This analysis explores Netflix's content library, focusing on key aspects such as genre distribution, popularity trends, and audience ratings. By analyzing a dataset of 9,827 entries, we aim to uncover insights into content trends, release patterns, and factors influencing viewer engagement. The goal is to provide data-driven insights that can help understand Netflix's content strategy and user preferences.

In [1]:
```
#Importing Required Libraries
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:
```
#Reading CSV File into DataFrame
df = pd.read_csv(r'D:\My projects\mymoviedb.csv', lineterminator = '\n')
```

In [4]:
```
#Displaying the First Few Rows of the Dataset
df.head()
```

Out[4]:

| | Release_Date | Title | Overview | Popularity | Vote_Count | Vote_Average | Original_Language | |
|---|---|---|---|---|---|---|---|---|
| **0** | 2021-12-15 | Spider-Man: No Way Home | Peter Parker is unmasked and no longer able to... | 5083.954 | 8940 | 8.3 | en | Adv |
| **1** | 2022-03-01 | The Batman | In his second year of fighting crime, Batman u... | 3827.658 | 1151 | 8.1 | en | N |
| **2** | 2022-02-25 | No Exit | Stranded at a rest stop in the mountains durin... | 2618.087 | 122 | 6.3 | en | |
| **3** | 2021-11-24 | Encanto | The tale of an extraordinary family, the Madri... | 2402.201 | 5076 | 7.7 | en | Ani C |
| **4** | 2021-12-22 | The King's Man | As a collection of history's worst tyrants and... | 1895.511 | 1793 | 7.0 | en | Adv |

In [5]:
```
#Dataset Overview and Data Types
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 [6]:
```
####Previewing the Genre Column
df['Genre'].head()
```

Out[6]:
```
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 [7]:
```
#Checking for Duplicate Rows in the Dataset
df.duplicated().sum()
```

Out[7]:
```
0
```

In [8]:
```
#Statistical Summary of the Dataset
df.describe()
```

Out[8]:

| | 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

-The data-set has 9,827 entries with no missing or duplicate values. -The Release Date needs to be properly formatted. -Some columns (Overview, Language, and Poster URL) may not be useful

for analysis. -The Popularity column has some unusual values. -Vote Average should be grouped for better insights. -The Genre column needs cleaning as it has extra spaces and multiple value

In [9]:
```
#Extracting Year from Date Column in DataFrame
df['Release_Date'] = pd.to_datetime(df['Release_Date'])
df['Release_Date'] = df['Release_Date'].dt.year
df['Release_Date'].dtypes
```

Out[9]:
```
dtype('int32')
```

In [10]:
```
#Dropping Specific Columns from DataFrame
cols = ['Overview', 'Original_Language', 'Poster_Url']
df.drop (cols, axis = 1, inplace = True)
df.columns
```

Out[10]:
```
Index(['Release_Date', 'Title', 'Popularity', 'Vote_Count', 'Vote_Average',
       'Genre'],
      dtype='object')
```

In [11]:
```
df
```

Out[11]:

| | 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 |
| **...** | ... | ... | ... | ... | ... | ... |
| **9822** | 1973 | Badlands | 13.357 | 896 | 7.6 | Drama, Crime |
| **9823** | 2020 | Violent Delights | 13.356 | 8 | 3.5 | Horror |
| **9824** | 2016 | The Offering | 13.355 | 94 | 5.0 | Mystery, Thriller, Horror |
| **9825** | 2021 | The United States vs. Billie Holiday | 13.354 | 152 | 6.7 | Music, Drama, History |
| **9826** | 1984 | Threads | 13.354 | 186 | 7.8 | War, Drama, Science Fiction |

9827 rows × 6 columns

Categorize Average Vote Column

In [12]:
```
#Categorizing Column Values Based on Descriptive Statistics
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], bins= edges , labels=labels, duplicates = 'drop')
    return df
```

In [13]:
```
#I have categorized the 'Vote_Average' column into four categories: 'not_popular', 'be
#based on the statistical distribution of the values, and then checked the unique cate
labels = ['not_popular', 'below_avg', 'average', 'popular']
categorize_col(df, 'Vote_Average' , labels)
df['Vote_Average'].unique()
```

Out[13]:
```
['popular', 'below_avg', 'average', 'not_popular', NaN]
Categories (4, object): ['not_popular' < 'below_avg' < 'average' < 'popular']
```

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 | 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 [15]:
```
#Counting the Frequency of Unique Values in 'Vote_Average' Column
df['Vote_Average'].value_counts()
```

Out[15]:
```
Vote_Average
not_popular    2467
popular        2450
average        2412
below_avg      2398
Name: count, dtype: int64
```

In [16]:
```
#Removing Missing Values and Checking for Remaining
df.dropna(inplace = True)
df.isna().sum
```

Out[16]:
```
<bound method NDFrame._add_numeric_operations.<locals>.sum of       Release_Date  Tit
le  Popularity  Vote_Count  Vote_Average  Genre
0            False  False       False       False         False  False
1            False  False       False       False         False  False
2            False  False       False       False         False  False
3            False  False       False       False         False  False
4            False  False       False       False         False  False
...            ...    ...         ...         ...           ...    ...
9822         False  False       False       False         False  False
9823         False  False       False       False         False  False
9824         False  False       False       False         False  False
9825         False  False       False       False         False  False
9826         False  False       False       False         False  False

[9727 rows x 6 columns]>
```

In [17]:
```
#I have split the 'Genre' column by commas and expanded it into multiple rows.
df['Genre'] = df['Genre'].str.split(', ')
df = df.explode('Genre').reset_index(drop=True)
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 |
| **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 [18]:
```
#Converting 'Genre' Column to Categorical Data Type
df['Genre'] = df['Genre'].astype('category')
df['Genre'].dtypes
```

Out[18]:
```
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)
```

In [19]:
```
#Displaying DataFrame Information
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 [20]:
```
#Counting Unique Values in Each Column
df.nunique()
```

Out[20]:
```
Release_Date     100
Title           9415
Popularity      8088
Vote_Count      3265
Vote_Average       4
Genre             19
dtype: int64
```

## Data Visualization

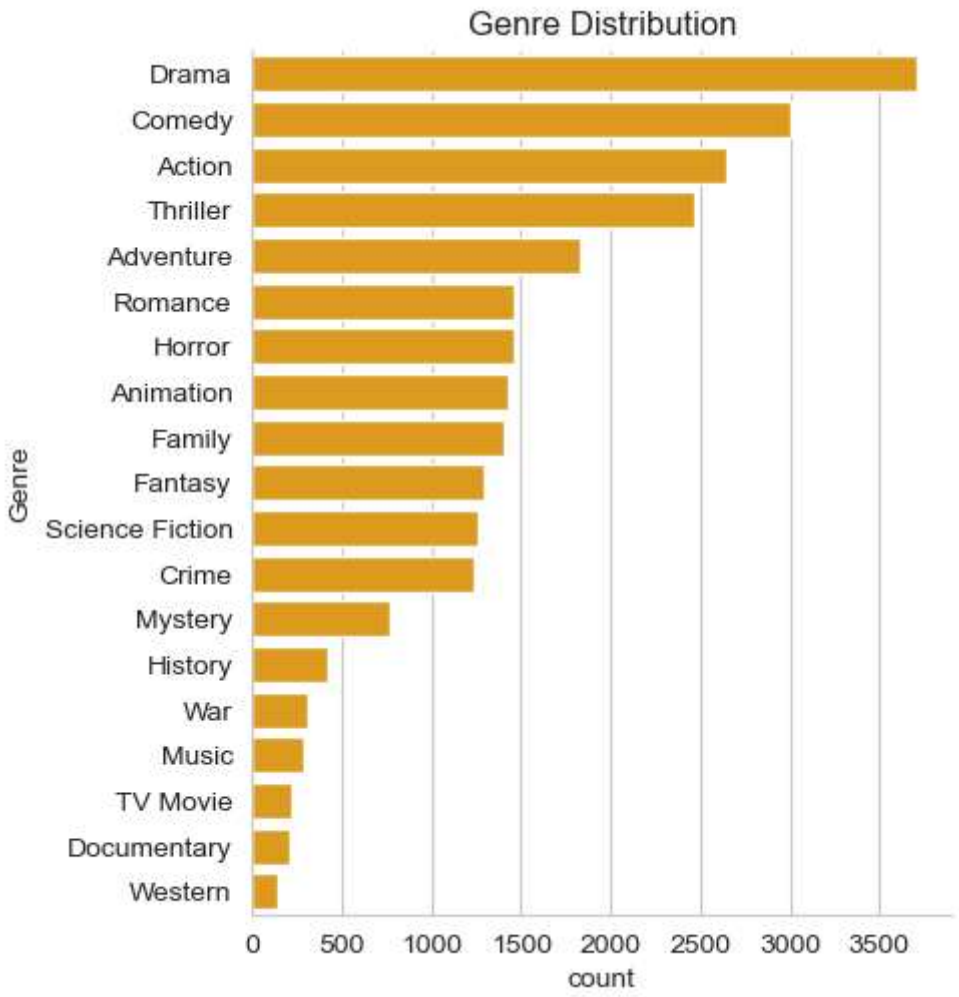
In [21]:
```
#Setting Seaborn Plot Style to 'whitegrid
sns.set_style('whitegrid')
```

In [22]:
```
#Getting Summary Statistics of 'Genre' Column
df['Genre'].describe()
```

Out[22]:
```
count     25552
unique       19
top       Drama
freq       3715
Name: Genre, dtype: object
```

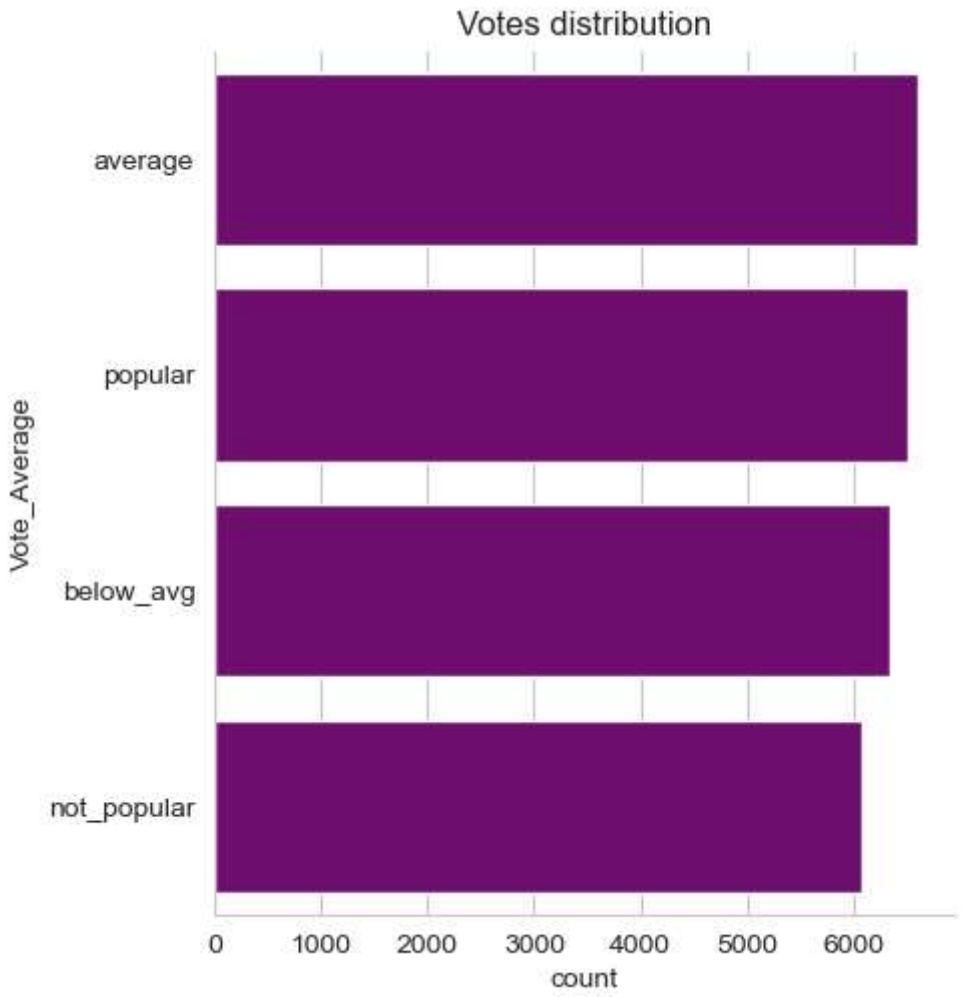
In [23]:
```
#Visualizing Genre Distribution with Count Plot
sns.catplot( y = 'Genre', data = df, kind ='count',
            order = df['Genre'].value_counts().index,
            color = 'orange')
plt.title('Genre Distribution')
plt.show()
```

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




In [24]:
```
#Visualizing Vote Distribution with Count Plot
sns.catplot( y ='Vote_Average', data=df, kind = 'count',
            order = df ['Vote_Average'].value_counts().index,
             color= 'purple')
plt.title('Votes distribution')
plt.show()
```

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




In [25]:
```
df.head(5)
```

Out[25]:

| | 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 [26]:
```
#Finding the Row with the Minimum Popularity Value
df[df['Popularity'] == df['Popularity'].min()]
```

Out[26]:

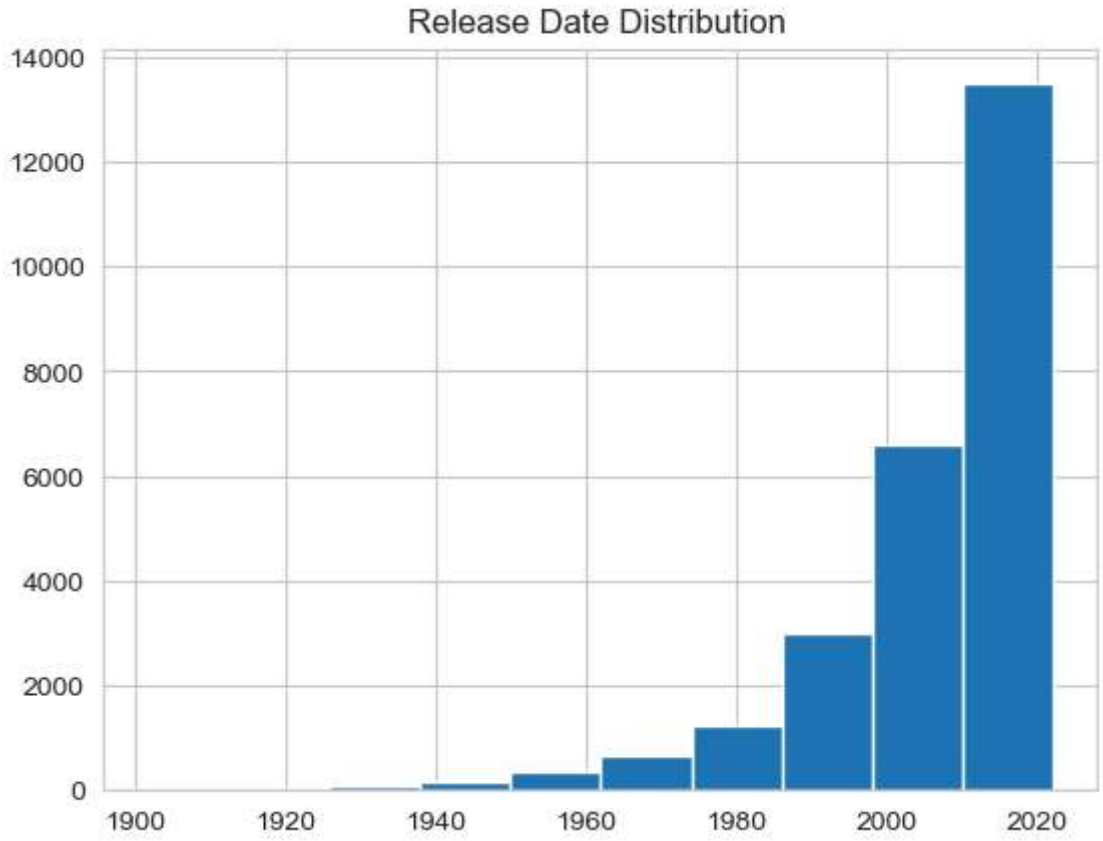
| | Release_Date | Title | Popularity | Vote_Count | Vote_Average | Genre |
|---|---|---|---|---|---|---|
| **25546** | 2021 | The United States vs. Billie Holiday | 13.354 | 152 | average | Music |
| **25547** | 2021 | The United States vs. Billie Holiday | 13.354 | 152 | average | Drama |
| **25548** | 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 [27]:
```
#Finding the Row with the Maximum Popularity Value
df[df['Popularity'] == df['Popularity'].max()]
```

Out[27]:

| | 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 [28]:
```
#Visualizing Release Date Distribution with Histogram
df['Release_Date'].hist()
plt.title('Release Date Distribution ')
plt.show()
```

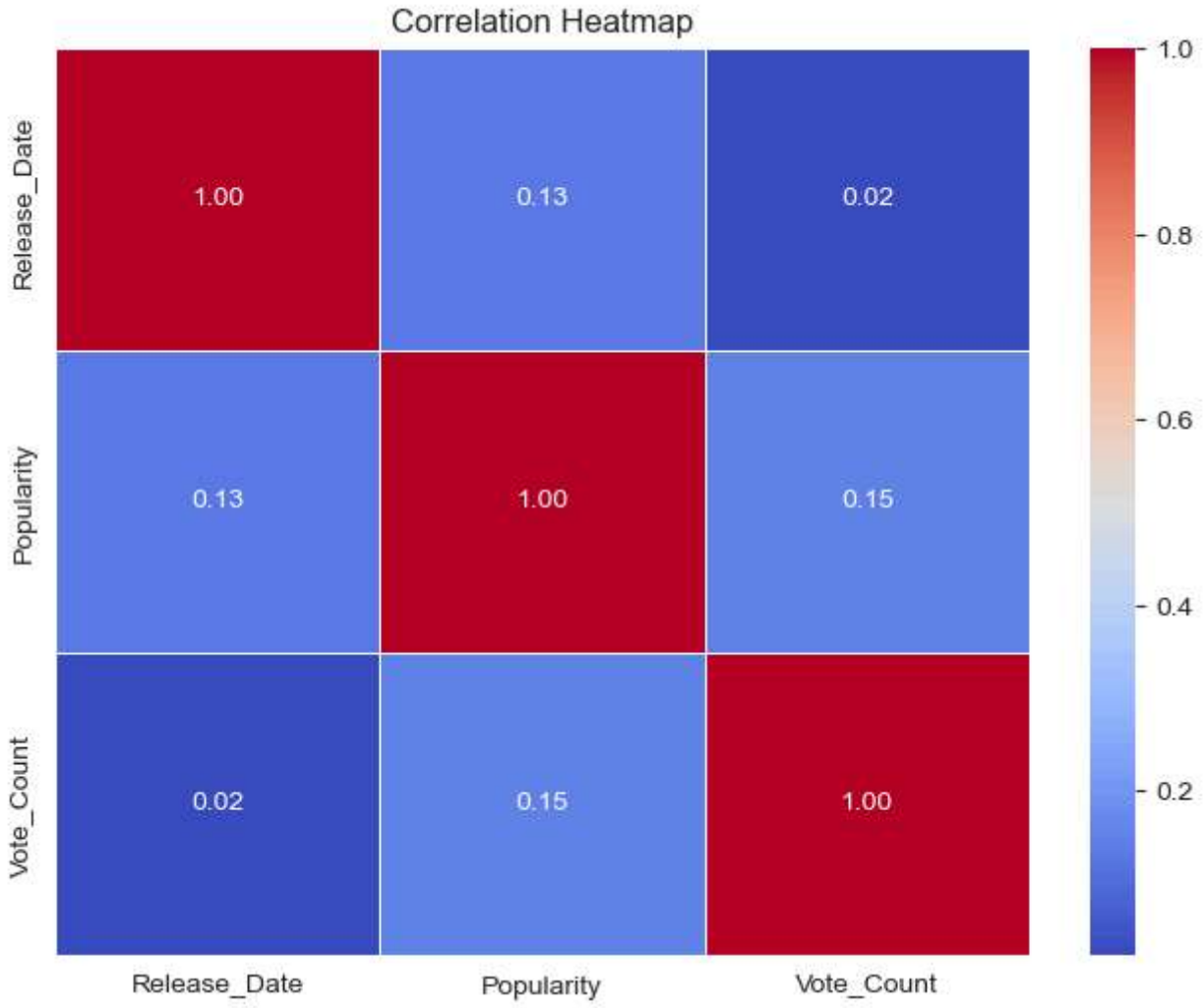



In [30]:
```
#Which Factors Influence a Movie's Popularity?

numerical_cols = ['Release_Date', 'Popularity', 'Vote_Count']

corr_matrix = df[numerical_cols].corr()

# Create the heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
plt.title("Correlation Heatmap")
plt.show()
```




## Q1: Most frequent genre

## Drama is the most frequent genre, appearing more than 14% of the time out of 19 genres.

## Q2: Genres with highest votes

## 25.5% of the dataset has popular votes (6520 rows). Drama has the highest popularity among genres, with over 18.5% of movies being popular.

## Q3: Movie with highest popularity

"Spider-Man: No Way Home" has the highest popularity. Its genres are Action, Adventure, and Science Fiction.

## Q4: Movie with lowest popularity

The United States, Thread" has the lowest popularity. Its genres are Music, Drama, War, Sci-Fi, and History.

## Q5: Year with the most movies

2020 has the most filmed movies in our dataset.

## Q6:Which Factors Influence a Movie's Popularity?

The heatmap shows that Vote Count has the strongest correlation with Popularity, while Vote Average and Release Date have weaker influences.

In [ ]:

In [ ]:

In [ ]: