDB Rating"].min())
print("Max Rating: ", data["IMDB Rating"].max())
```

```
Min Rating:  8.0
Max Rating:  9.2
```

In [155]:

```
#top 250 movies with the lowest rating, their ranks and their release years
data[data["IMDB Rating"]==8.0]
```

Out[155]:

	Rank	Name	Year of Release	IMDB Rating
210	211	Rocky	1976	8.0
211	212	Ford v Ferrari	2019	8.0
212	213	Platoon	1986	8.0
213	214	Pather Panchali	1955	8.0
214	215	Stand by Me	1986	8.0
215	216	The Terminator	1984	8.0
216	217	Spotlight	2015	8.0
217	218	Rush	2013	8.0
218	219	Logan	2017	8.0
219	220	Network	1976	8.0
220	221	Ratatouille	2007	8.0

In [157]:

```
#top 250 movies with the highest rating and their release date
data[data["IMDB Rating"]==9.2]
```

Out[157]:

	Rank	Name	Year of Release	IMDB Rating
0	1	The Shawshank Redemption	1994	9.2
1	2	The Godfather	1972	9.2

6. Summary

```
1. There are 86 years alltogether, from 1921 to 2022

2. from the top 250 movies data, the highest number of movies released is 8, in the year 1995

3. rating ranged from 8.0 to 9.2
```

```
4. 40 movies were rated 8.0, while 2 movies were rated 9.2 (The Shawshank Redemption (1994) and The God
father
(1972), in the order of their ranking)
```

In [ ]:

In [ ]: