```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df= pd.read_csv("mymoviedb.csv", lineterminator='\n')
df.head()
```

	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.org/t/p/orig
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/t/p/origi
			Stranded at						

```
#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
```

```
#Exploring Genre Column
df["Genre"].head()
```

	Genre
0	Action, Adventure, Science Fiction
1	Crime, Mystery, Thriller
2	Thriller
3	Animation, Comedy, Family, Fantasy
4	Action, Adventure, Thriller, War

dtype: object

```
#Check for Duplicated Rows
df.duplicated().sum()
```

```
0
```

```
#Exploring Summary Statistics
df.describe()
```

	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

. We have a dataframe consisting of 9827 rows and 9 column
. our dataset looks a bit tidy with no NaNs nor duplicated Values
. Release_Datecolumn needs tyo be casted into data time and extract only the year value
. Overview , Orginally_Language and poster_Url wouldnt be so useful during analysis , so will drop them.
. There is noticable outliers in popularity column.
. Vote_Average better be Categorised for Proper analysis.
. Genre column has comma Saperated Value and White Space that needs to be handled and casted into category

Data Cleaning

```
df.head()
```

	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.org/t/p/orig
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/t/p/origi
			Stranded at						

```
df["Release_Date"]=pd.to_datetime(df["Release_Date"])
print(df["Release_Date"].dtypes)
```

```
datetime64[ns]
```

```
df["Release_Date"]=df["Release_Date"].dt.year
df["Release_Date"].dtype
```

```
dtype('int32')
```

```
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   int32
 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), int32(1), int64(1), object(5)
memory usage: 652.7+ KB
```

```
df.head()
```

	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.org/t/p/orig
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/t/p/origi
			Stranded at						

Dropping Overview,Original_Language and Poster-Url

```
cols=["Overview","Original_Language","Poster_Url"] #Making List column to be Dropped
df.drop(cols,axis=1,inplace=True) #Permanent Delete
df.columns
```

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

```
df.head()
```

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	8.3	Action, Adventure, Science Fiction
1	2022	The Batman	3827.658	1151	8.1	Crime, Mystery, Thriller
2	2022	No Exit	2618.087	122	6.3	Thriller
3	2021	Encanto	2402.201	5076	7.7	Animation, Comedy, Family, Fantasy
4	2021	The King's Man	1895.511	1793	7.0	Action, Adventure, Thriller, War

Categorizing Vote_Average Column:

We would Split the Vote_Average Value and make 4 Categories:Popular,Average,Below_Average,Non_Popular to describe it more using Categorize_col() function provided above

```
def catigorize_col(df,col,labels):

    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

labels=["Not_Popular","Below_Average","Average","Popular"]
catigorize_col(df,"Vote_Average",labels)
df["Vote_Average"].unique()
```

```
['Popular', 'Below_Average', 'Average', 'Not_Popular', NaN]
Categories (4, object): ['Not_Popular' < 'Below_Average' < 'Average' < 'Popular']
```

```
df.head()
```

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

```
df["Vote_Average"].value_counts()
```

Vote_Average	count
Not_Popular	2467
Popular	2450
Average	2412
Below_Average	2398

```
df.dropna(inplace=True)
df.isna().sum()
```

	0
Release_Date	0
Title	0
Popularity	0
Vote_Count	0
Vote_Average	0
Genre	0

```
df.head()
```

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

We'd Split Genre into a list and then explode out dataframe to have only one Genre per row for each Movie

```
#split the string into list
df["Genre"]=df["Genre"].str.split(',')

#Explode the list
df=df.explode("Genre").reset_index(drop=True)
df.head()
```

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	Popular	Action
1	2021	Spider-Man: No Way Home	5083.954	8940	Popular	Adventure
2	2021	Spider-Man: No Way Home	5083.954	8940	Popular	Science Fiction
3	2022	The Batman	3827.658	1151	Popular	Crime
4	2022	The Batman	3827.658	1151	Popular	Mystery

```
df["Genre"]=df["Genre"].astype("category")
df["Genre"].dtype
```

```
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)
```

```
df.nunique()
```

	0
Release_Date	100
Title	9415
Popularity	8088
Vote_Count	3265
Vote_Average	4
Genre	19

```
df.head()
```

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	Popular	Action
1	2021	Spider-Man: No Way Home	5083.954	8940	Popular	Adventure
2	2021	Spider-Man: No Way Home	5083.954	8940	Popular	Science Fiction
3	2022	The Batman	3827.658	1151	Popular	Crime
4	2022	The Batman	3827.658	1151	Popular	Mystery

Data Visualization

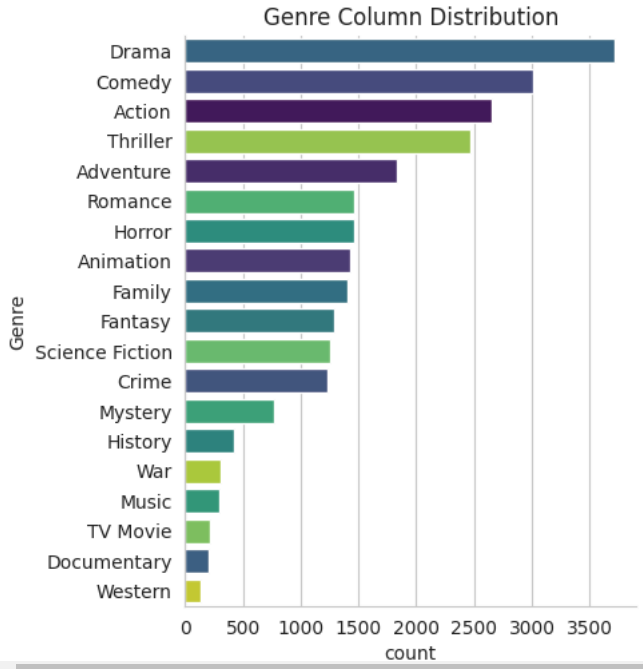
```
sns.set_style("whitegrid")
```

```
df["Genre"].describe()
```

	Genre
count	25552
unique	19
top	Drama
freq	3715

Q1. What is the Most Frequent Genre In The Dataset?

```
sns.catplot(y="Genre",data=df,kind="count",
            order=df["Genre"].value_counts().index,
            hue="Genre", legend=False,palette="viridis")
plt.title("Genre Column Distribution")
plt.show()
```




```
df["Genre"].value_counts()
```

Genre	count
Drama	3715
Comedy	3006
Action	2652
Thriller	2473
Adventure	1829
Romance	1461
Horror	1457
Animation	1426
Family	1405
Fantasy	1295
Science Fiction	1255
Crime	1235
Mystery	765
History	426
War	307
Music	291
TV Movie	214
Documentary	203
Western	137

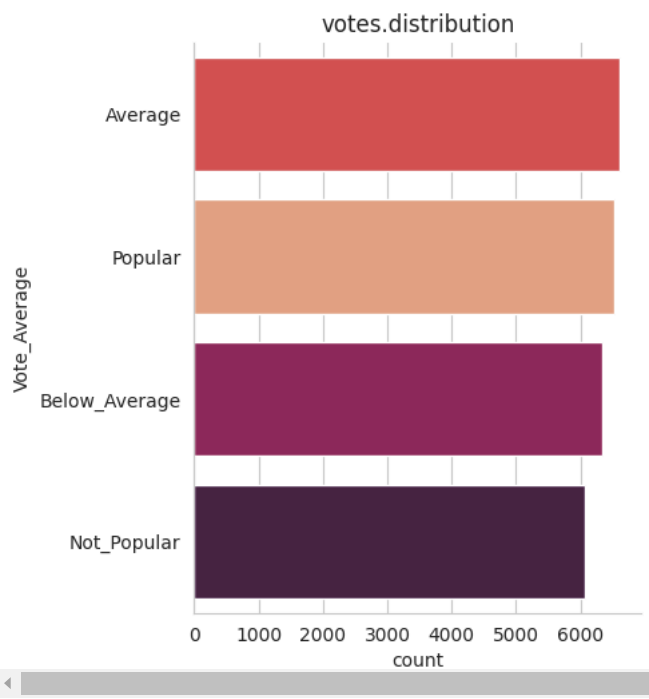
Q2.Which has Highest Votes in vote Avg Column?

```
df.head()
```

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	Popular	Action
1	2021	Spider-Man: No Way Home	5083.954	8940	Popular	Adventure
2	2021	Spider-Man: No Way Home	5083.954	8940	Popular	Science Fiction
3	2022	The Batman	3827.658	1151	Popular	Crime
4	2022	The Batman	3827.658	1151	Popular	Mystery

```
sns.catplot(y="Vote_Average", data=df, kind="count",
            order=df["Vote_Average"].value_counts().index,
            hue="Vote_Average", legend=False,
            palette="rocket")

plt.title("votes.distribution")
plt.show()
```



Q3.Which Movies Got The Highest Popularity?What its Genre

```
df[df["Popularity"]==df["Popularity"].max()]
```

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Genre
0	2021	Spider-Man: No Way Home	5083.954	8940	Popular	Action
1	2021	Spider-Man: No Way Home	5083.954	8940	Popular	Adventure
2	2021	Spider-Man: No Way Home	5083.954	8940	Popular	Science Fiction

Q4.Which Movies Got The Lowest Popularity? What its Genre

```
df[df["Popularity"]==df["Popularity"].min()]
```

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Genre
25546	2021	The United States vs. Billie Holiday	13.354	152	Average	Music
25547	2021	The United States vs. Billie Holiday	13.354	152	Average	Drama
25548	2021	The United States vs. Billie Holiday	13.354	152	Average	History
25549	1984	Threads	13.354	186	Popular	War
25550	1984	Threads	13.354	186	Popular	Drama
25551	1984	Threads	13.354	186	Popular	Science Fiction

Q5.Which Year Has the Most Filmmed Movies?

```
df["Release_Date"].hist()
plt.title("Release Date column distribution")
plt.show()
```