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

df=pd.read_csv('mymoviedb.csv', lineterminator= '\n')
df2 = df.copy() # Create an explicit working copy
df.head()
```

```
  Release_Date                    Title  \
0   2021-12-15  Spider-Man: No Way Home
1   2022-03-01               The Batman
2   2022-02-25                  No Exit
3   2021-11-24                  Encanto
4   2021-12-22            The King's Man

                                            Overview  Popularity
Vote_Count  \
0  Peter Parker is unmasked and no longer able to...    5083.954
8940
1  In his second year of fighting crime, Batman u...    3827.658
1151
2  Stranded at a rest stop in the mountains durin...    2618.087
122
3  The tale of an extraordinary family, the Madri...    2402.201
5076
4  As a collection of history's worst tyrants and...    1895.511
1793

   Vote_Average Original_Language                                  Genre
\
0           8.3                en  Action, Adventure, Science Fiction

1           8.1                en            Crime, Mystery, Thriller

2           6.3                en                            Thriller

3           7.7                en  Animation, Comedy, Family, Fantasy

4           7.0                en    Action, Adventure, Thriller, War


                                          Poster_Url
0  https://image.tmdb.org/t/p/original/1g0dhYtq4i...
1  https://image.tmdb.org/t/p/original/74xTEgt7R3...
2  https://image.tmdb.org/t/p/original/vDHsLnOWKl...
3  https://image.tmdb.org/t/p/original/4j0PNHkMr5...
4  https://image.tmdb.org/t/p/original/aq4Pwv5Xeu...
```

# Data Pre-Processing

```
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
```

```
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
```

**Checking duplicates**

```
df['Title'].duplicated().sum()
```

```
np.int64(314)
```

```
df=df.drop_duplicates(subset='Title')
```

```
df['Title'].duplicated().sum()
```

```
np.int64(0)
```

**Checking null values**

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

```
Release_Date          0
Title                 0
Overview              0
Popularity            0
Vote_Count            0
Vote_Average          0
Original_Language     0
Genre                 0
Poster_Url            0
dtype: int64
```

**We'd split genres into a list and then explode our dataframe to have only one genre per row for ezch movie**

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

```
      Release_Date                                      Title  \
0       2021-12-15               Spider-Man: No Way Home
1       2021-12-15               Spider-Man: No Way Home
2       2021-12-15               Spider-Man: No Way Home
3       2022-03-01                            The Batman
4       2022-03-01                            The Batman
...            ...                                   ...
24936   2021-03-31  The United States vs. Billie Holiday
24937   2021-03-31  The United States vs. Billie Holiday
24938   1984-09-23                               Threads
24939   1984-09-23                               Threads
24940   1984-09-23                               Threads

                                                Overview
Popularity  \
0      Peter Parker is unmasked and no longer able to...    5083.954

1      Peter Parker is unmasked and no longer able to...    5083.954

2      Peter Parker is unmasked and no longer able to...    5083.954

3      In his second year of fighting crime, Batman u...    3827.658

4      In his second year of fighting crime, Batman u...    3827.658

...                                                  ...         ...

24936  Billie Holiday spent much of her career being ...      13.354

24937  Billie Holiday spent much of her career being ...      13.354

24938  Documentary style account of a nuclear holocau...      13.354
```

```
24939  Documentary style account of a nuclear holocau...      13.354

24940  Documentary style account of a nuclear holocau...      13.354


       Vote_Count  Vote_Average Original_Language            Genre  \
0            8940           8.3                en           Action
1            8940           8.3                en        Adventure
2            8940           8.3                en  Science Fiction
3            1151           8.1                en            Crime
4            1151           8.1                en          Mystery
...           ...           ...               ...              ...
24936         152           6.7                en            Drama
24937         152           6.7                en          History
24938         186           7.8                en              War
24939         186           7.8                en            Drama
24940         186           7.8                en  Science Fiction

                                              Poster_Url
0      https://image.tmdb.org/t/p/original/1g0dhYtq4i...
1      https://image.tmdb.org/t/p/original/1g0dhYtq4i...
2      https://image.tmdb.org/t/p/original/1g0dhYtq4i...
3      https://image.tmdb.org/t/p/original/74xTEgt7R3...
4      https://image.tmdb.org/t/p/original/74xTEgt7R3...
...                                                  ...
24936  https://image.tmdb.org/t/p/original/vEzkxuE2sJ...
24937  https://image.tmdb.org/t/p/original/vEzkxuE2sJ...
24938  https://image.tmdb.org/t/p/original/lBhU4U9Eeh...
24939  https://image.tmdb.org/t/p/original/lBhU4U9Eeh...
24940  https://image.tmdb.org/t/p/original/lBhU4U9Eeh...

[24941 rows x 9 columns]
```

**Converting Release_Date column**

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24941 entries, 0 to 24940
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Release_Date       24941 non-null  object
 1   Title              24941 non-null  object
 2   Overview           24941 non-null  object
 3   Popularity         24941 non-null  float64
 4   Vote_Count         24941 non-null  int64
 5   Vote_Average       24941 non-null  float64
 6   Original_Language  24941 non-null  object
```

```
 7   Genre              24941 non-null  object
 8   Poster_Url         24941 non-null  object
dtypes: float64(2), int64(1), object(6)
memory usage: 1.7+ MB
```

```
# 2. Convert the 'Release_Date' column to datetime objects
df['Release_Date'] = pd.to_datetime(df['Release_Date'], format='%Y-%m-
%d')

# 3. Extract the year
df['Release_Year'] = df['Release_Date'].dt.year

# Display the data type of the 'Release_Year' column
print(df['Release_Year'].dtypes)
```

```
int32
  Release_Date  Release_Year
0   2021-12-15          2021
1   2021-12-15          2021
2   2021-12-15          2021
3   2022-03-01          2022
4   2022-03-01          2022
```

```
df.head()
```

```
  Release_Date                      Title  \
0   2021-12-15  Spider-Man: No Way Home
1   2021-12-15  Spider-Man: No Way Home
2   2021-12-15  Spider-Man: No Way Home
3   2022-03-01               The Batman
4   2022-03-01               The Batman

                                            Overview  Popularity
Vote_Count  \
0  Peter Parker is unmasked and no longer able to...    5083.954
8940
1  Peter Parker is unmasked and no longer able to...    5083.954
8940
2  Peter Parker is unmasked and no longer able to...    5083.954
8940
3  In his second year of fighting crime, Batman u...    3827.658
1151
4  In his second year of fighting crime, Batman u...    3827.658
1151

   Vote_Average Original_Language            Genre  \
0           8.3                en           Action
1           8.3                en        Adventure
2           8.3                en  Science Fiction
3           8.1                en           Crime
4           8.1                en          Mystery
```

```
                                          Poster_Url  Release_Year
0  https://image.tmdb.org/t/p/original/1g0dhYtq4i...          2021
1  https://image.tmdb.org/t/p/original/1g0dhYtq4i...          2021
2  https://image.tmdb.org/t/p/original/1g0dhYtq4i...          2021
3  https://image.tmdb.org/t/p/original/74xTEgt7R3...          2022
4  https://image.tmdb.org/t/p/original/74xTEgt7R3...          2022
```

**Removing unnecessary columns**

```
columns_to_remove =
['Overview','Original_Language','Poster_Url','Release_Date']  # List
the columns to remove
df.drop(columns=columns_to_remove, axis=1, inplace=True)

df.head()
```

```
                    Title  Popularity  Vote_Count  Vote_Average  \
0  Spider-Man: No Way Home    5083.954        8940           8.3
1  Spider-Man: No Way Home    5083.954        8940           8.3
2  Spider-Man: No Way Home    5083.954        8940           8.3
3               The Batman    3827.658        1151           8.1
4               The Batman    3827.658        1151           8.1

             Genre  Release_Year
0           Action          2021
1        Adventure          2021
2  Science Fiction          2021
3            Crime          2022
4          Mystery          2022
```

**We would cut the Vote_Average values and make 4 categories: popular average below_avg not_popular to describe it more using categorize_vote_average() function.**

```
def categorize_vote_average(Vote_Average):

  if Vote_Average >= 8:
    return 'popular'
  elif Vote_Average >= 6:
    return 'average'
  elif Vote_Average >= 4:
    return 'below_avg'
  else:
    return 'not_popular'

# Apply the function to the 'Vote_Average' column
df['Vote_Category'] =
df['Vote_Average'].apply(categorize_vote_average)
df
```

```
                                    Title  Popularity  Vote_Count  \
0                   Spider-Man: No Way Home    5083.954        8940
1                   Spider-Man: No Way Home    5083.954        8940
2                   Spider-Man: No Way Home    5083.954        8940
3                                The Batman    3827.658        1151
4                                The Batman    3827.658        1151
...                                     ...         ...         ...
24936  The United States vs. Billie Holiday      13.354         152
24937  The United States vs. Billie Holiday      13.354         152
24938                               Threads      13.354         186
24939                               Threads      13.354         186
24940                               Threads      13.354         186

       Vote_Average            Genre  Release_Year Vote_Category
0               8.3           Action          2021       popular
1               8.3        Adventure          2021       popular
2               8.3  Science Fiction          2021       popular
3               8.1            Crime          2022       popular
4               8.1          Mystery          2022       popular
...             ...              ...           ...           ...
24936           6.7            Drama          2021       average
24937           6.7          History          2021       average
24938           7.8              War          1984       average
24939           7.8            Drama          1984       average
24940           7.8  Science Fiction          1984       average

[24941 rows x 7 columns]

df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24941 entries, 0 to 24940
Data columns (total 7 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Title          24941 non-null  object
 1   Popularity     24941 non-null  float64
 2   Vote_Count     24941 non-null  int64
 3   Vote_Average   24941 non-null  float64
 4   Genre          24941 non-null  object
 5   Release_Year   24941 non-null  int32
 6   Vote_Category  24941 non-null  object
dtypes: float64(2), int32(1), int64(1), object(3)
memory usage: 1.2+ MB

df.nunique()

Title            9513
Popularity       7945
Vote_Count       3222
```

```
Vote_Average       74
Genre              19
Release_Year      102
Vote_Category       4
dtype: int64
```

```
df.head()
```

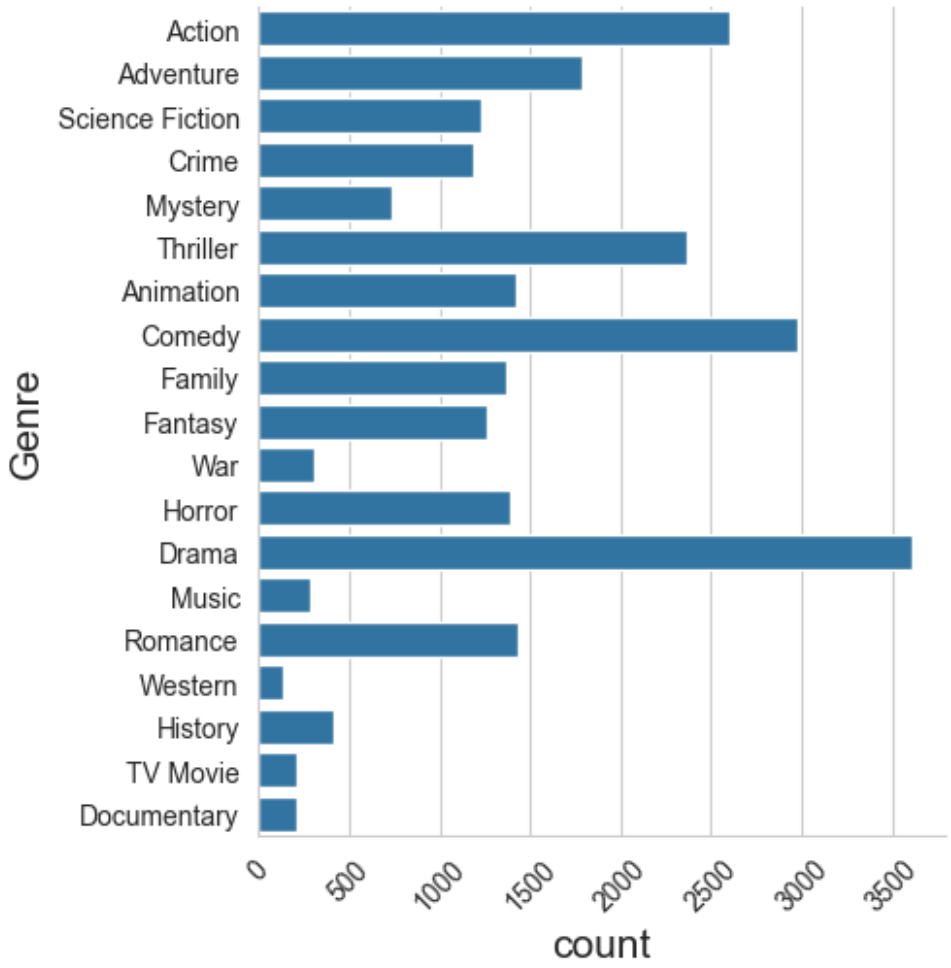
| | Title | Popularity | Vote_Count | Vote_Average | \ |
|---|---|---|---|---|---|
| 0 | Spider-Man: No Way Home | 5083.954 | 8940 | 8.3 | |
| 1 | Spider-Man: No Way Home | 5083.954 | 8940 | 8.3 | |
| 2 | Spider-Man: No Way Home | 5083.954 | 8940 | 8.3 | |
| 3 | The Batman | 3827.658 | 1151 | 8.1 | |
| 4 | The Batman | 3827.658 | 1151 | 8.1 | |

| | Genre | Release_Year | Vote_Category |
|---|---|---|---|
| 0 | Action | 2021 | popular |
| 1 | Adventure | 2021 | popular |
| 2 | Science Fiction | 2021 | popular |
| 3 | Crime | 2022 | popular |
| 4 | Mystery | 2022 | popular |

# Data Visualization

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

```
import matplotlib.pyplot as plt
import seaborn as sns
sns.catplot(data=df,y='Genre',kind='count')
plt.xticks(rotation=45)
plt.xlabel('count',size=15)
plt.ylabel('Genre',size=15)
sns.set_style(style='whitegrid')
plt.show()
```




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

```
Genre
Drama              3610
Comedy             2975
Action             2600
Thriller           2368
Adventure          1790
Romance            1431
Animation          1420
Horror             1389
Family             1369
Fantasy            1257
Science Fiction    1234
Crime              1191
Mystery             740
History             417
War                 302
Music               290
Documentary         215
```

```
TV Movie            210
Western             133
Name: count, dtype: int64
```

**Drama is the most frequent genre of the movies released on Netflix**

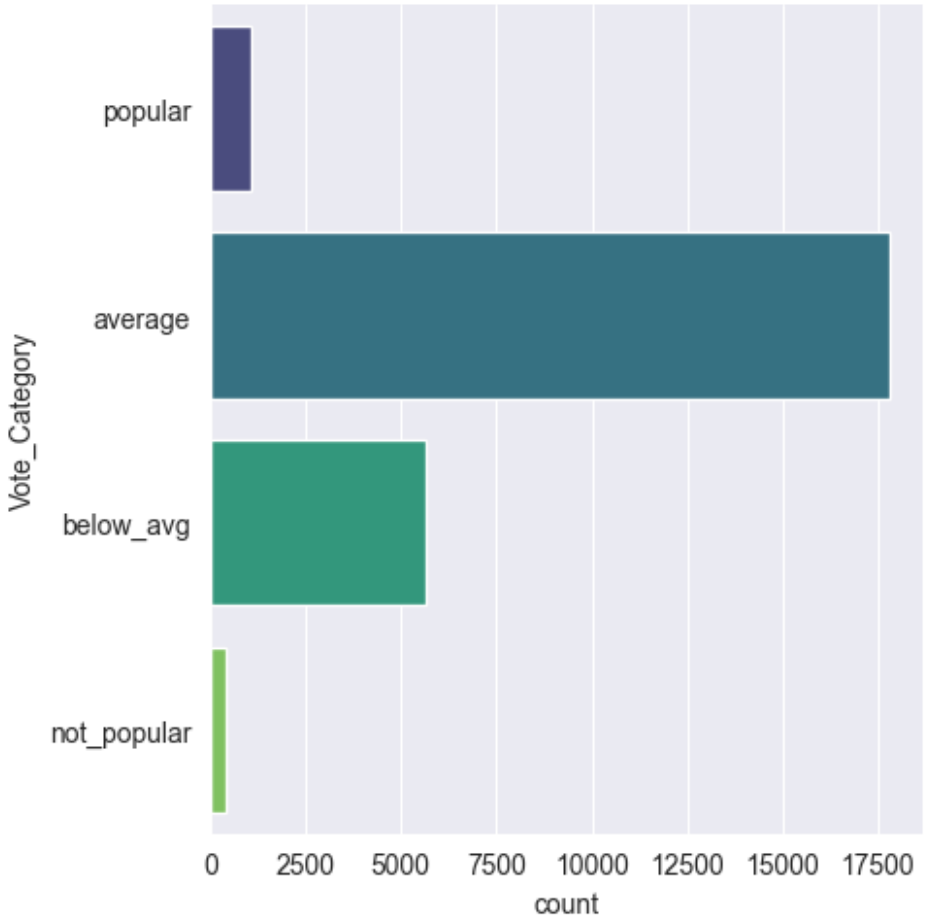
# Which has the highest votes in the Vote_Average column?

```
df.head()
```

| | Title | Popularity | Vote_Count | Vote_Average | Genre | Release_Year | Vote_Category |
|---|---|---|---|---|---|---|---|
| 0 | Spider-Man: No Way Home | 5083.954 | 8940 | 8.3 | Action | 2021 | popular |
| 1 | Spider-Man: No Way Home | 5083.954 | 8940 | 8.3 | Adventure | 2021 | popular |
| 2 | Spider-Man: No Way Home | 5083.954 | 8940 | 8.3 | Science Fiction | 2021 | popular |
| 3 | The Batman | 3827.658 | 1151 | 8.1 | Crime | 2022 | popular |
| 4 | The Batman | 3827.658 | 1151 | 8.1 | Mystery | 2022 | popular |

**we have categorize the vote_Average into Vote_categories which going to help us to plot graph more efficiently**

```
sns.catplot(y='Vote_Category',data=df,kind='count',palette='viridis',h
ue='Vote_Category',legend=False)
sns.set_style(style='darkgrid')
plt.show()
```




```
df['Vote_Category'].value_counts()
```

```
Vote_Category
average        17778
below_avg       5666
popular         1103
not_popular      394
Name: count, dtype: int64
```

**The distribution of votes across different vote_average categories reveals that the highest concentration of votes is observed within the average vote category.**

# Which movie got the highest popularity and what it's Genre

```
df.head()
```

```
                     Title  Popularity  Vote_Count  Vote_Average  \
0  Spider-Man: No Way Home    5083.954        8940           8.3
```

```
1  Spider-Man: No Way Home    5083.954        8940           8.3
2  Spider-Man: No Way Home    5083.954        8940           8.3
3               The Batman    3827.658        1151           8.1
4               The Batman    3827.658        1151           8.1
```

```
             Genre  Release_Year Vote_Category
0           Action          2021       popular
1        Adventure          2021       popular
2  Science Fiction          2021       popular
3            Crime          2022       popular
4          Mystery          2022       popular
```

```
g=df.groupby(['Title','Genre']).agg({'Popularity':'max'}).sort_values(
by=['Popularity'], ascending=False)
g
```

```
                                                    Popularity
Title                                Genre
Spider-Man: No Way Home              Adventure         5083.954
                                     Science Fiction   5083.954
                                     Action            5083.954
The Batman                           Mystery           3827.658
                                     Crime             3827.658
...                                                         ...
Threads                              War                 13.354
                                     Drama               13.354
The United States vs. Billie Holiday Drama               13.354
                                     Music               13.354
                                     History             13.354

[24941 rows x 1 columns]
```

# Identification of the Most Popular Film:

**Analysis of the dataset reveals that the film "Spider-Man: No Way Home" exhibits the highest popularity, registering a value of 5083.954. This film is categorized under the genres of Adventure, Science Fiction, and Action.**

# # Which movie got the lowest popularity and what it's Genre

```
#by using groupby function
g=df.groupby(['Title','Genre']).agg({'Popularity':'min'}).sort_values(
by=['Popularity'], ascending=True)
g.head()
```

```
                                                    Popularity
Title                                 Genre
Threads                               Drama             13.354
The United States vs. Billie Holiday  Music             13.354
Threads                               War               13.354
                                      Science Fiction   13.354
The United States vs. Billie Holiday  History           13.354
```

```
#by using simple masking or filtering method
min_popularity = df['Popularity'].min()
filt = df['Popularity'] == min_popularity
result = df.loc[filt, ['Title',
'Genre','Popularity']].head().to_string(index=False)
print(result)
```

```
                              Title   Genre  Popularity
The United States vs. Billie Holiday   Music      13.354
The United States vs. Billie Holiday   Drama      13.354
The United States vs. Billie Holiday History      13.354
                            Threads     War      13.354
                            Threads   Drama      13.354
```

# Identification of the Most Unpopular Film:

**Analysis of the dataset reveals that the film "The United States vs. Billie Holiday" and "Threads" exhibits the lowest popularity, registering a value of 13.354. These films are categorized under the genres of Music, Drama, War and History.**

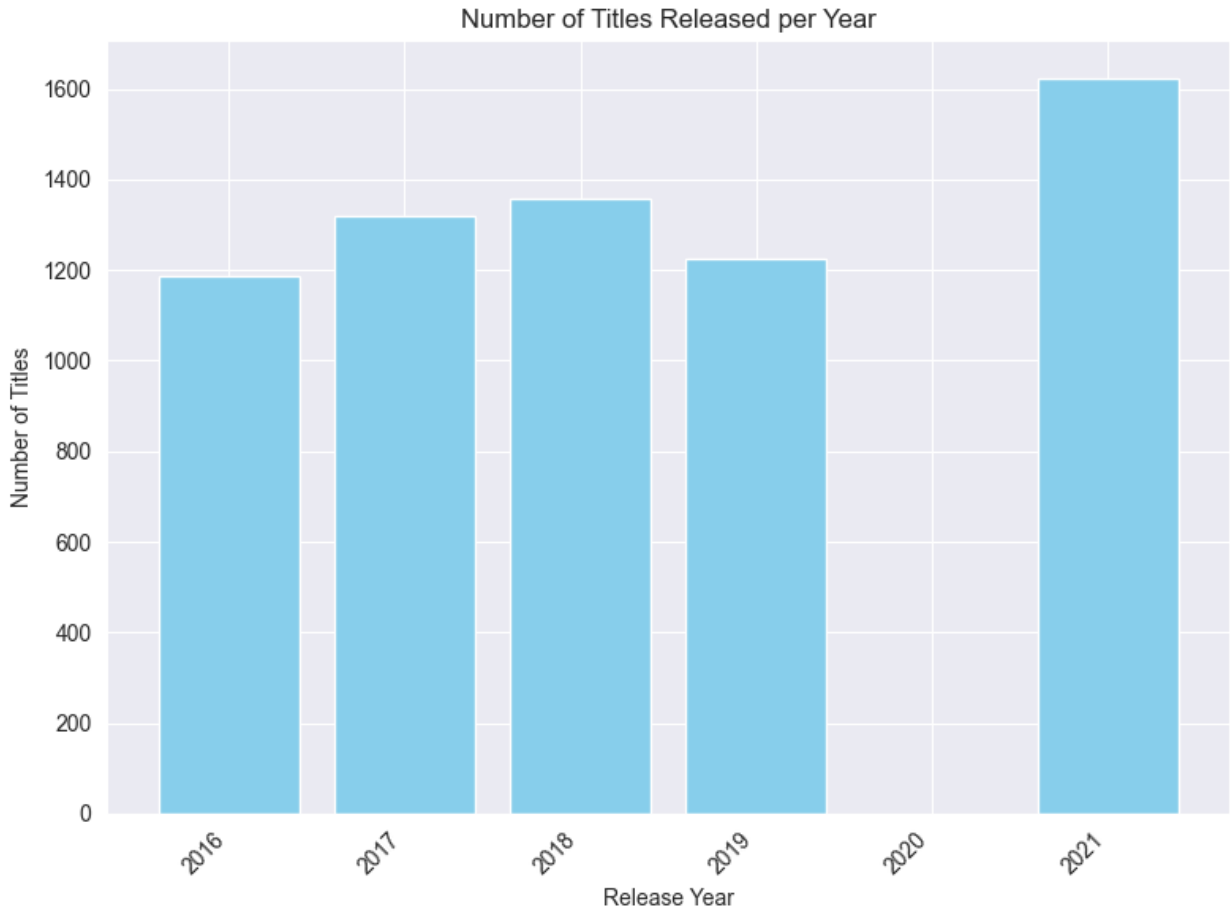
# Which year has the most filmmed Movies

```
g = df.groupby('Release_Year')
['Title'].count().sort_values(ascending=False).head()
g
```

```
Release_Year
2021    1624
2018    1358
2017    1320
2019    1225
2016    1188
Name: Title, dtype: int64
```

```
# Create the bar chart
plt.figure(figsize=(8, 6))
plt.bar(g.index, g.values, color='skyblue')
plt.xlabel("Release Year")
plt.ylabel("Number of Titles")
```

```
plt.title("Number of Titles Released per Year")
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```




**In the year 2021, Netflix demonstrated a peak in its film production output, with a total of 1624 movies released. This figure represents the highest number of films produced by Netflix within a single year in the observed dataset.**