In [3]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

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

In [ ]:
```python
# DATA PRE-PROCESSING
```

In [7]:
```python
df.head()
```

Out[7]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Language	Genre	Poster_Url
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/original/1g0dhYtq4i...
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/original/74xTEgt7R3...
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/t/p/original/vDHsLnOWKl...
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/t/p/original/4j0PNHkMr5...
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/t/p/original/aq4Pwv5Xeu...

In [151…
```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
```

In [153…
```python
df['Genre'].head()
```

Out[153…
```
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 [9]:
```python
# To check duplicate in the data

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

Out[9]: 0

In [201…
```python
# Check the statistics of Movies on the basis of Popularity , Vote_count , Vote_average columns
# Using describe function this function only works on Numbers.
# standard deviation formula (std) =
#      root( (∑(Xi - mean)^2) / n )

df.describe()
```

Out[201…

	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 [ ]:
```python
# Exploration Summary

# - We have a datframe(Table-Like Structure) 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 , Original_Language and Poster-Url wouldn't be so useful during analysis, so we'll drop them.
# - There is noticable outliners in Popularity column.
# - Vote_Average better be categorised for proper analysie.
# - Genre column has comma separated values and white spaces tha need to be handled and casted into category Exploration Summary.
```

In [11]:
```python
df['Release_Date'] = pd.to_datetime(df['Release_Date'])

print(df['Release_Date'].dtypes)
```

```
datetime64[ns]
```

In [13]:
```python
df['Release_Date'] = df['Release_Date'].dt.year

df['Release_Date'].dtypes
```

Out[13]: dtype('int32')

In [15]:
```python
df.head()
```

Out[15]:

	Release_Date	Title	Overview	Popularity	Vote_Count	Vote_Average	Original_Language	Genre	Poster_Url
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/original/1g0dhYtq4i...
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/original/74xTEgt7R3...
2	2022	No Exit	Stranded at a rest stop in the mountains durin...	2618.087	122	6.3	en	Thriller	https://image.tmdb.org/t/p/original/vDHsLnOWKl...
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/t/p/original/4j0PNHkMr5...
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/t/p/original/aq4Pwv5Xeu...

In [17]:
```python
# Dropping the Columns....
cols = ['Overview' , 'Original_Language' , 'Poster_Url']
```

In [19]:
```python
df.drop(cols , axis = 1 , inplace = True)

df.columns
```

Out[19]:
```
Index(['Release_Date', 'Title', 'Popularity', 'Vote_Count', 'Vote_Average',
       'Genre'],
      dtype='object')
```

In [169…
```python
df.head()
```

Out[169…

	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 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 [21]:
```python
def categorize_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 [23]:
```python
labels = ['not_popular' , 'below_average' , 'average' , 'popular']

categorize_col(df , 'Vote_Average' , labels)

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

Out[23]:
```
['popular', 'below_average', 'average', 'not_popular', NaN]
Categories (4, object): ['not_popular' < 'below_average' < 'average' < 'popular']
```

In [173…
```python
df.head()
```

Out[173…

	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

In [25]:
```python
df['Vote_Average'].value_counts()
```

Out[25]:
```
Vote_Average
not_popular      2467
popular          2450
average          2412
below_average    2398
Name: count, dtype: int64
```

In [27]:
```python
# Remove the duplicate values and NaNs from the columns..

df.dropna(inplace = True)

# To check the duplicates and NaNs values are remove from the column.

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

Out[27]:
```
Release_Date    0
Title           0
Popularity      0
Vote_Count      0
Vote_Average    0
Genre           0
dtype: int64
```

In [ ]:
```python
# we'd split genres into a list and then explode our dataframe to have only one genre per row for each movie
```

In [29]:
```python
df['Genre'] = df['Genre'].str.split(', ')

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

Out[29]:

	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 [31]:
```python
# Casting column into category
df['Genre'] = df['Genre'].astype('category')

df['Genre'].dtypes
```

Out[31]:
```
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 [33]:
```python
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 [35]:
```python
# unique values in the columns

df.nunique()
```

Out[35]:
```
Release_Date     100
Title           9415
Popularity      8088
Vote_Count      3265
Vote_Average       4
Genre             19
dtype: int64
```

In [39]:
```python
df.head()
```

Out[39]:

	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 [ ]:
```python
# DATA VISUALIZATION
```

In [55]:
```python
# When you write sns.set_style('whitegrid'), it applies the whitegrid style to all Seaborn plots.

sns.set_style('whitegrid')
```

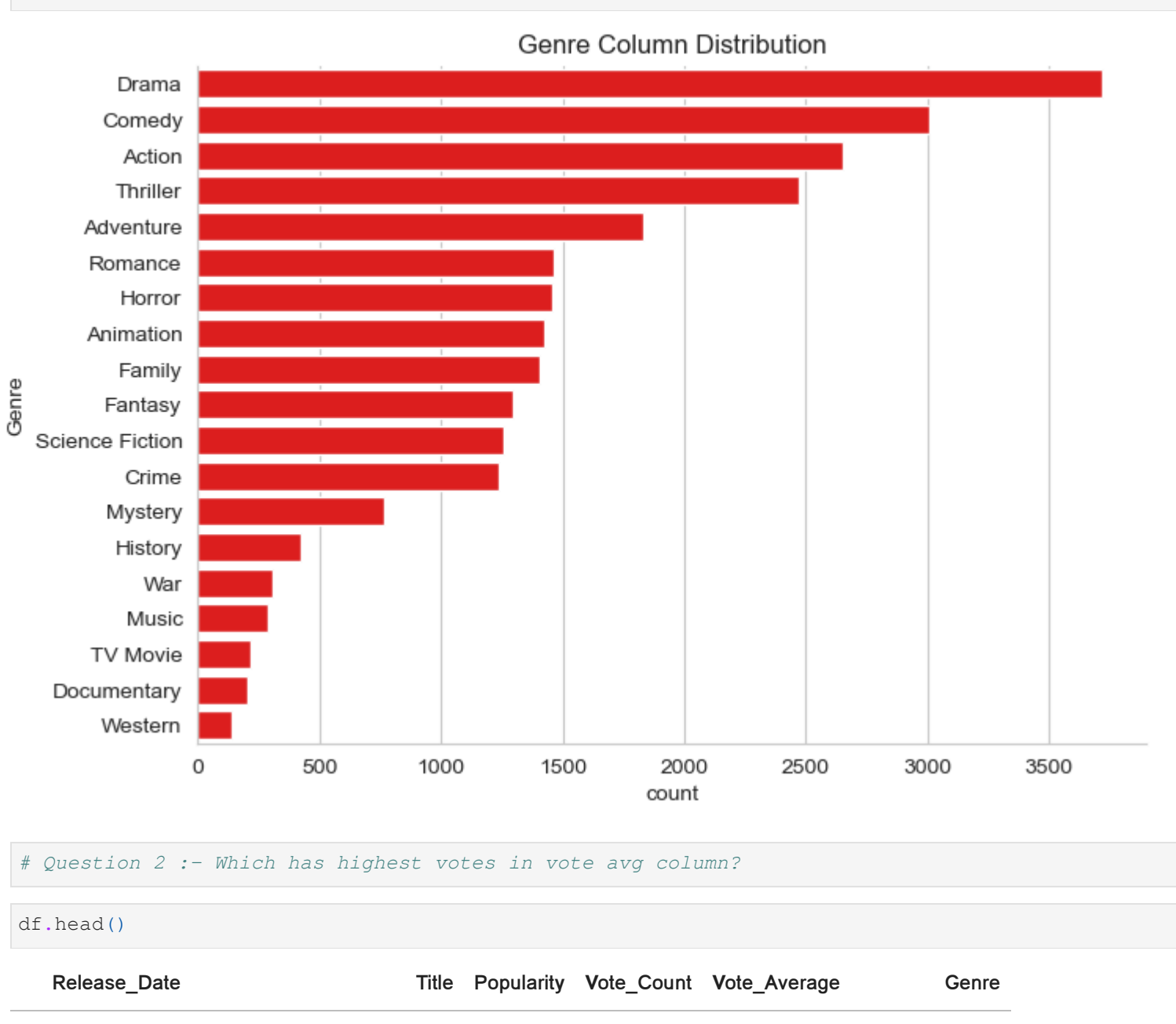
In [ ]:
```
QUESTION 1 :- What is the most frequent genre of movies released on Netflix?
```

In [41]:
```python
df['Genre'].describe()
```

Out[41]:
```
count      25552
unique        19
top        Drama
freq        3715
Name: Genre, dtype: object
```

In [105…
```python
sns.catplot( y = 'Genre' , data = df , kind = 'count' ,
            order = df['Genre'].value_counts().index ,
            color = '#FF0000' , aspect = 1.5)

plt.title('Genre Column Distribution')
plt.show()
```



In [ ]:
```python
# Question 2 :- Which has highest votes in vote avg column?
```

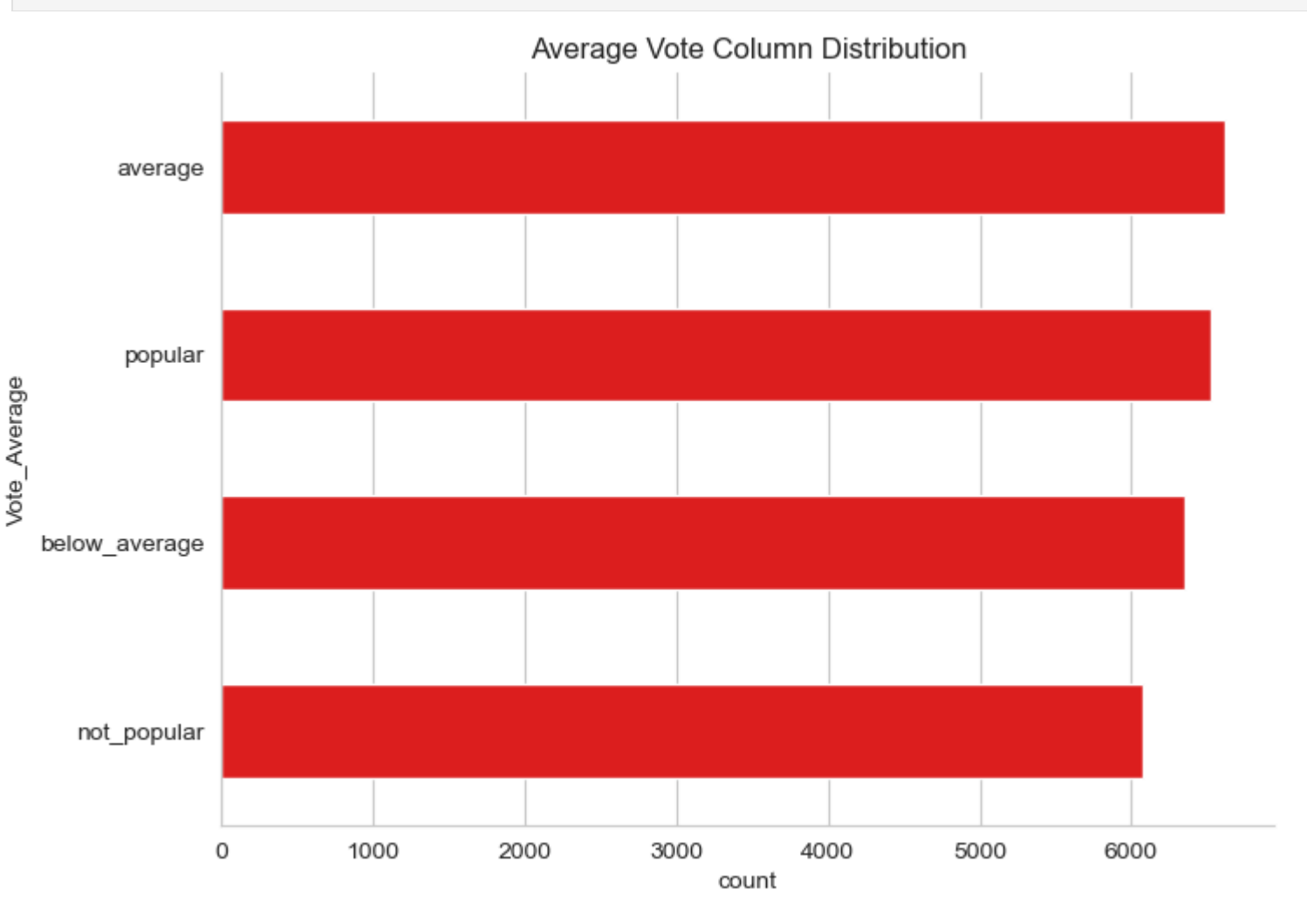
In [79]:
```python
df.head()
```

Out[79]:

	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 [117…
```python
sns.catplot(y = 'Vote_Average' , data = df , kind = 'count',
           order =  df['Vote_Average'].value_counts().index ,
           color = '#FF0000' , aspect = 1.5 , width = 0.5)

plt.title('Average Vote Column Distribution')
plt.show()
```



In [ ]:
```python
# QUESTION 3 :- What movie got the highest popularity? what's its genre ?
```

In [119…
```python
df.head(2)
```

Out[119…

	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

In [121…
```python
df[df['Popularity'] == df['Popularity'].max()]
```

Out[121…

	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

In [ ]:
```python
# QUESTION 4 :- What movie got the lowest popularity? what's its genre?
```

In [123…
```python
df[df['Popularity'] == df['Popularity'].min()]
```

Out[123…

	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

In [ ]:
```python
# Question 5 :- Which year has the most filmmed movies ?
```

In [127…
```python
df['Release_Date'].hist(color = '#FF0000')
plt.title('Release Date column Distribution')
plt.show()
```

In [ ]:
```
Conclusion

Q:-    What is the most frequent genre in the dataset ?

Ans :- 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 ?

Ans :- 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 genre ?

Ans :- Spider-Man :- No Way Home has the highest popularity rate in our dataset and it has genres of Action ,
       Adventure and Sience Fiction .

Q4 :-  What movie got the lowest popularity ? what's its genre ?

Ans :- The united states, thread has the highest lowest rate in our dataset and it has genres of music , drama ,
       'war', 'sci-fi' and history`.

Q5  :-  Which year has the most filmmed movies?

Ans :- year 2020 has the highest filmming rate in our dataset.
```

In [ ]:
```python
# Link of ChatGPT explaining how the conclusion is reached :- only read the last theory

#LINK

https://chatgpt.com/share/67a091ad-2c30-800a-b633-ee1728effa2f
```