Importing libraries

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

Importing dataset

In [2]:
```python
df = pd.read_csv("mymoviedb.csv", lineterminator='\n')
df.head()
```

Out[2]:

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

In [3]:
```python
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
```

Missing Values

In [4]:
```python
df.isna().sum()
```

Out[4]:
```
Release_Date         0
Title                0
Overview             0
Popularity           0
Vote_Count           0
Vote_Average         0
Original_Language    0
Genre                0
Poster_Url           0
dtype: int64
```

In [5]:
```python
df.duplicated().sum()
```

Out[5]:
```
0
```

In [6]:
```
df.describe()
```

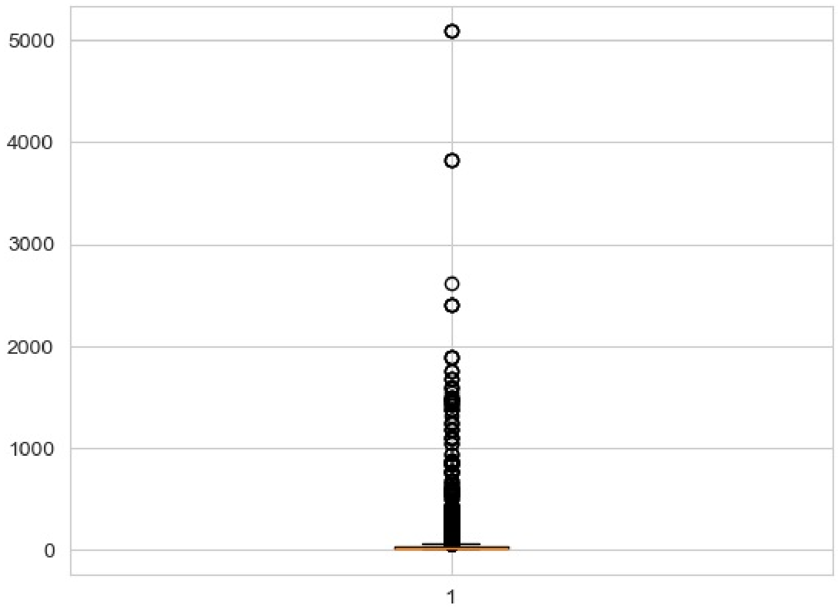
Out[6]:

	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

In [7]:
```
df['Genre'].head()
```

Out[7]:
```
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 [167…
```
plt.boxplot(df['Popularity'])
plt.show()
```



• Exploration Summary • We have a dataframe consisting of 9827 rows and 9 columns. • Our dataset looks a bit tidy with no NaNs nor duplicated values. • Release_Date column needs to be casted into date time and to extract only the year value. • Overview and Poster-Url wouldn't be so useful during analysis, so we'll drop them. • There is noticable outliers in Popularity column. • Vote_Average bettter be categorised for proper analysis. • Genre column has comma seperated values and white spaces that needs to be handled and casted into category.

In [9]:
```
df['Release_Date']=pd.to_datetime(df['Release_Date'])
print(df['Release_Date'].dtype)
```

```
datetime64[ns]
```

In [10]:
```
df['Release_Date'] = df['Release_Date'].dt.year
print(df['Release_Date'].dtype)
```

```
int32
```

In [11]:
```
df.head()
```

Out[11]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Language	Genre	
0	2021	Spider-Man: No Way Home	Peter Parker is unmasked and no longer able to...	5083.954	8940	8.3	en	Action, Adventure, Science Fiction	https://image.tmdb.o
1	2022	The Batman	In his second year of fighting crime, Batman u...	3827.658	1151	8.1	en	Crime, Mystery, Thriller	https://image.tmdb.org
2	2022	No Exit	Stranded at a rest stop in the mountains durin...	2618.087	122	6.3	en	Thriller	https://image.tmdb.org/
3	2021	Encanto	The tale of an extraordinary family, the Madri...	2402.201	5076	7.7	en	Animation, Comedy, Family, Fantasy	https://image.tmdb.org
4	2021	The King's Man	As a collection of history's worst tyrants and...	1895.511	1793	7.0	en	Action, Adventure, Thriller, War	https://image.tmdb.org

Droppping columns

In [13]:
```python
cols = ['Overview', 'Poster_Url']
```

In [14]:
```python
df.drop(cols, axis=1, inplace=True)
```

In [15]:
```python
df.columns
```

Out[15]:
```
Index(['Release_Date', 'Title', 'Popularity', 'Vote_Count', 'Vote_Average',
       'Original_Language', 'Genre'],
      dtype='object')
```

Categorizing Vote_Count column: We would cut the Vote_Average values and make 4 categories. Popular, Average, below_avg, not_popular to describe it more using categorize_col() function.

In [17]:
```python
def categorize_col(df, col, labels):

    edges = [
        df[col].describe()['min'] - 0.001,
        df[col].describe()['25%'],
        df[col].describe()['50%'],
        df[col].describe()['75%'],
        df[col].describe()['max'] + 0.001
    ]

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

In [18]:
```python
labels = ['not_popular', 'below_avg', 'average', 'popular']

categorize_col(df, 'Vote_Average', labels)

df['Vote_Average'].unique()
```

Out[18]:
```
['popular', 'below_avg', 'average', 'not_popular']
Categories (4, object): ['not_popular' < 'below_avg' < 'average' < 'popular']
```

In [36]:
```python
df.head()
```

Out[36]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Action, Adventure, Science Fiction
1	2022	The Batman	3827.658	1151	popular	en	Crime, Mystery, Thriller
2	2022	No Exit	2618.087	122	below_avg	en	Thriller
3	2021	Encanto	2402.201	5076	popular	en	Animation, Comedy, Family, Fantasy
4	2021	The King's Man	1895.511	1793	average	en	Action, Adventure, Thriller, War

In [38]:
```
df['Vote_Average'].value_counts()
```

Out[38]:
```
Vote_Average
not_popular    2567
popular        2450
average        2412
below_avg      2398
Name: count, dtype: int64
```

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

In [43]:
```
df['Genre'] = df['Genre'].str.split(', ')
```

In [49]:
```
df = df.explode('Genre').reset_index(drop=True)
df.head()
```

Out[49]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Action
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 Fiction
3	2022	The Batman	3827.658	1151	popular	en	Crime
4	2022	The Batman	3827.658	1151	popular	en	Mystery

Casting column into category

In [53]:
```
df['Genre'] = df['Genre'].astype('category')
df['Genre'].dtype
```

Out[53]:
```
CategoricalDtype(categories=['Action', 'Adventure', 'Animation', 'Comedy', 'Crime',
                  'Documentary', 'Drama', 'Family', 'Fantasy', 'History',
                  'Horror', 'Music', 'Mystery', 'Romance', 'Science Fiction',
                  'TV Movie', 'Thriller', 'War', 'Western'],
, ordered=False, categories_dtype=object)
```

In [55]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25793 entries, 0 to 25792
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype   
---  ------             --------------  -----   
 0   Release_Date       25793 non-null  int32   
 1   Title              25793 non-null  object  
 2   Popularity         25793 non-null  float64 
 3   Vote_Count         25793 non-null  int64   
 4   Vote_Average       25793 non-null  category
 5   Original_Language  25793 non-null  object  
 6   Genre              25793 non-null  category
dtypes: category(2), float64(1), int32(1), int64(1), object(2)
memory usage: 958.2+ KB
```

In [57]:
```
df.nunique()
```

Out[57]:
```
Release_Date          102
Title                9513
Popularity           8160
Vote_Count           3266
Vote_Average            4
Original_Language      43
Genre                  19
dtype: int64
```

Data visualization

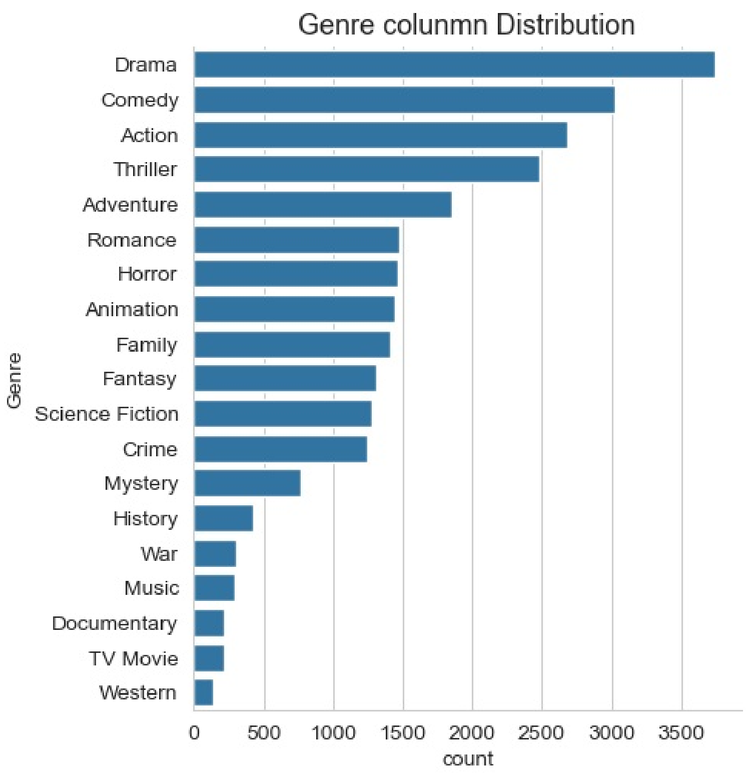
In [121…
```
sns.set_style('whitegrid')
```

Q.1 What is the most frequent genre of movies released on Netflix?

In [68]:
```
df['Genre'].describe()
```

Out[68]:
```
count      25793
unique        19
top        Drama
freq        3744
Name: Genre, dtype: object
```

In [125…
```
sns.catplot(y='Genre', data=df, kind='count', order=df['Genre'].value_counts().index)
plt.title('Genre colunmn Distribution',fontsize=13)
plt.show()
```



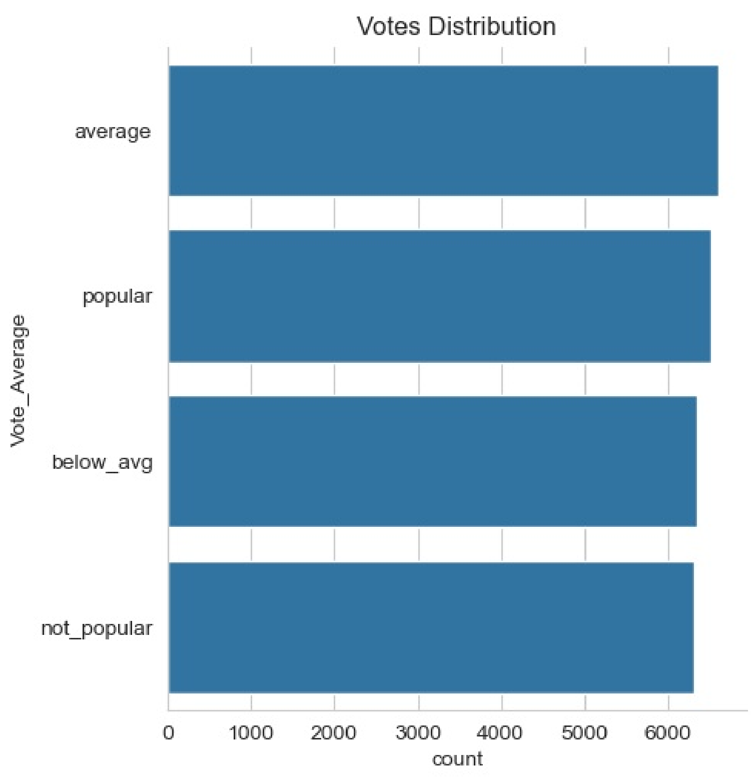
Q.2 Which has highest votes in vote avg column?

In [113…
```
df.head()
```

Out[113…

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Action
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 Fiction
3	2022	The Batman	3827.658	1151	popular	en	Crime
4	2022	The Batman	3827.658	1151	popular	en	Mystery

In [119…
```
sns.catplot(y='Vote_Average', data = df, kind='count', order=df['Vote_Average'].value_counts().index)
plt.title('Votes Distribution')
plt.show()
```



Q.3 What movie got the highest popularity? What's its genre?

```
In [134…  df.head(1)
```

Out[134…

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Action

```
In [142…  df[df['Popularity'] == df['Popularity'].max()]
```

Out[142…

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Action
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 Fiction

Highest popularity movie = Spider-Man: No Way Home Genre = Action, Adventure, Science Fiction

Q.4 What movie got the lowest popularity? What's its genre?

```
In [149…  df[df['Popularity'] == df['Popularity'].min()]
```

Out[149…

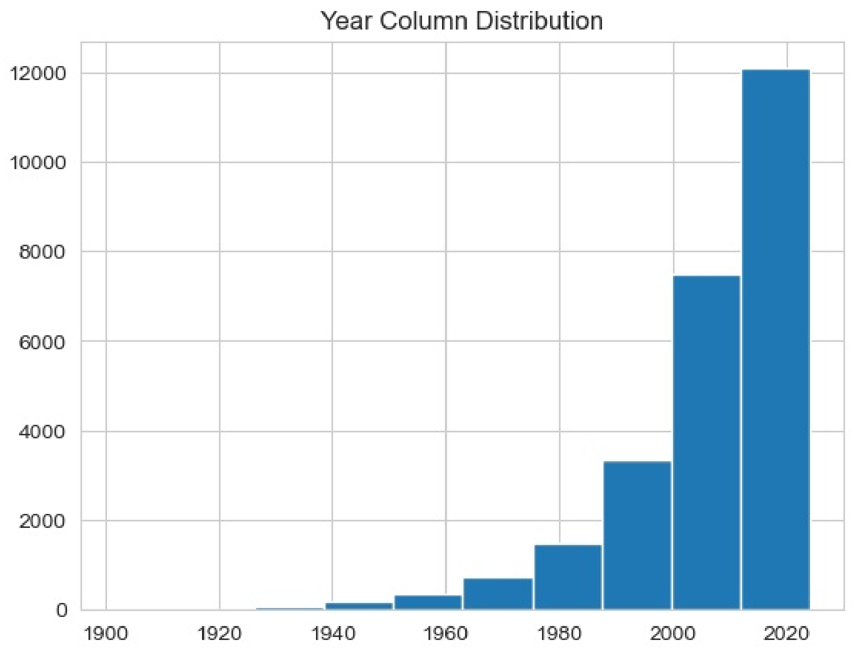
	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	Genre
25787	2021	The United States vs. Billie Holiday	13.354	152	average	en	Music
25788	2021	The United States vs. Billie Holiday	13.354	152	average	en	Drama
25789	2021	The United States vs. Billie Holiday	13.354	152	average	en	History
25790	1984	Threads	13.354	186	popular	en	War
25791	1984	Threads	13.354	186	popular	en	Drama
25792	1984	Threads	13.354	186	popular	en	Science Fiction

```
In [ ]:  Lowest popularity movie = The United States vs. Billlie Holiday
         Genre = Music, Drama, History

         Lowest popularity movie = Threads
         Genre = War, Drama, Science Fiction
```

Q.5 Which year has the most filmmed movies?

```
In [160…  df['Release_Date'].hist()
          plt.title('Year Column Distribution')
plt.show()
```



Year 2020 has the most filmmed movies.

Conclusion:

Q.1 What is the most frequent genre of movies released on Netflix?

> > > Drama genre is the most frequent genre in our dataset and has appeared more than 14% of the time among 19 other genres.

Q.2 Which has highest votes in vote avg column?

> > > Drama again gets the highest popularity among fans by being having more than 18.5% of movies popularities.

Q.3 What movie got the highest popularity? What's its genre?

> > > The Spider-Man: No Way Home has the highest popularity rate in our dataset and it has genres Action, Adventure, Science Fiction.

Q.4 What movie got the lowest popularity? What's its genre?

> > > The The United States vs. Billlie Holiday (Genre: Music, Drama, History) and Threads (Genre: War, Drama, Science Fiction) got the lowest popularity.

Q.5 Which year has the most filmmed movies?

> > > Year 2020 has the most filmmed rate in our datatset.