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

In [4]:
```
df = pd.read_csv('mymoviedb.csv', lineterminator='\n')
```

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

Out[6]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Lan
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 [7]:
```
df.tail()
```

Out[7]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Origina
9822	1973-10-15	Badlands	A dramatization of the Starkweather-Fugate kil...	13.357	896	7.6	
9823	2020-10-01	Violent Delights	A female vampire falls in love with a man she ...	13.356	8	3.5	
9824	2016-05-06	The Offering	When young and successful reporter Jamie finds...	13.355	94	5.0	
9825	2021-03-31	The United States vs. Billie Holiday	Billie Holiday spent much of her career being ...	13.354	152	6.7	
9826	1984-09-23	Threads	Documentary style account of a nuclear holocau...	13.354	186	7.8	

In [8]:
```
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 [13]:
```
df['Genre'].head()
```

Out[13]:
```
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 [14]:
```
df.duplicated().sum()
```

Out[14]:
```
0
```

In [15]:
```
df.describe()
```

Out[15]:

	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 [16]:
```
df.head()
```

Out[16]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Lan
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 [17]:
```
df['Release_Date'] = pd.to_datetime(df['Release_Date'])
```

In [18]:
```
print(df['Release_Date'].dtypes)
```

```
datetime64[ns]
```

In [20]:
```
df['Release_Date'] = df['Release_Date'].dt.year
df['Release_Date'].dtypes
```

Out[20]:
```
dtype('int32')
```

In [21]:
```
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
```

In [22]:
```
df.head()
```

Out[22]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Lan
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	

In [24]:
```
# making list of column to be dropped
cols = ['Overview', 'Original_Language', 'Poster_Url']
```

In [25]:
```
# dropping columns and confirming changes
df.drop(cols, axis = 1, inplace = True)
df.columns
```

Out[25]:
```
Index(['Release_Date', 'Title', 'Popularity', 'Vote_Count', 'Vote_Average',
       'Genre'],
      dtype='object')
```

In [26]:
```
df.head()
```

Out[26]:

	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

In [37]:
```
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
```

In [39]:
```
# define labels for edges
labels = ['not_popular', 'below_avg', 'average', 'popular']
```

In [41]:
```
# categorize column based on labels and edges
catigorize_col(df, 'Vote_Average', labels)
```

Out[41]:

	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
...	...	...	...	...	...	...
9822	1973	Badlands	13.357	896	popular	Drama, Crime
9823	2020	Violent Delights	13.356	8	not_popular	Horror
9824	2016	The Offering	13.355	94	not_popular	Mystery, Thriller, Horror
9825	2021	The United States vs. Billie Holiday	13.354	152	average	Music, Drama, History
9826	1984	Threads	13.354	186	popular	War, Drama, Science Fiction

9827 rows × 6 columns

In [42]:
```
 # confirming changes
df['Vote_Average'].unique()
```

Out[42]:
```
['popular', 'below_avg', 'average', 'not_popular', NaN]
Categories (4, object): ['not_popular' < 'below_avg' < 'average' < 'popular']
```

In [43]:
```
# exploring column
df['Vote_Average'].value_counts()
```

Out[43]:
```
Vote_Average
not_popular    2467
popular        2450
average        2412
below_avg      2398
Name: count, dtype: int64
```

In [44]:
```
# dropping NaNs
df.dropna(inplace = True)
```

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

Out[44]:
```
Release_Date    0
Title           0
Popularity      0
Vote_Count      0
Vote_Average    0
Genre           0
dtype: int64
```

In [45]:
```
df.head()
```

Out[45]:

	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 [46]:
```
# 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[46]:

	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 [47]:
```
# casting column into category
df['Genre'] = df['Genre'].astype('category')
```

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

Out[47]:
```
CategoricalDtype(categories=['Action', 'Adventure', 'Animation', 'Comedy', 'Crim
e',
                  'Documentary', 'Drama', 'Family', 'Fantasy', 'History',
                  'Horror', 'Music', 'Mystery', 'Romance', 'Science Fiction',
                  'TV Movie', 'Thriller', 'War', 'Western'],
, ordered=False, categories_dtype=object)
```

In [48]:
```
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 [49]:
```
df.nunique()
```

Out[49]:
```
Release_Date     100
Title           9415
Popularity      8088
Vote_Count      3265
Vote_Average       4
Genre             19
dtype: int64
```

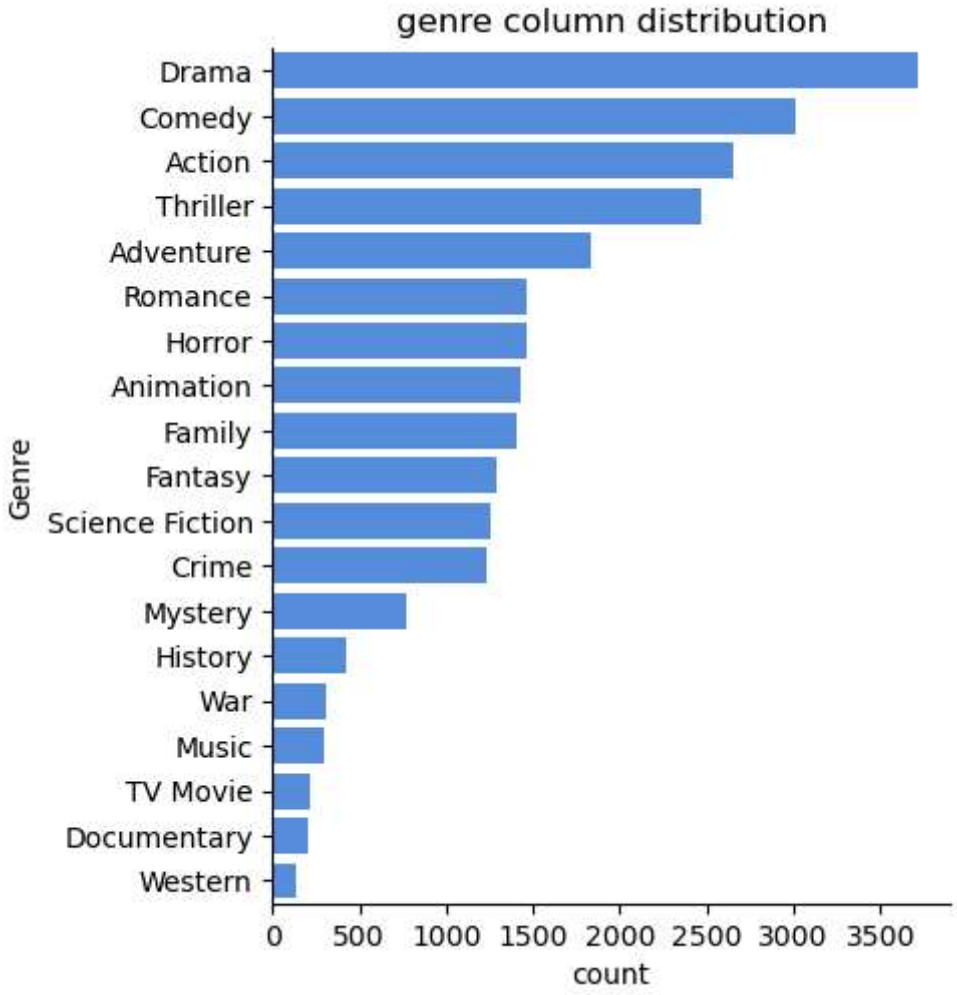
Q1: What is the most frequent genre in the dataset?

In [51]:
```
# showing stats. on genre column
df['Genre'].describe()
```

Out[51]:
```
count     25552
unique       19
top       Drama
freq       3715
Name: Genre, dtype: object
```

In [53]:
```
# visualizing genre column
sns.catplot(y = 'Genre', data = df, kind = 'count',
order = df['Genre'].value_counts().index,
color = '#4287f5')
plt.title('genre column distribution')
plt.show()
```

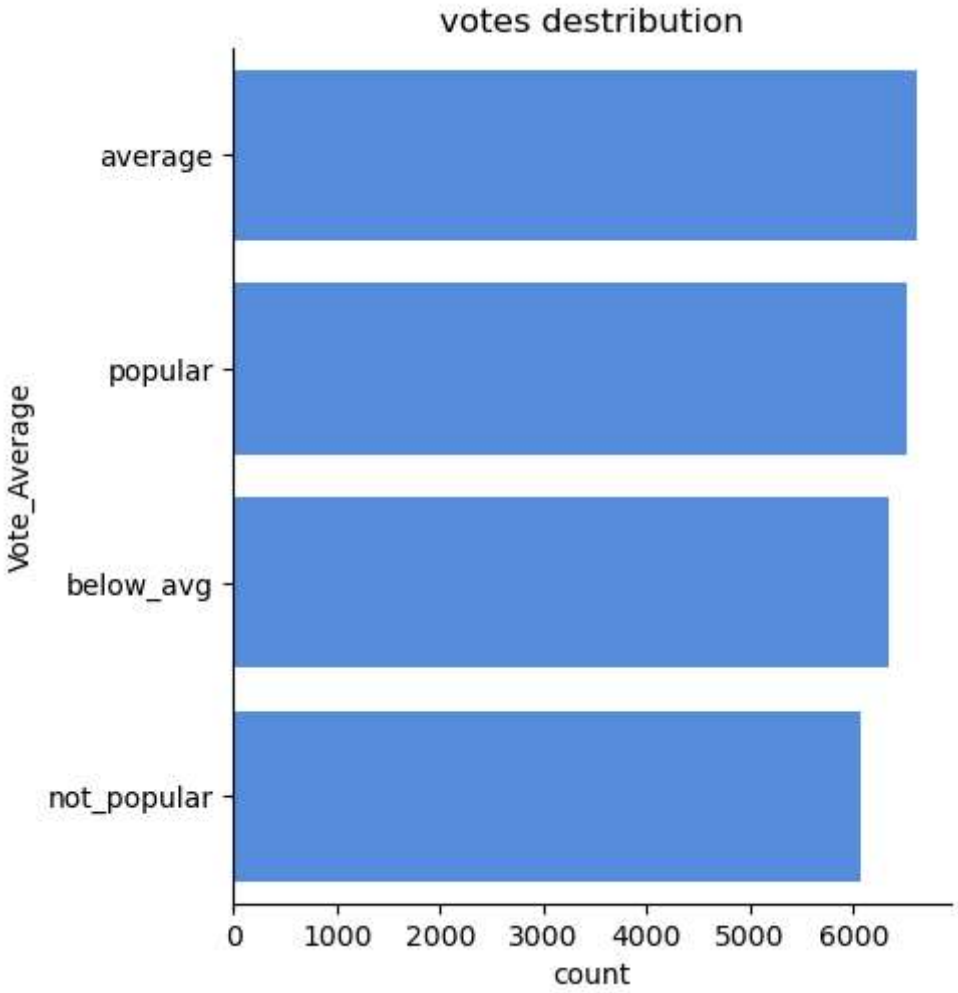
```
C:\Users\Admin\anaconda3\Lib\site-packages\seaborn\categorical.py:641: FutureWarnin
g: The default of observed=False is deprecated and will be changed to True in a futu
re version of pandas. Pass observed=False to retain current behavior or observed=Tru
e to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
C:\Users\Admin\anaconda3\Lib\site-packages\seaborn\categorical.py:641: FutureWarnin
g: The default of observed=False is deprecated and will be changed to True in a futu
re version of pandas. Pass observed=False to retain current behavior or observed=Tru
e to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
```



Q2: What genres has highest votes ?

In [56]:
```
sns.catplot(y = 'Vote_Average', data = df, kind = 'count',
order = df['Vote_Average'].value_counts().index,color = '#4287f5')
plt.title('votes destribution')
plt.show()
```

```
C:\Users\Admin\anaconda3\Lib\site-packages\seaborn\categorical.py:641: FutureWarnin
g: The default of observed=False is deprecated and will be changed to True in a futu
re version of pandas. Pass observed=False to retain current behavior or observed=Tru
e to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
C:\Users\Admin\anaconda3\Lib\site-packages\seaborn\categorical.py:641: FutureWarnin
g: The default of observed=False is deprecated and will be changed to True in a futu
re version of pandas. Pass observed=False to retain current behavior or observed=Tru
e to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
```



Q3: What movie got the highest popularit what's its genre ?

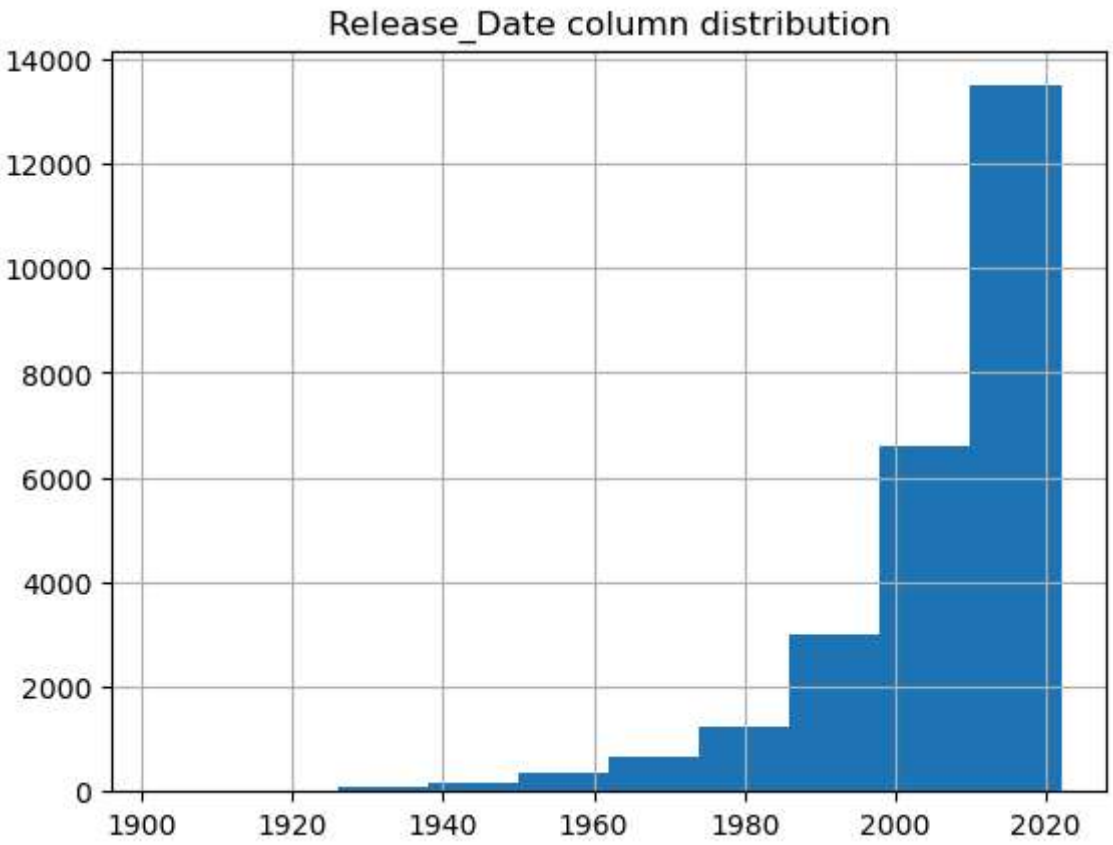
In [57]:
```
# checking max popularity in dataset
df[df['Popularity'] == df['Popularity'].max()]
```

Out[57]:

	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

Q5: Which year has the most filmmed movies?

In [58]:
```
df['Release_Date'].hist()
plt.title('Release_Date column distribution')
plt.show()
```



Q1: What is the most frequent genre in the dataset? Drama genre is the most frequent genre in our dataset and has appeared more than 14% of the times among 19 other genres. Q2: What genres has highest votes ? we have 25.5% of our dataset with popular vote (6520 rows). Drama again gets the highest popularity among fans by being having more than 18.5% of movies popularities. Q3: What movie got the highest popularity ? what's its Action , genre ? Spider-Man: No Way Home has the highest popularity rate in our dataset and it has genres of Adventure and Sience Fiction . Q3: What movie got the lowest popularity ? what's its genre ? The united states, thread' has the highest lowest rate in our dataset and it has

genres of music , drama , 'war', 'sci-fi' and history`. Q4: Which year has the most filmmed movies? year 2020 has the highest filmming rate in our dataset

In [ ]: