In [2]:
```
#Import Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [18]:
```
df=pd.read_csv('mymoviedb.csv', lineterminator='\n')
```

In [19]:
```
df.head()
```

Out[19]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Langu
0	2021-12-15	Spider-Man: No Way Home	Peter Parker is unmasked and no longer able to...	5083.954	8940	8.3	
1	2022-03-01	The Batman	In his second year of fighting crime, Batman u...	3827.658	1151	8.1	
2	2022-02-25	No Exit	Stranded at a rest stop in the mountains durin...	2618.087	122	6.3	
3	2021-11-24	Encanto	The tale of an extraordinary family, the Madri...	2402.201	5076	7.7	
4	2021-12-22	The King's Man	As a collection of history's worst tyrants and...	1895.511	1793	7.0	

In [20]:
```
# viewing dataset info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9827 entries, 0 to 9826
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Release_Date       9827 non-null   object 
 1   Title              9827 non-null   object 
 2   Overview           9827 non-null   object 
 3   Popularity         9827 non-null   float64
 4   Vote_Count         9827 non-null   int64  
 5   Vote_Average       9827 non-null   float64
 6   Original_Language  9827 non-null   object 
 7   Genre              9827 non-null   object 
 8   Poster_Url         9827 non-null   object 
dtypes: float64(2), int64(1), object(6)
memory usage: 691.1+ KB
```

In [21]:
```
# exploring genres column
df['Genre'].head()
```

Out[21]:
```
0    Action, Adventure, Science Fiction
1             Crime, Mystery, Thriller
2                              Thriller
3    Animation, Comedy, Family, Fantasy
4      Action, Adventure, Thriller, War
Name: Genre, dtype: object
```

In [24]:
```
# check for duplicated rows
df.duplicated().sum()
```

Out[24]:
```
0
```

In [25]:
```
# exploring summary statistics
df.describe()
```

Out[25]:

	Popularity	Vote_Count	Vote_Average
count	9827.000000	9827.000000	9827.000000
mean	40.326088	1392.805536	6.439534
std	108.873998	2611.206907	1.129759
min	13.354000	0.000000	0.000000
25%	16.128500	146.000000	5.900000
50%	21.199000	444.000000	6.500000
75%	35.191500	1376.000000	7.100000
max	5083.954000	31077.000000	10.000000

```
Exploration Summary

1 The dataframe consists of 9,827 rows and 9 columns.
2 The dataset appears tidy, with no NaN or duplicated values.
```

```
3 The data type of the "Release_date" column needs to be converted to
datetime format, and only the year should be extracted.
4 The "Overview" and "Poster_Url" columns will be dropped, as they are not
useful for further analysis.
5 There are noticeable outliers in the "Popularity" column.
6 The "Vote_Average" column should be categorized for proper analysis.
7 The "Genre" column contains comma-separated values and extra white
spaces that need to be cleaned.
```

Data Cleaning

Casting Release_Date column and extracing year values

In [31]:
```
# casting column
df['Release_Date']=pd.to_datetime(df['Release_Date'])

# confirming changes
print(df['Release_Date'].dtypes)
```

```
datetime64[ns]
```

In [33]:
```
df['Release_Date']=df['Release_Date'].dt.year

df['Release_Date'].dtypes
```

Out[33]:
```
dtype('int32')
```

In [34]:
```
df.head()
```

Out[34]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Langu
0	2021	Spider-Man: No Way Home	Peter Parker is unmasked and no longer able to...	5083.954	8940	8.3	
1	2022	The Batman	In his second year of fighting crime, Batman u...	3827.658	1151	8.1	
2	2022	No Exit	Stranded at a rest stop in the mountains durin...	2618.087	122	6.3	
3	2021	Encanto	The tale of an extraordinary family, the Madri...	2402.201	5076	7.7	
4	2021	The King's Man	As a collection of history's worst tyrants and...	1895.511	1793	7.0	

Dropping Overview and Poster-Url

In [35]:
```
# making list of column to be dropped
cols=['Overview', 'Poster_Url']
```

In [36]:
```
# dropping columns and confirming changes
df.drop(cols,axis=1,inplace=True)
```

In [39]:
```
df.columns
```

Out[39]:
```
Index(['Release_Date', 'Title', 'Popularity', 'Vote_Count', 'Vote_Averag
e',
       'Original_Language', 'Genre'],
      dtype='object')
```

In [40]:
```
df.head()
```

Out[40]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	Gen
0	2021	Spider-Man: No Way Home	5083.954	8940	8.3	en	Actio Adventur Scienc Fictic
1	2022	The Batman	3827.658	1151	8.1	en	Crim Myster Thrill
2	2022	No Exit	2618.087	122	6.3	en	Thrill
3	2021	Encanto	2402.201	5076	7.7	en	Animatio Comec Fami Fantas
4	2021	The King's Man	1895.511	1793	7.0	en	Actio Adventur Thrille W

categorizing Vote_Average column

We would cut the Vote_Average values and make 4 categories: popular, average, below_avg, not_popular to describe it more using catigorize_col() function provided above.

In [44]:
```
def categorize_col(df, col, labels):

# setting the edges to cut the column accordingly
   edges = [df[col].describe()['min'],
            df[col].describe()['25%'],
            df[col].describe()['50%'],
            df[col].describe()['75%'],
            df[col].describe()['max']]

   df[col] = pd.cut(df[col], edges, labels = labels, duplicates='drop')
   return df
```

In [46]:
```
# define labels for edges
labels = ['not_popular', 'below_avg', 'average', 'popular']

# categorize column based on labels and edges
categorize_col(df, 'Vote_Average', labels)

# confirming changes
df['Vote_Average'].unique()
```

Out[46]:
```
['popular', 'below_avg', 'average', 'not_popular', NaN]
Categories (4, object): ['not_popular' < 'below_avg' < 'average' < 'popula
r']
```

In [47]:
```
df.head()
```

Out[47]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	Gen
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Actio Adventur Scienc Fictic
1	2022	The Batman	3827.658	1151	popular	en	Crim Myster Thrill
2	2022	No Exit	2618.087	122	below_avg	en	Thrill
3	2021	Encanto	2402.201	5076	popular	en	Animatio Comec Fami Fantas
4	2021	The King's Man	1895.511	1793	average	en	Actio Adventur Thrille W

In [50]:
```
# exploring column
df['Vote_Average'].value_counts()
```

Out[50]:
```
Vote_Average
not_popular    2467
popular        2450
average        2412
below_avg      2398
Name: count, dtype: int64
```

In [51]:
```
#checking for Null Values
df.isna().sum()
```

Out[51]:
```
Release_Date           0
Title                  0
Popularity             0
Vote_Count             0
Vote_Average         100
Original_Language      0
Genre                  0
dtype: int64
```

In [52]:
```
# dropping NaNs
df.dropna(inplace=True)

# confirming
df.isna().sum()
```

Out[52]:
```
Release_Date         0
Title                0
Popularity           0
Vote_Count           0
Vote_Average         0
Original_Language    0
Genre                0
dtype: int64
```

In [53]:
```
df.head()
```

Out[53]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	Gen
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Actio Adventur Scienc Fictic
1	2022	The Batman	3827.658	1151	popular	en	Crim Myster Thrill
2	2022	No Exit	2618.087	122	below_avg	en	Thrill
3	2021	Encanto	2402.201	5076	popular	en	Animatio Comec Fami Fantas
4	2021	The King's Man	1895.511	1793	average	en	Actio Adventur Thrille W

split genres into a list and then explode our dataframe to have only one genre per row for each movie

In [55]:
```
# split the strings into lists
df['Genre']=df['Genre'].str.split(', ')

# explode the lists
df=df.explode('Genre').reset_index(drop=True)

df.head()
```

Out[55]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Actio
1	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Adventure
2	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Science Fictio
3	2022	The Batman	3827.658	1151	popular	en	Crime
4	2022	The Batman	3827.658	1151	popular	en	Myster

In [61]:
```
# casting column into category
df['Genre'] = df['Genre'].astype('category')

# confirming changes
df['Genre'].dtypes
```

Out[61]:
```
CategoricalDtype(categories=['Action', 'Adventure', 'Animation', 'Comedy',
'Crime',
                  'Documentary', 'Drama', 'Family', 'Fantasy', 'History',
                  'Horror', 'Music', 'Mystery', 'Romance', 'Science Fictio
n',
                  'TV Movie', 'Thriller', 'War', 'Western'],
, ordered=False)
```

In [62]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25552 entries, 0 to 25551
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype   
---  ------             --------------  -----   
 0   Release_Date       25552 non-null  int32   
 1   Title              25552 non-null  object  
 2   Popularity         25552 non-null  float64 
 3   Vote_Count         25552 non-null  int64   
 4   Vote_Average       25552 non-null  category
 5   Original_Language  25552 non-null  object  
 6   Genre              25552 non-null  category
dtypes: category(2), float64(1), int32(1), int64(1), object(2)
memory usage: 949.2+ KB
```

In [63]:
```
df.nunique()
```

Out[63]:
```
Release_Date          100
Title                9415
Popularity           8088
Vote_Count           3265
Vote_Average            4
Original_Language      42
Genre                  19
dtype: int64
```

In [79]:
```
df.head()
```

Out[79]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Actio
1	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Adventure
2	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Science Fictio
3	2022	The Batman	3827.658	1151	popular	en	Crime
4	2022	The Batman	3827.658	1151	popular	en	Myster

In [76]:
```
df.to_csv('cleaned_mymoviedb.xlsx', index=False)
```

In [77]:
```
pip install openpyxl
```

```
Requirement already satisfied: openpyxl in c:\users\stalin\anaconda3\lib\s
ite-packages (3.0.10)
Requirement already satisfied: et_xmlfile in c:\users\stalin\anaconda3\lib
\site-packages (from openpyxl) (1.1.0)
Note: you may need to restart the kernel to use updated packages.
```

In [78]:
```
df.to_excel('cleaned_mymoviedb.xlsx', index=False, engine='openpyxl')
```

In [ ]:
```

```