In [25]:
```
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')
```

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

Out[26]:

	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 [27]:
```
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 [28]:
```
df['Genre'].head()
```

Out[28]:
```
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 [29]:
```
df.describe()
```

Out[29]:

	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 columns.
- Our dataset looks a nit tidy with no NaN nor duplicated values.

- Release_Date column needs to be casted into date time format and to extract only year value
- Overview and POster_Url wouldn't be so useful during analysis.
- There is noticable outlierws in popularity column
- Vote_average better be categorised for proper analysis
- Genre column has comma separated valued and white spaces that needs to be handled

Data cleaning

- Casting Release_date column and extracting year only

In [30]:
```
df.head()
```

Out[30]:

	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 [31]:
```
df['Release_Date']=pd.to_datetime(df['Release_Date'])
```

In [32]:
```
df['Release_Date']=df['Release_Date'].dt.year
df['Release_Date'].dtypes
```

Out[32]: dtype('int32')

In [33]:
```
df.head()
```

Out[33]:

	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 [34]:
```
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
```

Dropping Overview and Poster Url

In [35]:
```
cols = ['Overview', 'Poster_Url']
df.drop(cols, axis = 1, inplace= True)
df.columns
```

Out[35]:
```
Index(['Release_Date', 'Title', 'Popularity', 'Vote_Count', 'Vote_Average',
       'Original_Language', 'Genre'],
      dtype='object')
```

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

Out[36]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	G
0	2021	Spider-Man: No Way Home	5083.954	8940	8.3	en	Ac Adven Sci Fi
1	2022	The Batman	3827.658	1151	8.1	en	Cr Mys Th
2	2022	No Exit	2618.087	122	6.3	en	Th
3	2021	Encanto	2402.201	5076	7.7	en	Anima Com Fa Far
4	2021	The King's Man	1895.511	1793	7.0	en	Ac Adven Thr

Categorizing Vote_Average column: Here we would cut Vote_average values and make 4 categories popular, average, below_avg, notp_popular to describe it more using catigorize_col() function provided below

In [40]:
```
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 [42]:
```
#define lables for edges
labels = ['not_popular', 'below_avg', 'average', 'popular']
```

```
#categorize columns based on edges and labels
catigorize_col(df, 'Vote_Average', labels)

#confirming change
df['Vote_Average'].unique()
```

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

In [43]:
```
df.head()
```

Out[43]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	G
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Ac Adven Sci Fi
1	2022	The Batman	3827.658	1151	popular	en	Cr Mys Th
2	2022	No Exit	2618.087	122	below_avg	en	Th
3	2021	Encanto	2402.201	5076	popular	en	Anima Com Fa Far
4	2021	The King's Man	1895.511	1793	average	en	Ac Adven Thr

In [45]:
```
df['Vote_Average'].value_counts()
```

Out[45]:
```
Vote_Average
not_popular    2467
popular        2450
average        2412
below_avg      2398
Name: count, dtype: int64
```

In [46]:
```
df.dropna(inplace=True)

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

Out[46]:
```
Release_Date         0
Title                0
Popularity           0
Vote_Count           0
Vote_Average         0
Original_Language    0
Genre                0
dtype: int64
```

We'd split genres into a list and then explode our dataframe to have on;ly on genre per row for each movie

In [48]:
```
df['Genre'] = df['Genre'].str.split(', ')
df = df.explode('Genre').reset_index(drop=True)
df.head()
```

Out[48]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	Ge
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Act
1	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Advent
2	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Scie Fict
3	2022	The Batman	3827.658	1151	popular	en	Cr
4	2022	The Batman	3827.658	1151	popular	en	Myst

In [50]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25552 entries, 0 to 25551
Data columns (total 7 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   Original_Language  25552 non-null  object
 6   Genre              25552 non-null  object
dtypes: category(1), float64(1), int32(1), int64(1), object(3)
memory usage: 1.1+ MB
```

In [51]:
```
#casting columns into category
df['Genre'] = df['Genre'].astype('category')
df['Genre'].dtypes
```

Out[51]:
```
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 [54]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25552 entries, 0 to 25551
Data columns (total 7 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   Original_Language  25552 non-null  object
 6   Genre              25552 non-null  category
dtypes: category(2), float64(1), int32(1), int64(1), object(2)
memory usage: 949.2+ KB
```

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

Out[55]:
```
Release_Date          100
Title                9415
Popularity           8088
Vote_Count           3265
Vote_Average            4
Original_Language      42
Genre                  19
dtype: int64
```

In [56]:
```
df.head()
```

Out[56]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	Ge
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Act
1	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Advent
2	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Scie Fict
3	2022	The Batman	3827.658	1151	popular	en	Cr
4	2022	The Batman	3827.658	1151	popular	en	Myst

Data Visualization

In [65]:
```
#setting up seaborn configuration
sns.set_style('darkgrid')
```

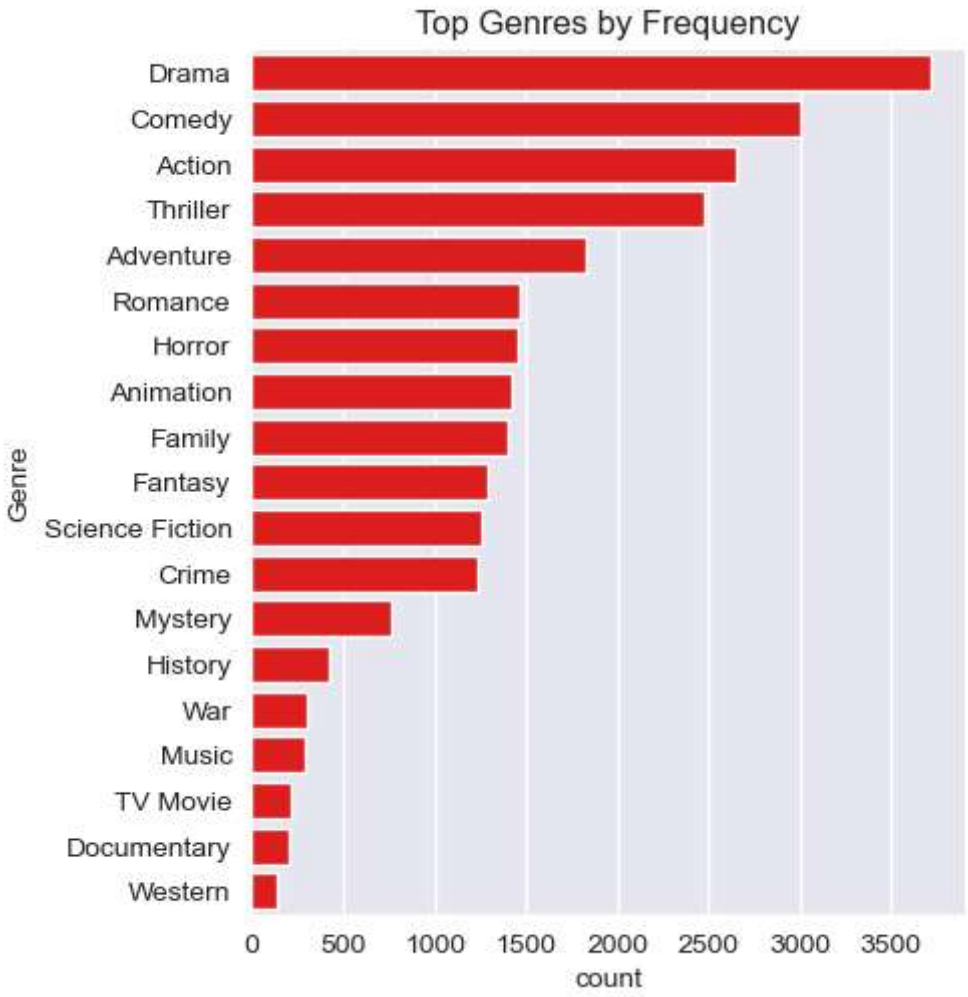
What is the most frequent genre of movies released on Netflix?

In [59]:
```
df['Genre'].describe()
```

Out[59]:
```
count     25552
unique       19
top       Drama
freq       3715
Name: Genre, dtype: object
```

In [66]:
```
sns.catplot(y = 'Genre', data = df, kind = 'count',
            order = df['Genre'].value_counts().index,
            color = 'red')

plt.title('Top Genres by Frequency')
plt.show()
```



Which genre has the highest vote ?

sns.catplot(y = 'Vote_Average', data = df, kind = 'count', order = df['Vote_Average'].value_counts().index, color = 'red')

plt.title('Vote distribution') plt.show()

What movie got the highest popularity? what's its genre?

In [75]:
```
df.head(2)
```

Out[75]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	Ge
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Act
1	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Advent

In [76]:
```
df[df['Popularity']== df['Popularity'].max()]
```

Out[76]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	Ge
0	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Act
1	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Advent
2	2021	Spider-Man: No Way Home	5083.954	8940	popular	en	Scie Fict

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

In [77]:
```
df[df['Popularity']== df['Popularity'].min()]
```

Out[77]:

	Release_Date	Title	Popularity	Vote_Count	Vote_Average	Original_Language	
25546	2021	The United States vs. Billie Holiday	13.354	152	average	en	
25547	2021	The United States vs. Billie Holiday	13.354	152	average	en	[
25548	2021	The United States vs. Billie Holiday	13.354	152	average	en	H
25549	1984	Threads	13.354	186	popular	en	
25550	1984	Threads	13.354	186	popular	en	[
25551	1984	Threads	13.354	186	popular	en	S F

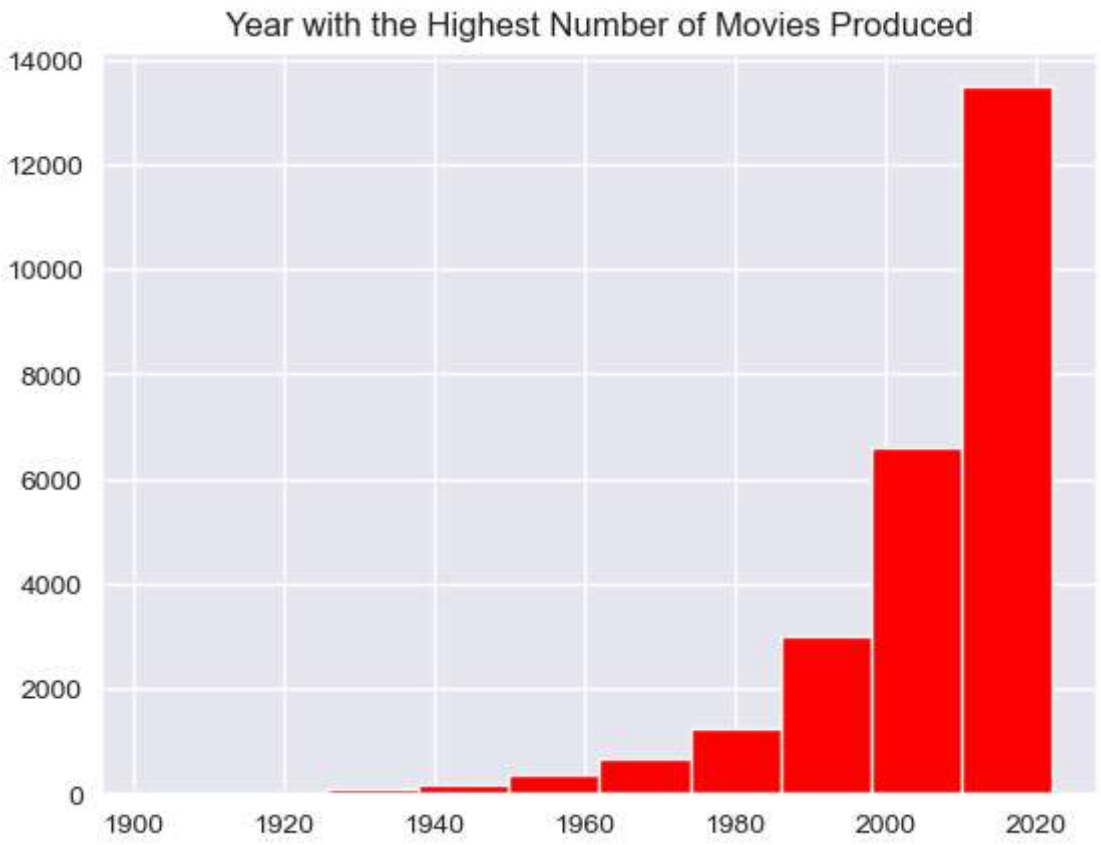
Which is the most common language?

In [87]:
```
df['Original_Language'].value_counts().head(1)
```

Out[87]:
```
Original_Language
en    19844
Name: count, dtype: int64
```

Which year has the most filmmed movies?

In [88]:
```
df['Release_Date'].hist(color='red')
plt.title("Year with the Highest Number of Movies Produced")
plt.show()
```



Conclusion

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 genre ? 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 ?* 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 is the most common language? English is the most commom language.

Q6: Which year has the most filmmed movies? Year 2020 has the highest filmming rate in our dataset.

In [89]:
```
df.to_csv('processed_movies.csv', index=False)
```

In [ ]: