In [1]:

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

In [2]:

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

In [3]:

```
df.head()
```

Out[3]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Origi
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 [4]:

```
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 [5]:

```
df['Genre'].head()
```

Out[5]:

```
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 [6]:

```
df.duplicated().sum()
```

Out[6]:

```
0
```

In [7]:
```
df.describe()
```

Out[7]:

	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 0 columns.
- our dataset looks a bit tidy with no NaNs nor duplicated values.
- Release_Data column needs to be casted into date time and to extract only the year value.
- Overview, Original_Language and Poster-Url wouldn't be so useful during analysis, so we'll drop them.
- there is noticable outliers in Popularity column
- Vote_Average better be categorised for proper analysis.
- Genre column has comma separeted values and white spaces that needs to be handled and casted into category Exploration summary

In [8]:
```
df.head()
```

Out[8]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Origi
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 [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

df['Release_Date'].dtypes
```

Out[10]:
```
dtype('int32')
```

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

Out[11]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Origi
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 the Columns

In [12]:
```
cols = ['Overview', 'Original_Language', 'Poster_Url']
```

In [13]:
```
df.drop(cols,axis = 1, inplace = True)
df.columns
```

Out[13]:
```
Index(['Release_Date', 'Title', 'Popularity', 'Vote_Count', 'Vote_
Average',
       'Genre'],
      dtype='object')
```

In [14]:
```
df.head()
```

Out[14]:

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

In [15]:
```
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= '
    return df
```

In [16]:
```
labels = ['not_popular', 'below_avg', 'average', 'popular']

catigorize_col(df, 'Vote_Average', labels)

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

Out[16]:
```
['popular', 'below_avg', 'average', 'not_popular', NaN]
Categories (4, object): ['not_popular' < 'below_avg' < 'average' <
'popular']
```

In [17]:
```
df.head()
```

Out[17]:

	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_avg	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

In [18]:
```
df['Vote_Average'].value_counts()
```

Out[18]:
```
Vote_Average
not_popular    2467
popular        2450
average        2412
below_avg      2398
Name: count, dtype: int64
```

In [19]:
```
df.dropna(inplace = True)

df.isna().sum()
```

Out[19]:
```
Release_Date    0
Title           0
Popularity      0
Vote_Count      0
Vote_Average    0
Genre           0
dtype: int64
```

In [20]:
```
df.head()
```

Out[20]:

	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_avg	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 do split genre's into a list and then explode our dataframe to have only one genre per row for each movie.

In [21]:
```
df['Genre'] = df['Genre'].str.split(', ')

df = df.explode('Genre').reset_index(drop=True)
df.head()
```

Out[21]:

	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

In [22]:
```
df['Genre'] = df['Genre'].astype('category')

df['Genre'].dtypes
```

Out[22]:
```
CategoricalDtype(categories=['Action', 'Adventure', 'Animation',
'Comedy', 'Crime',
                  'Documentary', 'Drama', 'Family', 'Fantasy', 'Hi
story',
                  'Horror', 'Music', 'Mystery', 'Romance', 'Scienc
e Fiction',
                  'TV Movie', 'Thriller', 'War', 'Western'],
, ordered=False)
```

In [23]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25552 entries, 0 to 25551
Data columns (total 6 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   Genre         25552 non-null  category
dtypes: category(2), float64(1), int32(1), int64(1), object(1)
memory usage: 749.6+ KB
```

In [24]:
```
df.nunique()
```

Out[24]:
```
Release_Date     100
Title           9415
Popularity      8088
Vote_Count      3265
Vote_Average       4
Genre             19
dtype: int64
```

```
Now that our dataset is clean and tidy, we are left with a total
of 6 columns and 25551 rows to dig into during our analysis.
```

# Data Visualization

here we'd use Matplotlib and seaborn for making some information visuals to gain insights about our data.

In [25]:
```
sns.set_style('whitegrid')
```

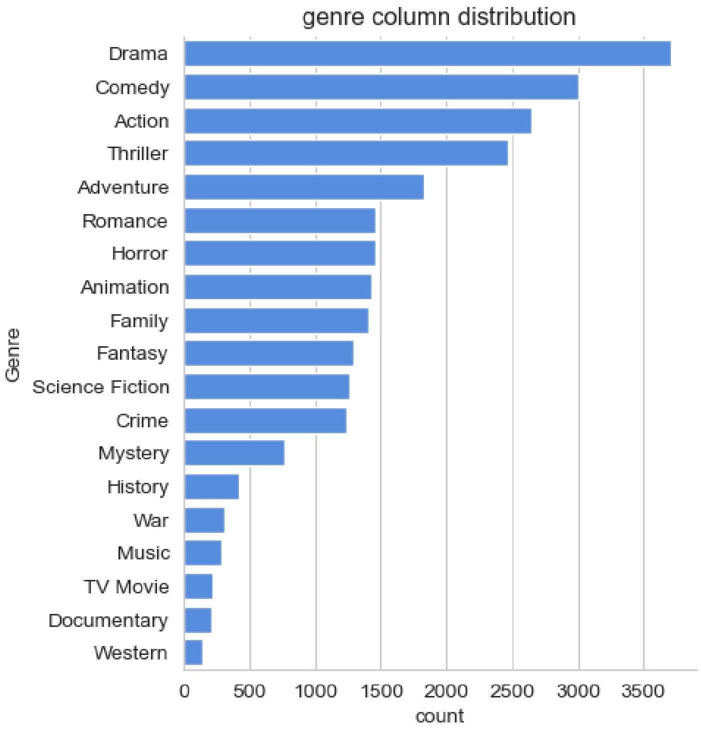
What is the most frequent genre in the dataset ?

In [26]:
```
df['Genre'].describe()
```

Out[26]:
```
count     25552
unique       19
top       Drama
freq       3715
Name: Genre, dtype: object
```

In [28]:
```
sns.catplot(y = 'Genre', data = df, kind = 'count', order = df['Genr
plt.title('genre column distribution')
plt.show()
```

```
C:\Users\Priyansu\anaconda3\Lib\site-packages\seaborn\axisgrid.py:
118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



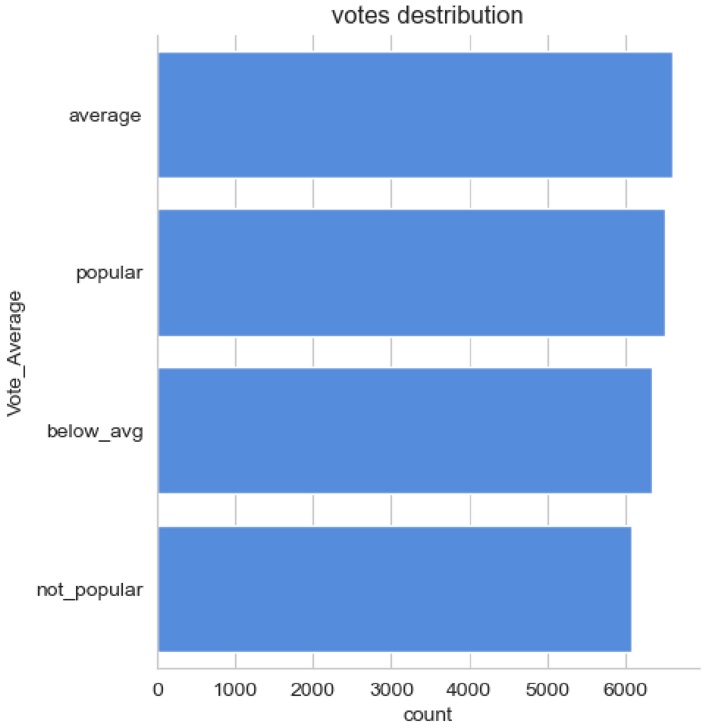
What genres has the highest vote ?

In [31]:

```
sns.catplot(y = 'Vote_Average', data = df, kind = 'count', order = d

plt.title('votes destribution')
plt.show()
```

```
C:\Users\Priyansu\anaconda3\Lib\site-packages\seaborn\axisgrid.py:
118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



Which movie got the highest popularity ? Whats its genre?

In [32]:

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

Out[32]:

	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

What movie got the lowest popularity? whats it genre ?

In [34]:

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

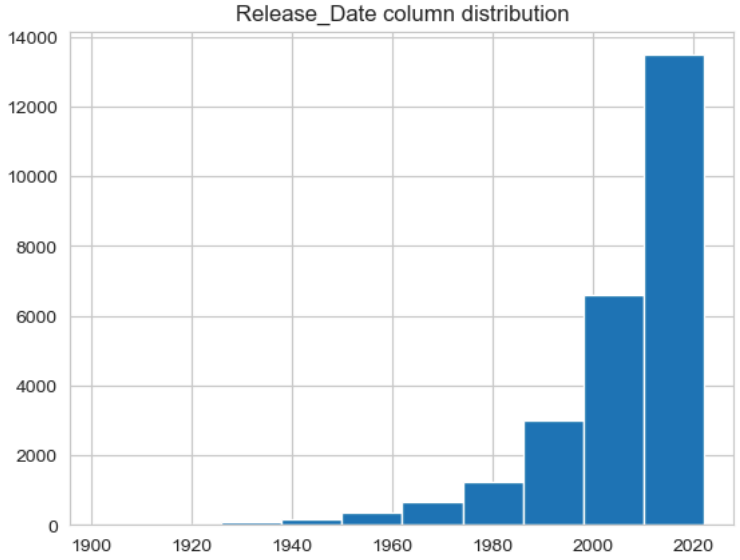
Out[34]:

	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

Which year has the most filmmed movies ?

In [35]:

```
df['Release_Date'].hist()
plt.title('Release_Date column distribution')
plt.show()
```



In [ ]:

In [ ]: