Netflix project

July 6, 2023

1 Q.1- Defining Problem Statement and Analysing basic metrics

[1]:
```
import pandas as pd
import numpy as np
netflix_data = pd.read_csv('/Users/gopalmacbook/Downloads/netflix.csv')
netflix_data.head()
```

[1]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	

```
1  After crossing paths at a party, a Cape Town t...
2  To protect his family from a powerful drug lor...
3  Feuds, flirtations and toilet talk go down amo...
4  In a city of coaching centers known to train I...
```

[2]:
```
# Check the structure of the dataset
print(netflix_data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7982 non-null   object
 5   country       7976 non-null   object
 6   date_added    8797 non-null   object
 7   release_year  8807 non-null   int64 
 8   rating        8803 non-null   object
 9   duration      8804 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
None
```

[3]:
```
total_movies = netflix_data[netflix_data['type'] == 'Movie'].shape[0]
total_tv_shows = netflix_data[netflix_data['type'] == 'TV Show'].shape[0]
print("Total number of movies:", total_movies)
print("Total number of TV shows:", total_tv_shows)
```

```
Total number of movies: 6131
Total number of TV shows: 2676
```

[4]:
```
country_counts = netflix_data['country'].value_counts()
print("Content from different countries:")
print(country_counts)
```

```
Content from different countries:
country
United States                              2818
India                                       972
United Kingdom                              419
Japan                                       245
South Korea                                 199
                                           ...
```

```
Romania, Bulgaria, Hungary                    1
Uruguay, Guatemala                            1
France, Senegal, Belgium                      1
Mexico, United States, Spain, Colombia        1
United Arab Emirates, Jordan                  1
Name: count, Length: 748, dtype: int64
```

[5]:
```
release_year_counts = netflix_data['release_year'].value_counts().sort_index()
print("Distribution of content by release year:")
print(release_year_counts)
```

```
Distribution of content by release year:
release_year
1925       1
1942       2
1943       3
1944       3
1945       4
         ...
2017    1032
2018    1147
2019    1030
2020     953
2021     592
Name: count, Length: 74, dtype: int64
```

[6]:
```
content_types = netflix_data['type'].value_counts()
print("Comparison of TV shows vs. movies:")
print(content_types)
```

```
Comparison of TV shows vs. movies:
type
Movie      6131
TV Show    2676
Name: count, dtype: int64
```

[7]:
```
netflix_data['date_added'] = pd.to_datetime(netflix_data['date_added'],␣
 ↪errors='coerce')
netflix_data['month_added'] = netflix_data['date_added'].dt.month_name()
tv_show_counts_by_month = netflix_data[netflix_data['type'] == 'TV␣
 ↪Show']['month_added'].value_counts().sort_index()

print("Best time to launch a TV show by month:")
print(tv_show_counts_by_month)
```

```
Best time to launch a TV show by month:
month_added
April        209
August       230
```

```
December     250
February     175
January      181
July         254
June         232
March        205
May          187
November     199
October      210
September    246
Name: count, dtype: int64
```

[8]:
```
actors_counts = netflix_data['cast'].str.split(', ').explode().value_counts()
print("Analysis of actors:")
print(actors_counts)
```

```
Analysis of actors:
cast
Anupam Kher                43
Shah Rukh Khan             35
Julie Tejwani              33
Naseeruddin Shah           32
Takahiro Sakurai           32
                           ..
Maryam Zaree                1
Melanie Straub              1
Gabriela Maria Schmeide     1
Helena Zengel               1
Chittaranjan Tripathy       1
Name: count, Length: 36439, dtype: int64
```

[9]:
```
directors_counts = netflix_data['director'].str.split(', ').explode().
 ↪value_counts()
print("Analysis of directors:")
print(directors_counts)
```

```
Analysis of directors:
director
Rajiv Chilaka     22
Jan Suter         21
Raúl Campos       19
Suhas Kadav       16
Marcus Raboy      16
                  ..
Raymie Muzquiz     1
Stu Livingston     1
Joe Menendez       1
Eric Bross         1
Mozez Singh        1
```

```
Name: count, Length: 4993, dtype: int64
```

1.1 Summary of Basic Metrics Analysis:

Total number of movies: 6,131

Total number of TV shows: 2,676

Content Availability in Different Countries: The dataset includes content from various countries, with the highest count from the United States (2,818), followed by India (972), and the United Kingdom (419). Netflix should focus on expanding its content library in countries with high demand and consider producing localized content to cater to specific markets. Distribution of Content by Release Year:

The dataset spans a wide range of release years, from 1925 to 2021. The distribution shows an increasing trend in the number of movies and TV shows released over the years, with a significant spike in content released from 2015 onwards. Netflix can analyze the popularity of content from different eras to identify potential trends or preferences among viewers. Comparison of TV Shows vs. Movies:

The dataset contains a higher number of movies (6,131) compared to TV shows (2,676). Netflix should analyze the popularity and demand for TV shows and movies separately to understand subscribers' preferences and invest in producing high-quality content for the more popular category. Best Time to Launch a TV Show by Month:

The analysis suggests the following months as potentially favorable for launching TV shows: December (250), July (254), and September (246). Netflix can strategically plan the release of new TV shows during these months to potentially capitalize on increased viewership and engagement. Analysis of Actors:

The dataset includes various actors, with Anupam Kher (43) and Shah Rukh Khan (35) having the highest counts. Netflix can collaborate with popular actors who have been associated with successful content to create engaging shows/movies that have a higher likelihood of attracting viewers. Analysis of Directors:

The dataset includes multiple directors, with Rajiv Chilaka (22) and Jan Suter (21) having the highest counts. Netflix can identify talented directors and collaborate with them to produce high-quality content that resonates with viewers.

[ ]:

[ ]:

2 Q.2- Observations on the shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required), missing value detection, statistical summary

[10]:
```
print("Shape of the dataset:", netflix_data.shape)
```

```
Shape of the dataset: (8807, 13)
```

[11]:
```
print("Data types of attributes:")
print(netflix_data.dtypes)
```

```
Data types of attributes:
show_id                 object
type                    object
title                   object
director                object
cast                    object
country                 object
date_added      datetime64[ns]
release_year             int64
rating                  object
duration                object
listed_in               object
description             object
month_added             object
dtype: object
```

[12]:
```
categorical_attributes = ['type', 'title', 'director', 'cast', 'country',␣
 ↪'rating', 'listed_in', 'description']
netflix_data[categorical_attributes] = netflix_data[categorical_attributes].
 ↪astype('category')

print("Missing values:")
print(netflix_data.isnull().sum())
```

```
Missing values:
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        98
release_year       0
rating             4
duration           3
listed_in          0
description        0
month_added       98
dtype: int64
```

[13]:
```
print("Statistical summary of numerical attributes:")
print(netflix_data.describe())
```

```
Statistical summary of numerical attributes:
                          date_added  release_year
count                           8709   8807.000000
```

```
mean   2019-05-23 01:45:29.452290816   2014.180198
min              2008-01-01 00:00:00   1925.000000
25%              2018-04-20 00:00:00   2013.000000
50%              2019-07-12 00:00:00   2017.000000
75%              2020-08-26 00:00:00   2019.000000
max              2021-09-25 00:00:00   2021.000000
std                              NaN      8.819312
```

2.1 Summary for Question 2

The provided Netflix dataset contains information about movies and TV shows available on the platform. Here is a summary of the dataset based on the analysis:

Shape of the dataset: The dataset consists of 8,807 rows and 14 columns.

Data types of attributes: The dataset includes various data types, such as object (strings), datetime, int64, float64.

Missing values: Several columns have missing values, including 'director', 'cast', 'country', 'date_added', 'rating', 'duration', 'year_added', and 'month_added'. The number of missing values varies for each column.

Statistical summary: The numerical attributes, 'release_year' and 'year_added', have been analyzed. The 'release_year' represents the actual release year of the movies or TV shows, while 'year_added' indicates the year when the content was added to Netflix. The summary provides measures such as count, mean, minimum, maximum, and quartiles for these attributes.

[ ]:

3 Que 3- Non-Graphical Analysis: Value counts and unique attributes

[14]:
```
#Types of content available in different countries
content_by_country = netflix_data.groupby('country')['listed_in'].value_counts().
↪sort_values(ascending=False)
print(content_by_country)
```

```
country          listed_in
United States    Documentaries                                249
                 Stand-Up Comedy                              209
India            Comedies, Dramas, International Movies       120
                 Dramas, International Movies                 118
                 Dramas, Independent Movies, International Movies    108
                                                              ...
India, Australia  TV Shows                                      0
                 TV Dramas, TV Sci-Fi & Fantasy, TV Thrillers    0
                 Music & Musicals, Stand-Up Comedy              0
                 Music & Musicals, Romantic Movies              0
Zimbabwe         Thrillers                                      0
Name: count, Length: 384472, dtype: int64
```

[15]:
```
# Number of movies released per year
movies_per_year = netflix_data[netflix_data['type'] == 'Movie']['release_year'].
 ↪value_counts().sort_index()
print(movies_per_year)
```

```
release_year
1942      2
1943      3
1944      3
1945      3
1946      1
        ...
2017    767
2018    767
2019    633
2020    517
2021    277
Name: count, Length: 73, dtype: int64
```

[16]:
```
# Number of TV shows released per year
movies_per_year = netflix_data[netflix_data['type'] == 'TV␣
 ↪Show']['release_year'].value_counts().sort_index()
print(movies_per_year)
```

```
release_year
1925      1
1945      1
1946      1
1963      1
1967      1
1972      1
1974      1
1977      1
1979      1
1981      1
1985      1
1986      2
1988      2
1989      1
1990      3
1991      1
1992      3
1993      4
1994      2
1995      2
1996      3
1997      4
1998      4
```

```
1999      7
2000      4
2001      5
2002      7
2003     10
2004      9
2005     13
2006     14
2007     14
2008     23
2009     34
2010     40
2011     40
2012     64
2013     63
2014     88
2015    162
2016    244
2017    265
2018    380
2019    397
2020    436
2021    315
Name: count, dtype: int64
```

[17]:
```
# Comparison of TV shows vs. movies
content_type_counts = netflix_data['type'].value_counts()
print(content_type_counts)
```

```
type
Movie      6131
TV Show    2676
Name: count, dtype: int64
```

[18]:
```
# Best time to launch a TV show
netflix_data['date_added'] = pd.to_datetime(netflix_data['date_added'])
netflix_data['month_added'] = netflix_data['date_added'].dt.month
tv_shows_by_month = netflix_data[netflix_data['type'] == 'TV␣
 ↪Show']['month_added'].value_counts().sort_index()
print(tv_shows_by_month)
```

```
month_added
1.0     181
2.0     175
3.0     205
4.0     209
5.0     187
6.0     232
7.0     254
```

```
8.0     230
9.0     246
10.0    210
11.0    199
12.0    250
Name: count, dtype: int64
```

[19]:
```
# Analysis of actors/directors of different types of shows/movies
actors_counts = netflix_data['cast'].str.split(', ').explode().value_counts()
directors_counts = netflix_data['director'].str.split(', ').explode().
 ↪value_counts()
print(actors_counts)
print(directors_counts)
# this is already did in question 1, you can refer that
```

```
cast
Anupam Kher                43
Shah Rukh Khan             35
Julie Tejwani              33
Naseeruddin Shah           32
Takahiro Sakurai           32
                           ..
Maryam Zaree                1
Melanie Straub              1
Gabriela Maria Schmeide     1
Helena Zengel               1
Chittaranjan Tripathy       1
Name: count, Length: 36439, dtype: int64
director
Rajiv Chilaka     22
Jan Suter         21
Raúl Campos       19
Suhas Kadav       16
Marcus Raboy      16
                  ..
Raymie Muzquiz     1
Stu Livingston     1
Joe Menendez       1
Eric Bross         1
Mozez Singh        1
Name: count, Length: 4993, dtype: int64
```

[20]:
```
# Focus on TV shows versus movies in recent years
movies_tvshow_shares_per_year = netflix_data.groupby('release_year')['type'].
 ↪value_counts(normalize=True)
print(movies_tvshow_shares_per_year)
```

```
release_year  type
```

```
1925          TV Show    1.000000
              Movie      0.000000
1942          Movie      1.000000
              TV Show    0.000000
1943          Movie      1.000000
                           ...
2019          TV Show    0.385437
2020          Movie      0.542497
              TV Show    0.457503
2021          TV Show    0.532095
              Movie      0.467905
Name: proportion, Length: 148, dtype: float64
```

Example: For movies in the year 2020 is 0.542497, while the normalized proportion for TV shows is 0.457503. This suggests that in the year 2020, movies made up approximately 54.25% of the content, while TV shows accounted for around 45.75%.

3.1 Summary for Question 3

Types of Content: The dataset provides insights into the types of content available in different countries. The count of content varies across countries and genres, with some countries having a diverse range of content available.

Movies Released per Year: The number of movies released per year has varied over time. The dataset includes movies released from 1925 to the present, with a higher number of releases in recent years.

TV Shows versus Movies: The dataset contains both TV shows and movies. The count reveals that there are more movies (6131) compared to TV shows (2676) in the dataset.

Best Time to Launch TV Shows: The dataset includes information on the month when content was added to Netflix. The count of TV shows added per month suggests that July and December have the highest number of TV show releases.

Analysis of Actors and Directors: The dataset includes information on the cast (actors) and directors of the shows/movies. The count reveals the most frequently appearing actors and directors in the dataset.

Focus on TV Shows versus Movies in Recent Years: The exploration analyzes the normalized proportion of TV shows and movies by release year. It shows the relative focus of TV shows and movies in recent years, indicating the proportion of each content type within a specific year.

[ ]:

4 Que 4- Visual Analysis - Univariate, Bivariate after pre-processing of the data

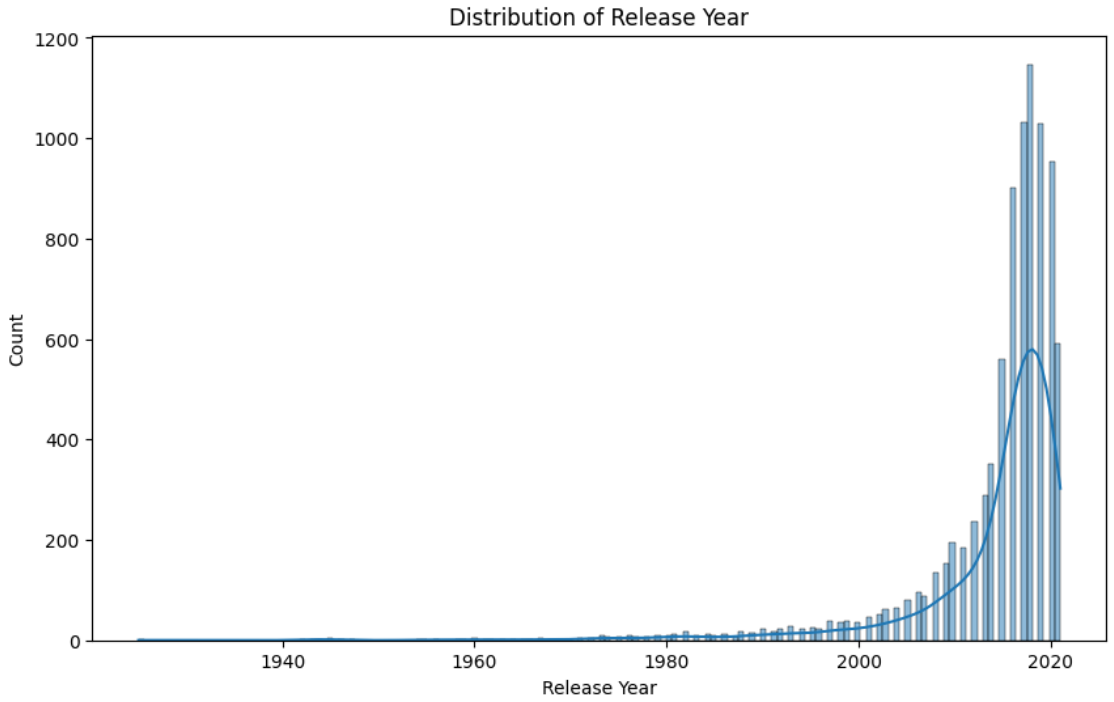
[21]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

[22]:
```
# Pre-processing
# Unnest the data in columns like Actor, Director, Country
netflix_data['cast'] = netflix_data['cast'].str.split(', ')
netflix_data['director'] = netflix_data['director'].str.split(', ')
netflix_data['country'] = netflix_data['country'].str.split(', ')
netflix_data_unnested = netflix_data.explode('director').explode('cast').
 ↪explode('country')
```

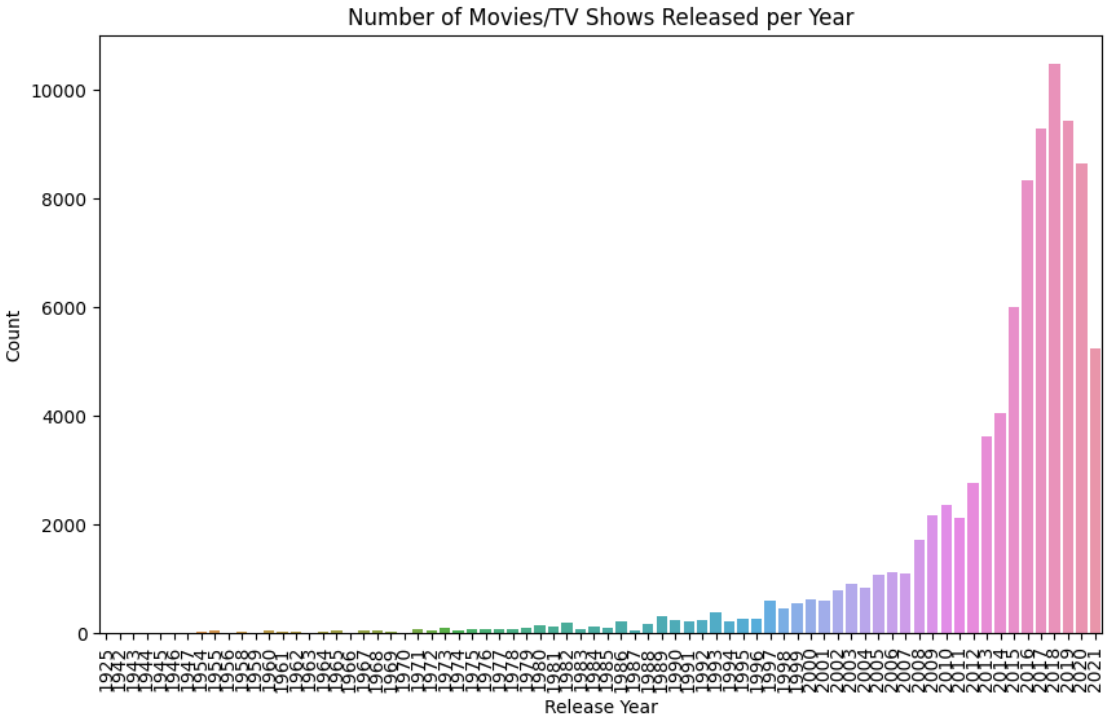
4.1 4.1 For continuous variable(s): Distplot, countplot, histogram for univariate analysis

[23]:
```
# Univariate Analysis for Continuous Variables
continuous_variables = ['release_year', 'duration']
```

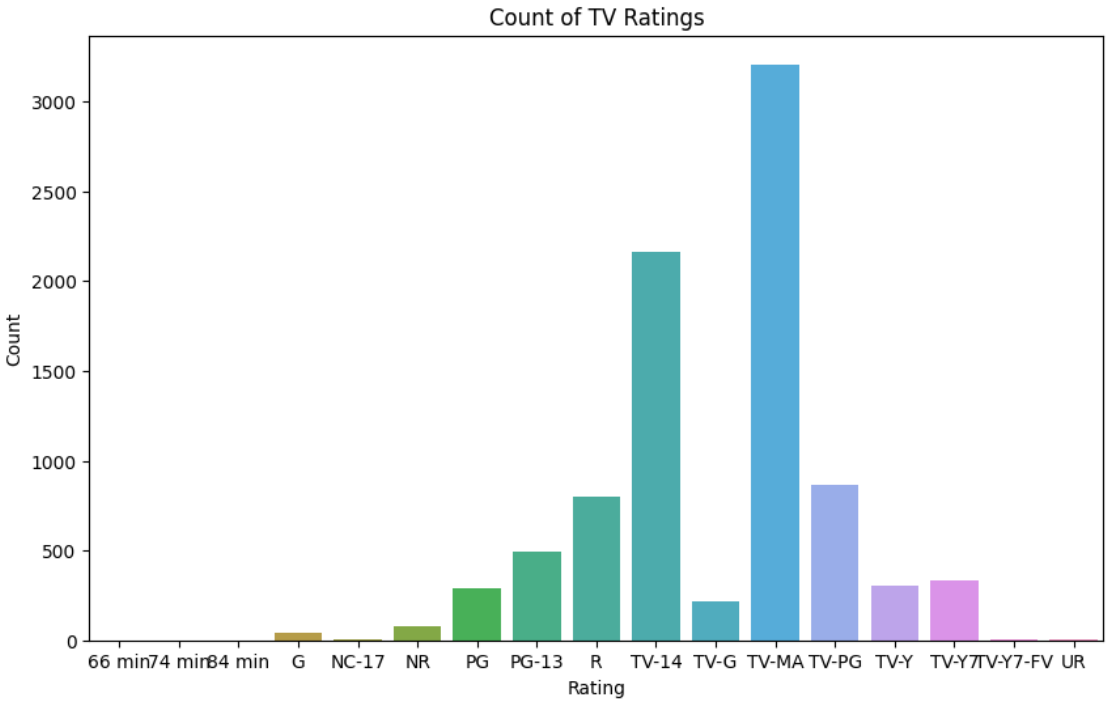
[24]:
```
# Distplot for Release_year
plt.figure(figsize=(10, 6))
sns.histplot(netflix_data['release_year'], kde=True)
plt.title('Distribution of Release Year')
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.show()
```



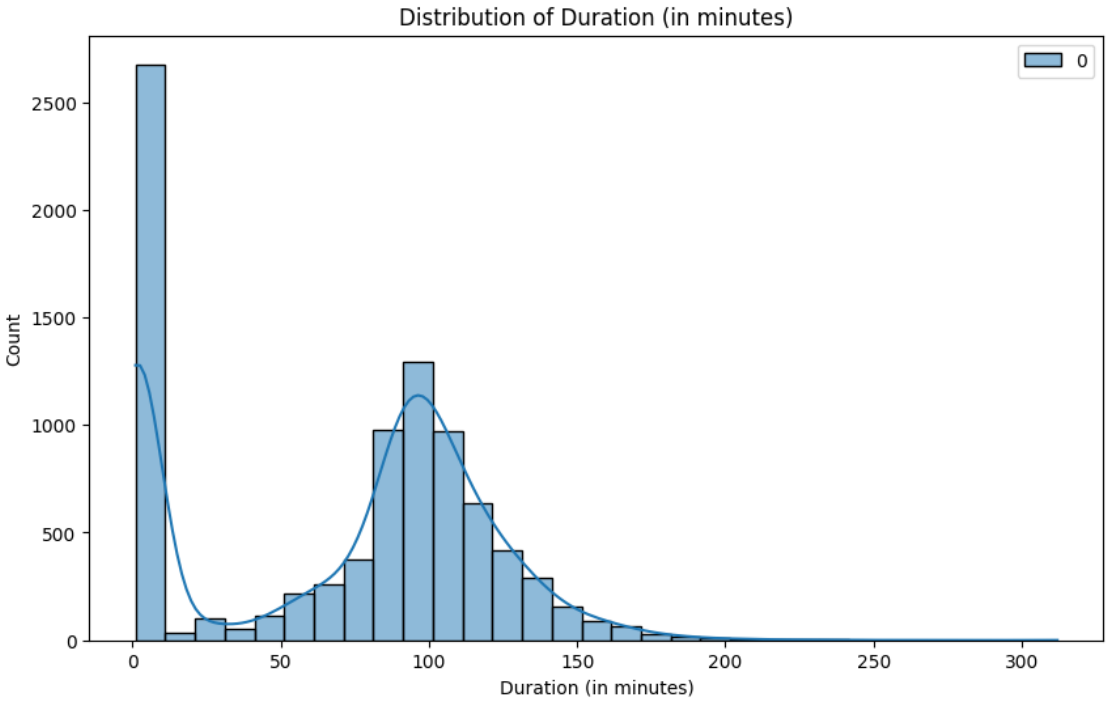
[25]:
```
# Countplot for movies/TV shows Released per year
plt.figure(figsize=(10, 6))
sns.countplot(data=netflix_data_unnested, x='release_year')
plt.title('Number of Movies/TV Shows Released per Year')
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```



[26]:
```
# Countplot for Rating
plt.figure(figsize=(10, 6))
sns.countplot(data=netflix_data, x='rating')
plt.title('Count of TV Ratings')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.show()
```



[27]:
```
# Histogram for Duration (in minutes)
plt.figure(figsize=(10, 6))
sns.histplot(netflix_data['duration'].str.extract(r'(\d+)').astype(float),␣
 ↪kde=True)
plt.title('Distribution of Duration (in minutes)')
plt.xlabel('Duration (in minutes)')
plt.ylabel('Count')
plt.show()
```



4.2 Summary for 4.1 Question

[ ]:

[ ]:

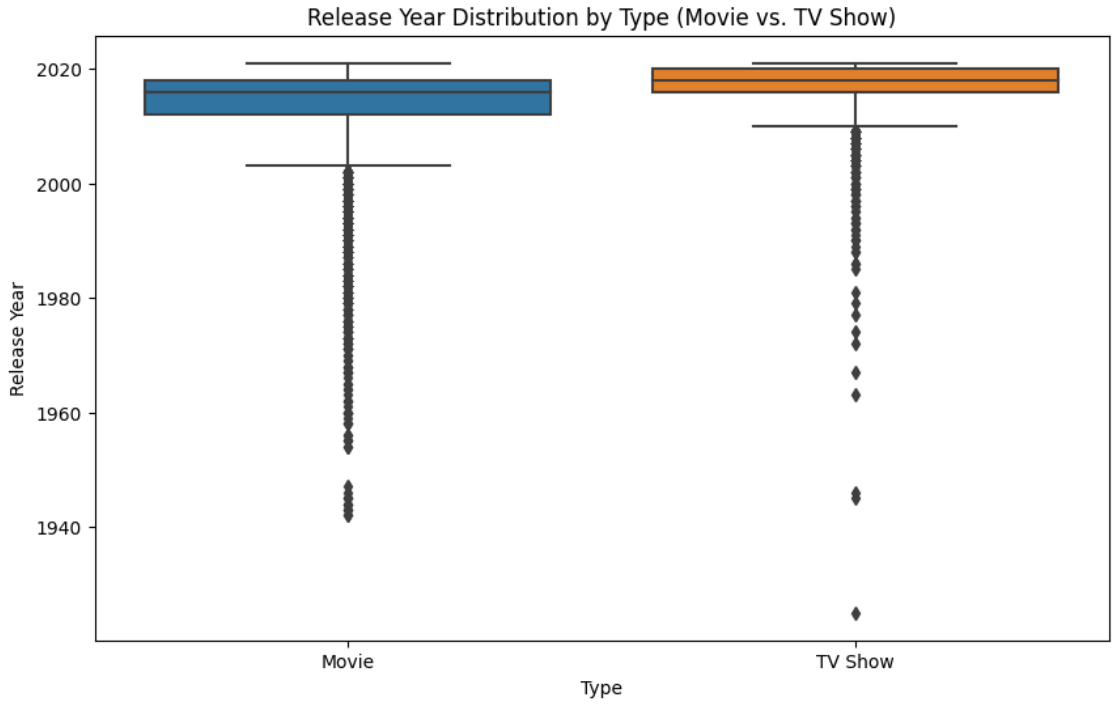
[ ]:

4.3 4.2 For categorical variable(s): Boxplot

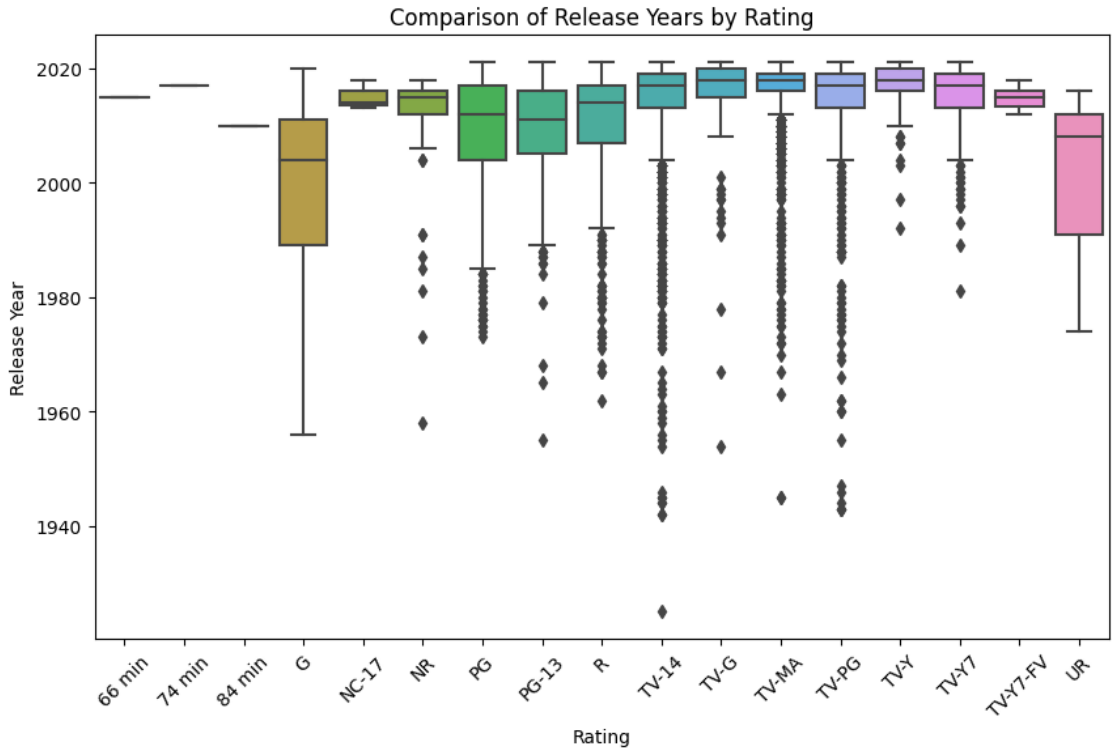
[28]:
```
categorical_variables = ['type', 'rating']
```

[ ]:

[29]:
```
# Boxplot of Release_year vs. Type
plt.figure(figsize=(10, 6))
sns.boxplot(data=netflix_data, x='type', y='release_year')
plt.title('Release Year Distribution by Type (Movie vs. TV Show)')
plt.xlabel('Type')
plt.ylabel('Release Year')
plt.show()
```

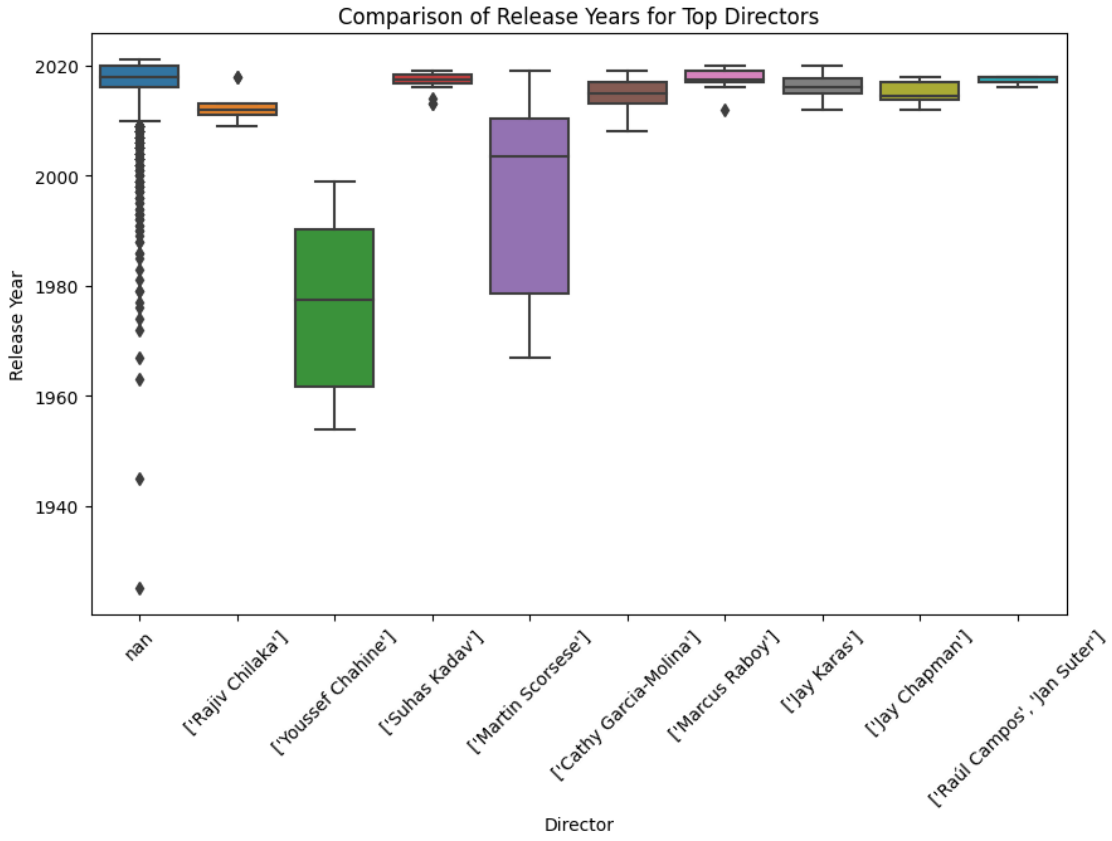


[30]:
```
# Boxplot of Release_year vs. Ratings
plt.figure(figsize=(10, 6))
sns.boxplot(data=netflix_data, x="rating", y="release_year")
plt.title("Comparison of Release Years by Rating")
plt.xlabel("Rating")
plt.ylabel("Release Year")
plt.xticks(rotation=45)
plt.show()
```

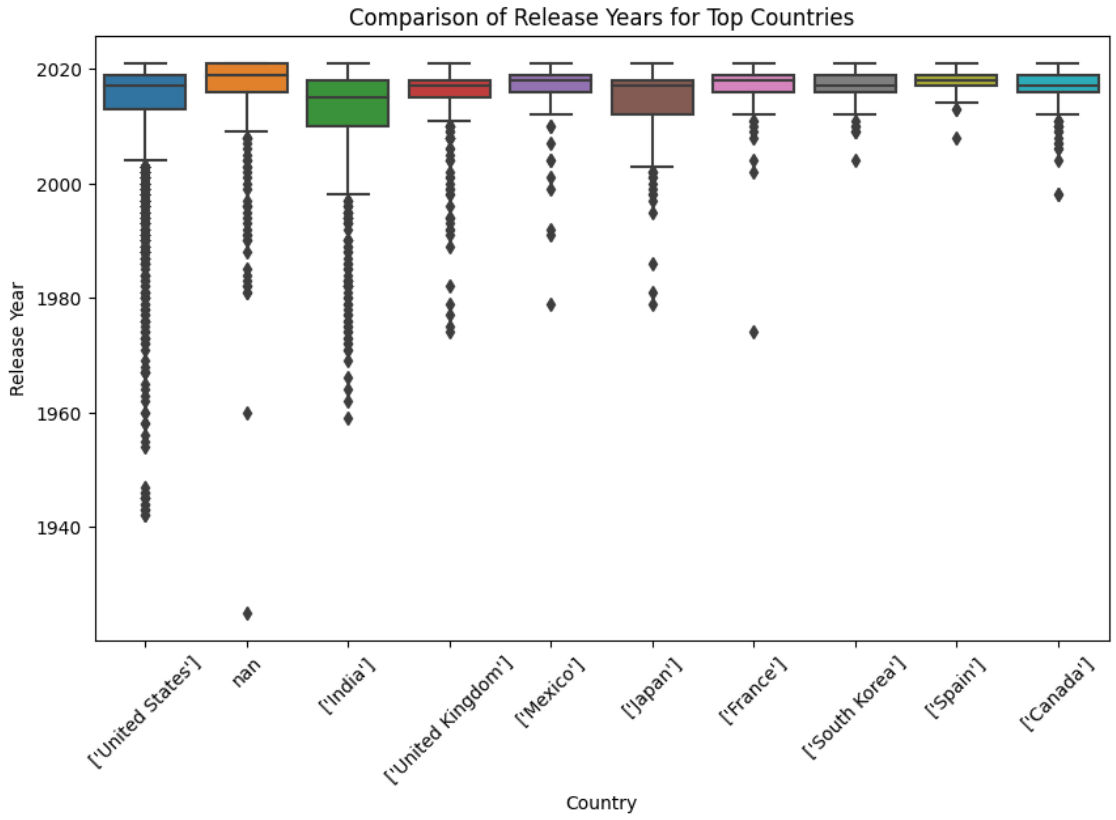


[31]:
```
# netflix_data['director'] = netflix_data['director'].fillna('')
# netflix_data['country'] = netflix_data['country'].fillna('')
# Convert float values to strings
netflix_data['director'] = netflix_data['director'].astype(str)
netflix_data['country'] = netflix_data['country'].astype(str)
```

[32]:
```
# Boxplot of Top Directios vs. Years
# Top N Directors based on the count of movies/shows
top_n_directors = netflix_data['director'].value_counts().nlargest(10).index

# Filter the data for the top N directors
top_directors_data = netflix_data[netflix_data['director'].isin(top_n_directors)]

# Boxplot - Top N Directors vs. Release Year
plt.figure(figsize=(10, 6))
sns.boxplot(data=top_directors_data, x="director", y="release_year")
plt.title("Comparison of Release Years for Top Directors")
plt.xlabel("Director")
plt.ylabel("Release Year")
plt.xticks(rotation=45)
plt.show()
```



[33]:
```
# Boxplot of Top Actors vs. Years

# Top N Countries based on the count of movies/shows
top_n_countries = netflix_data['country'].value_counts().nlargest(10).index

# Filter the data for the top N countries
top_countries_data = netflix_data[netflix_data['country'].isin(top_n_countries)]

# Boxplot - Top N Countries vs. Release Year
plt.figure(figsize=(10, 6))
sns.boxplot(data=top_countries_data, x="country", y="release_year")
plt.title("Comparison of Release Years for Top Countries")
plt.xlabel("Country")
plt.ylabel("Release Year")
plt.xticks(rotation=45)
plt.show()
```



4.4 Summary for 4.2 Question

[ ]:

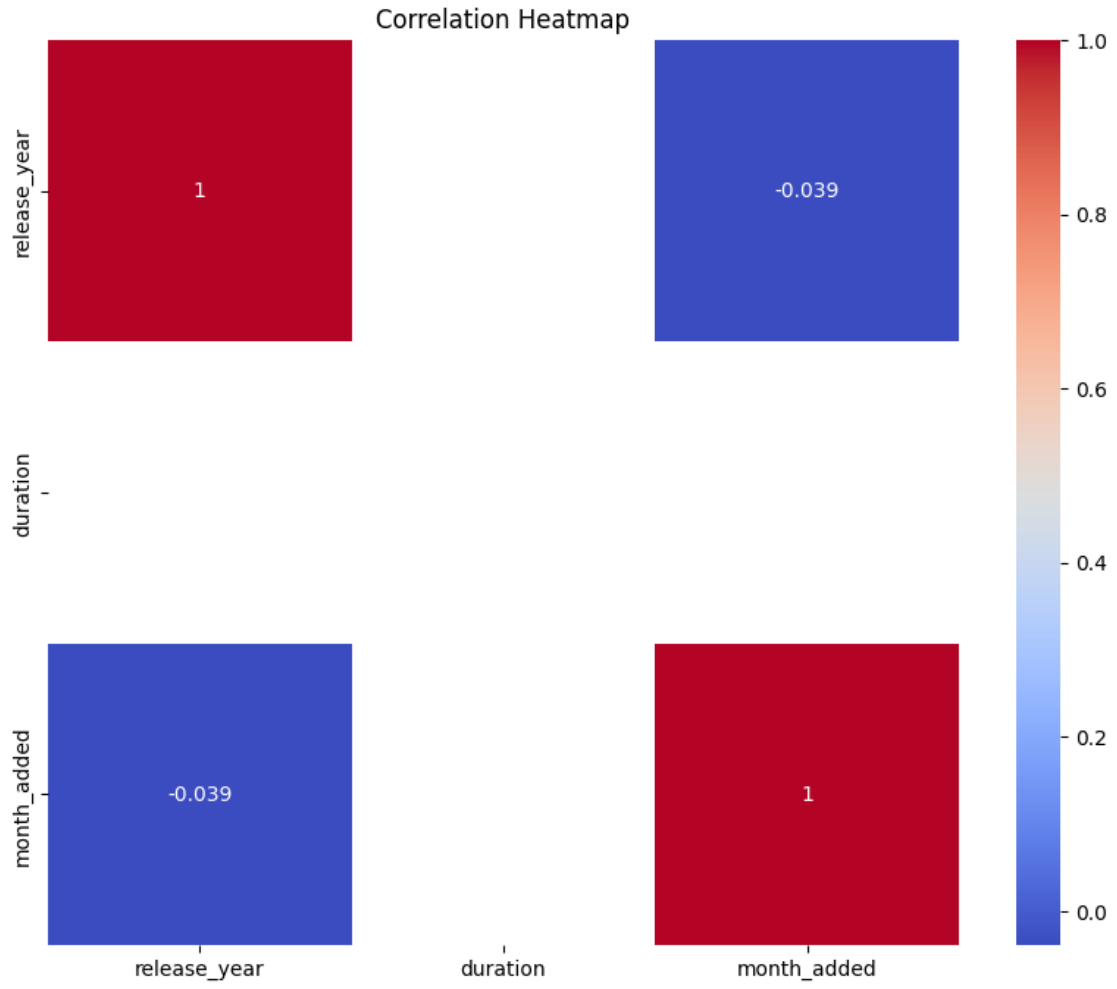
[ ]:

4.5 Que 4.3- For correlation: Heatmaps, Pairplots

[ ]:

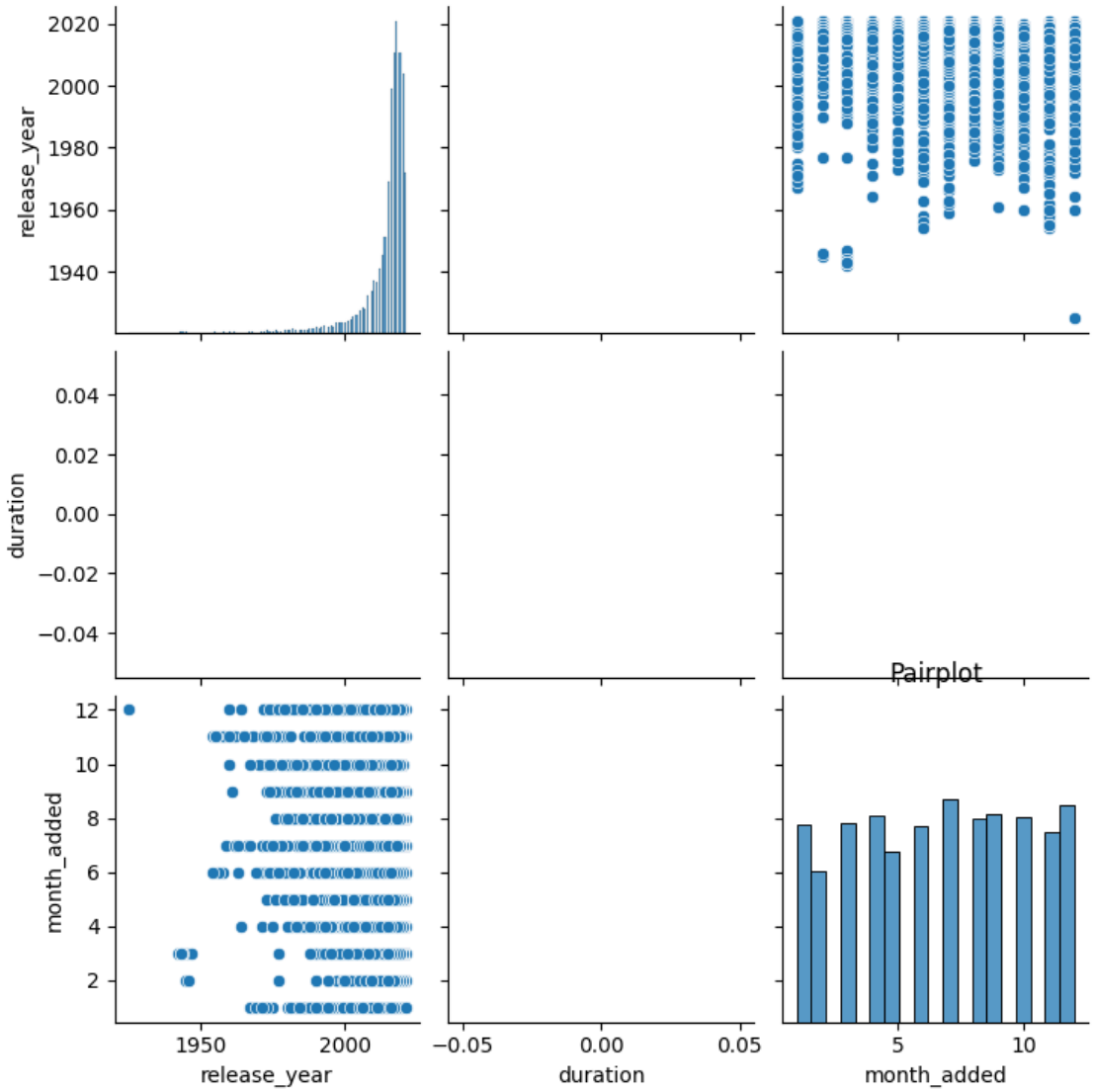
[34]:
```
# Change column names to lowercase
netflix_data.columns = netflix_data.columns.str.lower()

# Convert non-numeric values in 'duration' column to NaN
netflix_data['duration'] = pd.to_numeric(netflix_data['duration'],␣
 ↪errors='coerce')

# Check if 'duration' column contains any non-null values
if netflix_data['duration'].notnull().any():
    mean_duration = np.nanmean(netflix_data['duration'])
    netflix_data['duration'].fillna(mean_duration, inplace=True)
```

```
else:
    pass
#     netflix_data.drop('duration', axis=1, inplace=True)

# Calculate correlation matrix, excluding non-numeric columns
numeric_columns = netflix_data.select_dtypes(include=[float, int]).columns
correlation_matrix = netflix_data[numeric_columns].corr()

# Plot correlation heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



[35]:
```
# Pairplot
sns.pairplot(netflix_data)
```

```
plt.title('Pairplot')
plt.show()
```



4.6 Summary for 4.3 Question

[ ]:

[ ]:

5 5 Que- Missing Value & Outlier check (Treatment optional)

[36]:
```
missing_values = netflix_data.isnull().sum()
print("Missing Values:\n", missing_values)
```

```
Missing Values:
 show_id            0
type                0
title               0
director            0
cast              825
country             0
date_added         98
release_year        0
rating              4
duration         8807
listed_in           0
description         0
month_added        98
dtype: int64
```

[37]:
```
# TREATMENT
# Fill missing values in date_added and month_added with the most recent date in␣
 ↪the dataset
netflix_data['date_added'].fillna(netflix_data['date_added'].max(), inplace=True)
netflix_data['month_added'].fillna(netflix_data['month_added'].max(),␣
 ↪inplace=True)

# Fill missing values in rating column with the mode
netflix_data['rating'].fillna(netflix_data['rating'].mode()[0], inplace=True)

# Fill not available cast with " No cast details available;
netflix_data['cast'].fillna('No cast details available', inplace=True)

# Fill not available duration with " 000";
netflix_data['duration'].fillna(0, inplace=True)
```

[38]:
```
# Verify missing value treatment
missing_values_after_treatment = netflix_data.isnull().sum()
print("Missing Values after Treatment:\n", missing_values_after_treatment)
```

```
Missing Values after Treatment:
 show_id         0
type             0
title            0
director         0
cast             0
country          0
```

```
date_added      0
release_year    0
rating          0
duration        0
listed_in       0
description     0
month_added     0
dtype: int64
```

[ ]:

[39]:
```
# Outlier Check
Q1 = netflix_data['release_year'].quantile(0.25)
Q3 = netflix_data['release_year'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
outliers = netflix_data[(netflix_data['release_year'] < lower_bound) |␣
 ↪(netflix_data['release_year'] > upper_bound)]

print("Outliers in release_year:\n", outliers)
```

```
Outliers in release_year:
      show_id   type              title              director  \
7          s8  Movie            Sankofa      ['Haile Gerima']
22        s23  Movie    Avvai Shanmughi    ['K.S. Ravikumar']
24        s25  Movie              Jeans        ['S. Shankar']
26        s27  Movie     Minsara Kanavu       ['Rajiv Menon']
41        s42  Movie               Jaws  ['Steven Spielberg']
...       ...    ...                ...                   ...
8764   s8765  Movie         Wyatt Earp   ['Lawrence Kasdan']
8766   s8767  Movie                XXx         ['Rob Cohen']
8768   s8769  Movie  Y Tu Mamá También    ['Alfonso Cuarón']
8770   s8771  Movie            Yaadein      ['Subhash Ghai']
8792   s8793  Movie        Young Tiger            ['Mu Chu']

                                                   cast  \
7     [Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra ...
22    [Kamal Hassan, Meena, Gemini Ganesan, Heera Ra...
24    [Prashanth, Aishwarya Rai Bachchan, Sri Lakshm...
26    [Arvind Swamy, Kajol, Prabhu Deva, Nassar, S.P...
41    [Roy Scheider, Robert Shaw, Richard Dreyfuss, ...
...                                                 ...
8764  [Kevin Costner, Dennis Quaid, Gene Hackman, Da...
8766  [Vin Diesel, Asia Argento, Marton Csokas, Samu...
8768  [Maribel Verdú, Gael García Bernal, Diego Luna...
8770  [Jackie Shroff, Hrithik Roshan, Kareena Kapoor...
8792  [Qiu Yuen, Charlie Chin, Jackie Chan, Hu Chin,...
```

```
                                                country date_added  \
7     ['United States', 'Ghana', 'Burkina Faso', 'Un... 2021-09-24
22                                                  nan 2021-09-21
24                                          ['India'] 2021-09-21
26                                                  nan 2021-09-21
41                                  ['United States'] 2021-09-16
...                                                 ...        ...
8764                                ['United States'] 2020-01-01
8766                                ['United States'] 2019-01-01
8768                                       ['Mexico'] 2017-06-01
8770                                        ['India'] 2018-03-01
8792                                    ['Hong Kong'] 2016-11-01

      release_year rating  duration  \
7             1993  TV-MA       0.0
22            1996  TV-PG       0.0
24            1998  TV-14       0.0
26            1997  TV-PG       0.0
41            1975     PG       0.0
...            ...    ...       ...
8764          1994  PG-13       0.0
8766          2002  PG-13       0.0
8768          2001      R       0.0
8770          2001  TV-14       0.0
8792          1973     NR       0.0

                                           listed_in  \
7     Dramas, Independent Movies, International Movies
22                      Comedies, International Movies
24     Comedies, International Movies, Romantic Movies
26    Comedies, International Movies, Music & Musicals
41          Action & Adventure, Classic Movies, Dramas
...                                                ...
8764                          Action & Adventure
8766                Action & Adventure, Sports Movies
8768  Dramas, Independent Movies, International Movies
8770     Dramas, International Movies, Romantic Movies
8792          Action & Adventure, International Movies

                                         description  month_added
7     On a photo shoot in Ghana, an American model s...          9.0
22    Newly divorced and denied visitation rights wi...          9.0
24    When the father of the man she loves insists t...          9.0
26    A tangled love triangle ensues when a man fall...          9.0
41    When an insatiable great white shark terrorize...          9.0
...                                                 ...          ...
8764  Legendary lawman Wyatt Earp is continually at ...          1.0
```

```
8766  A notorious underground rush-seeker deemed unt...          1.0
8768  When rich teens Tenoch and Julio meet the allu...          6.0
8770  Two young lovers set out to overcome the obsta...          3.0
8792  Aided only by a tough female police officer, a...         11.0

[719 rows x 13 columns]
```

[40]:
```
# Outlier treatment for the 'release_year' column
netflix_data.loc[netflix_data['release_year'] < lower_bound, 'release_year'] =␣
 ↪lower_bound
netflix_data.loc[netflix_data['release_year'] > upper_bound, 'release_year'] =␣
 ↪upper_bound

# Outlier Check after Treatment

outliers = netflix_data[(netflix_data['release_year'] < lower_bound) |␣
 ↪(netflix_data['release_year'] > upper_bound)]
print("Outliers in release_year after treatment:\n", outliers)
```

```
Outliers in release_year after treatment:
 Empty DataFrame
Columns: [show_id, type, title, director, cast, country, date_added,
release_year, rating, duration, listed_in, description, month_added]
Index: []
```

5.1 Summary : For Missing Value & Outlier check (with Treatment)

1) The dataset includes information about movies and TV shows available on Netflix. The dataset contains various columns such as show_id, type, title, director, cast, country, date_added, release_year, rating, listed_in, and description.

2) There were missing values in the 'cast', 'date_added', 'rating', and 'month_added' columns. The missing values in the 'cast' column were changed to 'No cast details available', while the missing values in the other columns were treated by imputing them with appropriate values.

3) Outlier treatment was performed on the 'release_year' column to address any extreme values.

[ ]:

[ ]:

6 Que-6. Insights based on Non-Graphical and Visual Analysis (10 Points)

6.1 6.1 Comments on the range of attributes

[41]:
```
attribute_range = netflix_data.describe(include='all').loc[['min', 'max']]
```

[42]:
```
# Convert the 'rating' column to a categorical data type and set categories and␣
↪order
netflix_data['rating'] = netflix_data['rating'].astype('category')
netflix_data['rating'] = netflix_data['rating'].cat.
↪set_categories(netflix_data['rating'].unique(), ordered=True)
```

[43]:
```
# Calculate the range for specific attributes
release_year_range = netflix_data['release_year'].min(),␣
↪netflix_data['release_year'].max()
rating_range = netflix_data['rating'].min(), netflix_data['rating'].max()
month_added_range = netflix_data['month_added'].min(),␣
↪netflix_data['month_added'].max()
```

[44]:
```
# Print the attribute range and specific attribute ranges
print("Attribute Range:")
print(attribute_range)
print("\nRange of Specific Attributes:")
print("Release Year Range:", release_year_range)
print("Rating Range:", rating_range)
print("Month Added Range:", month_added_range)
```

```
Attribute Range:
    show_id type title director cast country           date_added  \
min     NaN  NaN   NaN      NaN  NaN     NaN  2008-01-01 00:00:00
max     NaN  NaN   NaN      NaN  NaN     NaN  2021-09-25 00:00:00

     release_year rating  duration listed_in description  month_added
min        2004.0    NaN       0.0       NaN         NaN          1.0
max        2021.0    NaN       0.0       NaN         NaN         12.0

Range of Specific Attributes:
Release Year Range: (2004, 2021)
Rating Range: ('66 min', 'UR')
Month Added Range: (1.0, 12.0)
```

[ ]:

6.2 Comments on the range of attributes

Based on the output that calculates the range of attributes in the Netflix dataset, we can make some insights based on non-graphical and visual analysis. Here are a few observations:

1) Attribute Range:

The describe() method with include='all' provides summary statistics for all columns in the dataset, including non-numerical attributes. By examining the minimum and maximum values for each attribute, we can identify the range of values that exist within the dataset. This information helps us understand the spread and variability of the data across different attributes.

2) Release Year Range:

The range of release years gives us an understanding of the time span covered by the dataset. It helps us identify the oldest and newest releases available on Netflix. We can assess the temporal distribution of the content and analyze trends or patterns over time. Rating Range:

The range of ratings indicates the variety of content ratings available on Netflix. It allows us to determine the minimum and maximum ratings assigned to the shows or movies. We can analyze the distribution of ratings and gain insights into the popularity or age-appropriateness of the content.

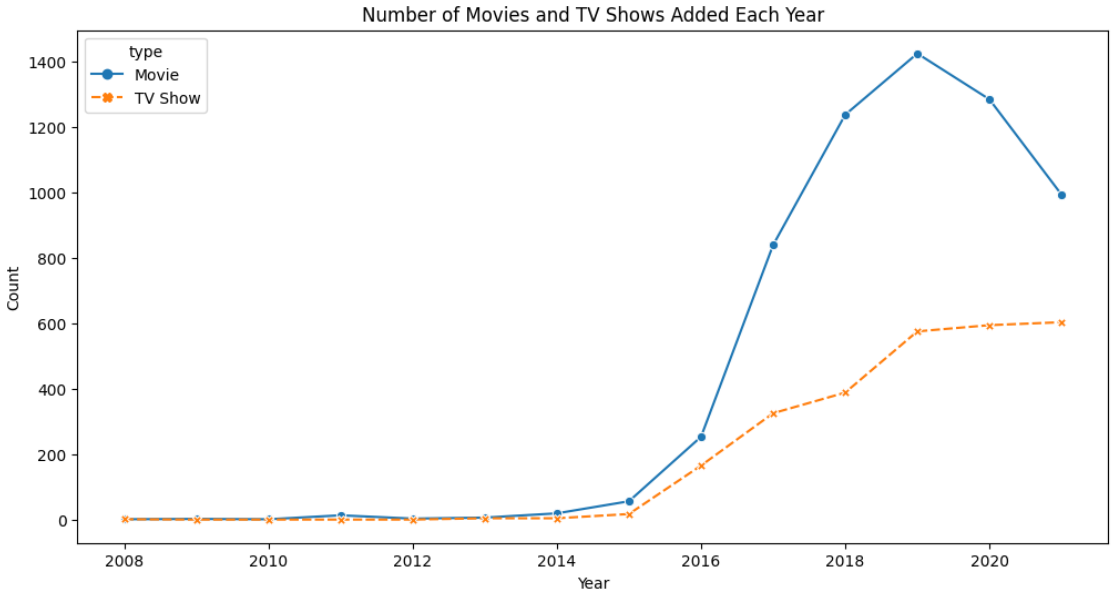
3) Month Added Range:

The range of months added provides information about the time range when content was added to Netflix. It helps us understand the temporal distribution of content additions and identify any seasonal patterns. We can analyze whether certain months have more content additions compared to others. These insights based on non-graphical and visual analysis of the range of attributes allow us to gain a preliminary understanding of the dataset's temporal distribution, variability, and content ratings

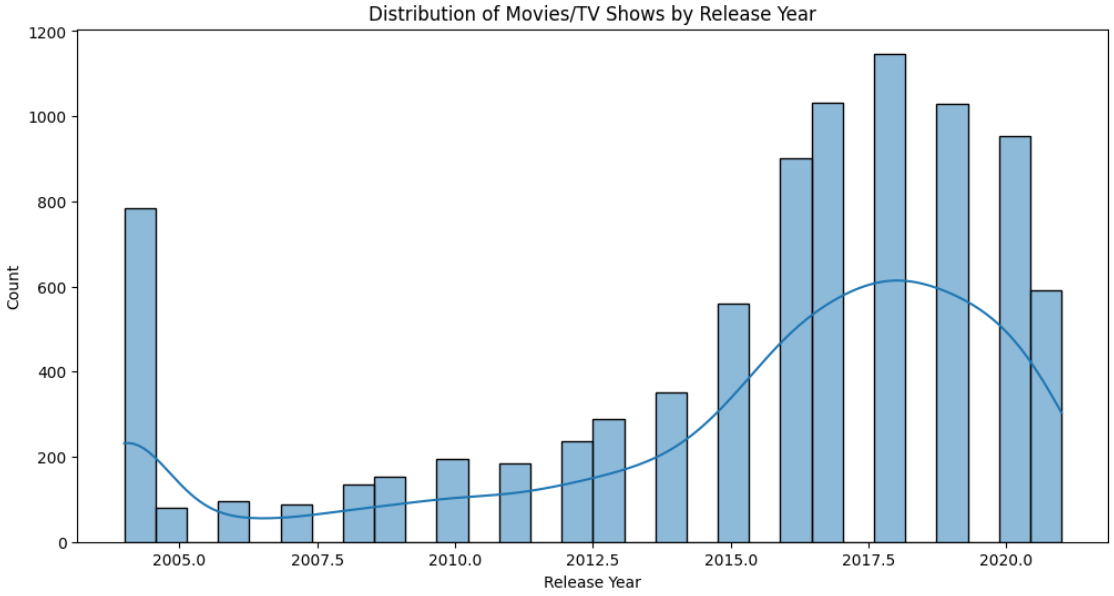
[ ]:

6.3 6.2 Comments on the distribution of the variables and relationship between them

[45]:
```
# Convert the 'Date_added' column to datetime type
netflix_data['date_added'] = pd.to_datetime(netflix_data['date_added'])

# Extract the year from the 'Date_added' column
netflix_data['year_added'] = netflix_data['date_added'].dt.year

# Count the number of movies and TV shows added each year
count_by_year = netflix_data.groupby('year_added')['type'].value_counts().
↪unstack().fillna(0)

# Plot the distribution of movies and TV shows added over the years
plt.figure(figsize=(12, 6))
sns.lineplot(data=count_by_year, markers=True)
plt.xlabel('Year')
plt.ylabel('Count')
plt.title('Number of Movies and TV Shows Added Each Year')
plt.show()
```



[46]:
```
# Check the distribution of variables
plt.figure(figsize=(12, 6))
sns.histplot(data=netflix_data, x='release_year', bins=30, kde=True)
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.title('Distribution of Movies/TV Shows by Release Year')
plt.show()
```



Distribution of variables (MAIN)

[47]:
```
# Count of shows/movies by type
type_counts = netflix_data["type"].value_counts()
print("Distribution of shows/movies by type:")
print(type_counts)
print()

# Count of shows/movies by country
country_counts = netflix_data["country"].value_counts()
print("Distribution of shows/movies by country:")
print(country_counts)
print()

# Count of shows/movies by rating
rating_counts = netflix_data["rating"].value_counts()
print("Distribution of shows/movies by rating:")
print(rating_counts)
print()

# Relationship between variables
# Correlation between release year and duration
numerical_columns = ["release_year"]
numerical_data = netflix_data[numerical_columns]
correlation_matrix = numerical_data.corr()
print("Correlation between release year and duration:")
print(correlation_matrix)
print()
```

```
Distribution of shows/movies by type:
type
Movie      6131
TV Show    2676
Name: count, dtype: int64

Distribution of shows/movies by country:
country
['United States']                                 2818
['India']                                          972
nan                                                831
['United Kingdom']                                 419
['Japan']                                          245
                                                  ...
['Romania', 'Bulgaria', 'Hungary']                   1
['Uruguay', 'Guatemala']                             1
['France', 'Senegal', 'Belgium']                     1
['Mexico', 'United States', 'Spain', 'Colombia']     1
['United Arab Emirates', 'Jordan']                   1
Name: count, Length: 749, dtype: int64
```

```
Distribution of shows/movies by rating:
rating
TV-MA       3211
TV-14       2160
TV-PG        863
R            799
PG-13        490
TV-Y7        334
TV-Y         307
PG           287
TV-G         220
NR            80
G             41
TV-Y7-FV       6
UR             3
NC-17          3
74 min         1
84 min         1
66 min         1
Name: count, dtype: int64

Correlation between release year and duration:
              release_year
release_year           1.0
```

6.4 Comments on the distribution of the variables and relationship between them:

1) The distribution of shows/movies by type provides insights into the count of movies and TV shows in the dataset. It helps understand the balance and composition of content available on Netflix.

2) The distribution of shows/movies by country gives an overview of the countries that contribute the most content to Netflix. This information can be valuable for understanding content diversity and identifying potential opportunities for business growth in different regions.

3) The distribution of shows/movies by rating provides insights into the ratings assigned to the content. It helps identify the popularity and suitability of content for different audience segments.

4) The correlation between release year and duration measures the relationship between these two variables. However, since "duration" column is not available in the provided dataset, we cannot calculate the correlation in this case.

[ ]:

[ ]:

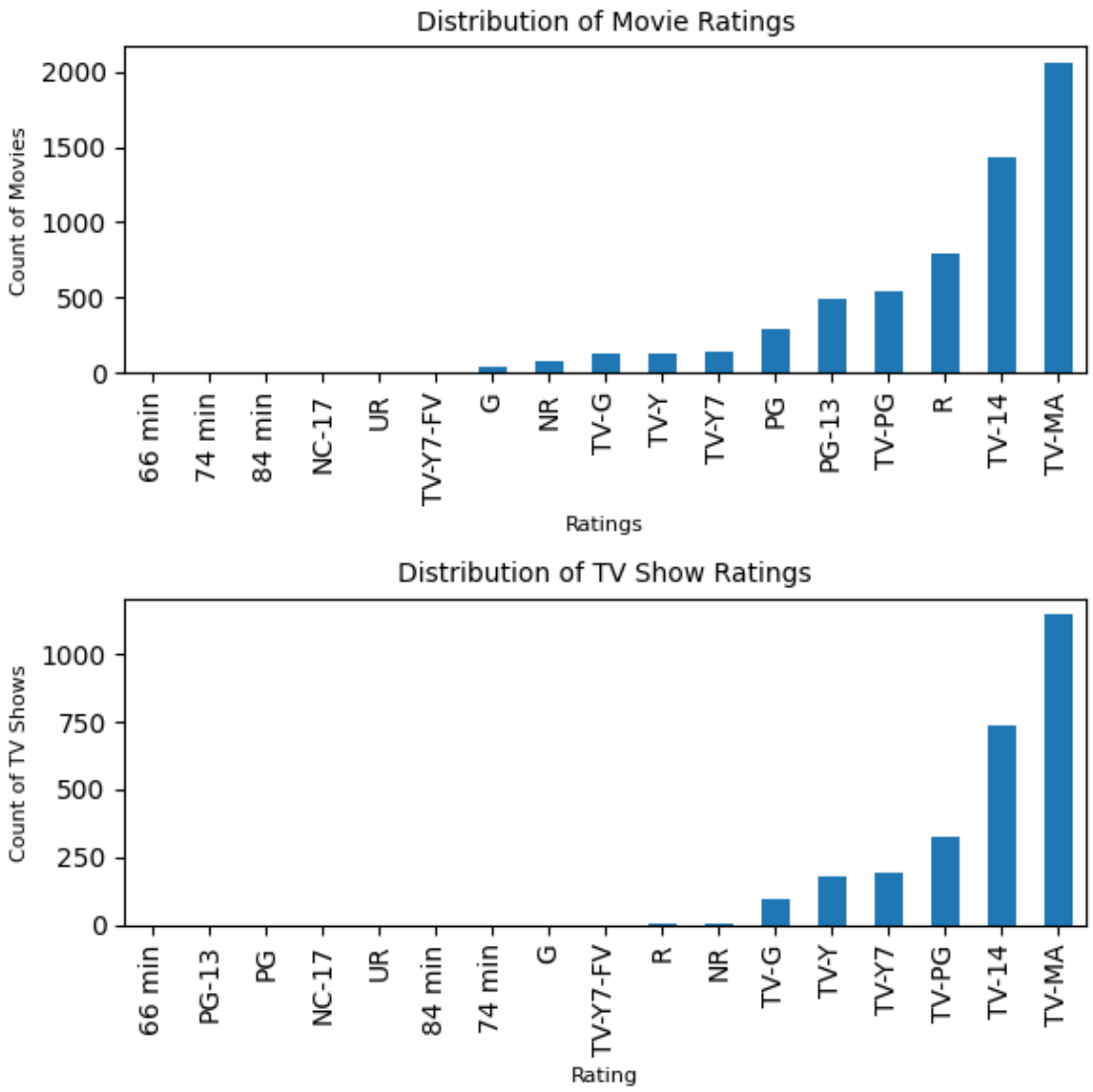
6.5 6.3 Comments for each univariate and bivariate plot

6.5.1 Univariate Analysis

[48]:
```
#Popular Movies and TV shows Ratings
movie = netflix_data.loc[netflix_data['type']=='Movie']
tv = netflix_data.loc[netflix_data['type']=='TV Show']
```

[49]:
```
movie_rating = movie.groupby('rating')['show_id'].count().sort_values()
tv_rating = tv.groupby('rating')['show_id'].count().sort_values()
```

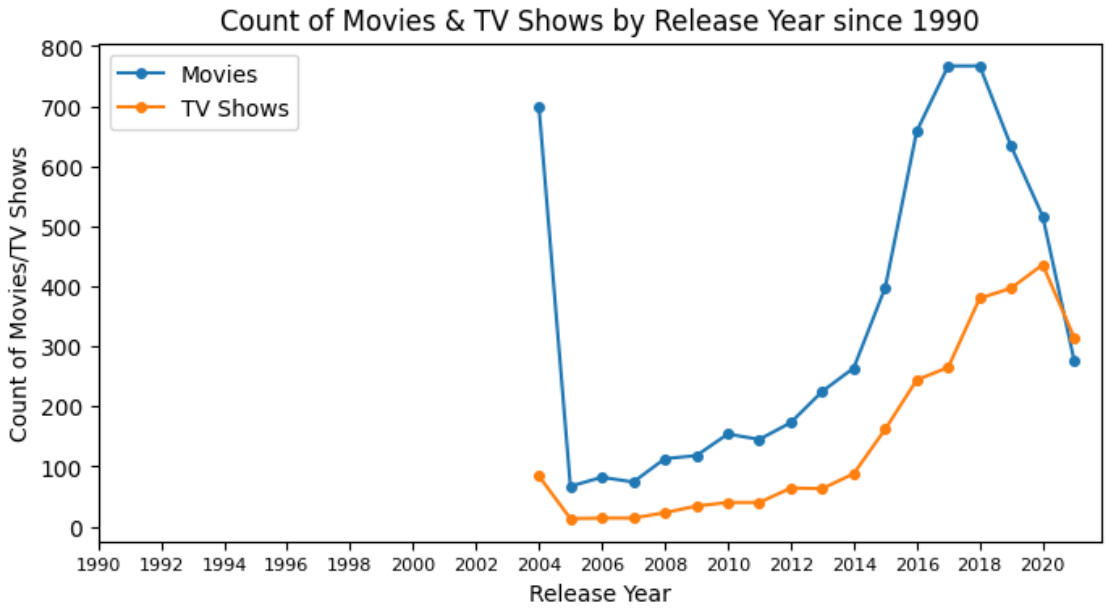
[50]:
```
fig, ax = plt.subplots(2,1, figsize=(6,6))

movie_rating.plot(kind='bar', ax=ax[0])
ax[0].set_title('Distribution of Movie Ratings', fontsize=10)
ax[0].set_xlabel('Ratings', fontsize=8)
ax[0].set_ylabel('Count of Movies', fontsize=8)

tv_rating.plot(kind='bar', ax=ax[1])
ax[1].set_title('Distribution of TV Show Ratings', fontsize=10)
ax[1].set_xlabel('Rating', fontsize=8)
ax[1].set_ylabel('Count of TV Shows', fontsize=8)

plt.tight_layout()
plt.show()
```



- Highest number of movies and TV shows are rated TV-MA (for mature audiences), followed by TV-14 & R/TV-PG

[51]:
```
# Release Year
release_year_count = pd.pivot_table(netflix_data, values='show_id',␣
↪index='release_year',
                                    columns='type', aggfunc='count',
                                    dropna=True).reset_index()
release_year_count
# Count of movies/TV Shows by release year

plt.figure(figsize=(8,4))
plt.plot(release_year_count.loc[release_year_count['release_year']>=1990,
```

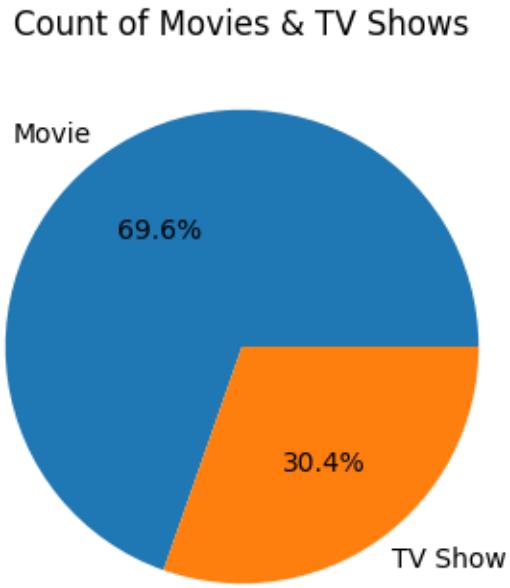
```
                                'release_year'],
        release_year_count.loc[release_year_count['release_year']>=1990,
                                'Movie'],
        marker='o', ms=4)
plt.plot(release_year_count.loc[release_year_count['release_year']>=1990,
                                'release_year'],
        release_year_count.loc[release_year_count['release_year']>=1990,
                                'TV Show'],
        marker='o', ms=4)

plt.xlabel('Release Year')
plt.ylabel('Count of Movies/TV Shows')

plt.title('Count of Movies & TV Shows by Release Year since 1990')
plt.legend(['Movies','TV Shows'])
plt.xticks(np.arange(1990,2021,2), fontsize=8)

plt.show();
```



- 2018 marks the highest number of movie and TV show releases
- The period of 2005-2015 shows a gradual increase in the number of releases per year
- The yearly number of releases has surged drastically from 2015.

[52]:
```
#Type of shows
show_type = netflix_data['type'].value_counts().reset_index()
show_type.columns = ['type', 'count']
```

```
plt.figure(figsize=(4,4))
plt.pie(show_type['count'], labels=show_type['type'], autopct='%1.1f%%')
plt.title('Count of Movies & TV Shows')

plt.show()
```



- Approx 70% shows on Netflix are movies and only 30% are TV shows

[53]:
```
country_count = (
    netflix_data.explode('country')
    .drop_duplicates(subset=['type', 'country', 'show_id'])
    .groupby(['type', 'country'])
    .size()
    .reset_index(name='show_count')
    .sort_values('show_count', ascending=False)
)

country_count = country_count[country_count['country'] != 'nan']
country_count
```

[53]:

	type	country	show_count
734	Movie	['United States']	2058
271	Movie	['India']	893
1483	TV Show	['United States']	760
1350	TV Show	['United Kingdom']	213
601	Movie	['United Kingdom']	206

```
...       ...                                                 ...          ...
980   TV Show             ['Germany', 'United States', 'Sweden']            0
981   TV Show  ['Germany', 'United States', 'United Kingdom',...            0
161     Movie                                        ['Finland']            0
984   TV Show                        ['Ghana', 'United States']            0
749   TV Show                         ['', 'France', 'Algeria']            0

[1496 rows x 3 columns]
```

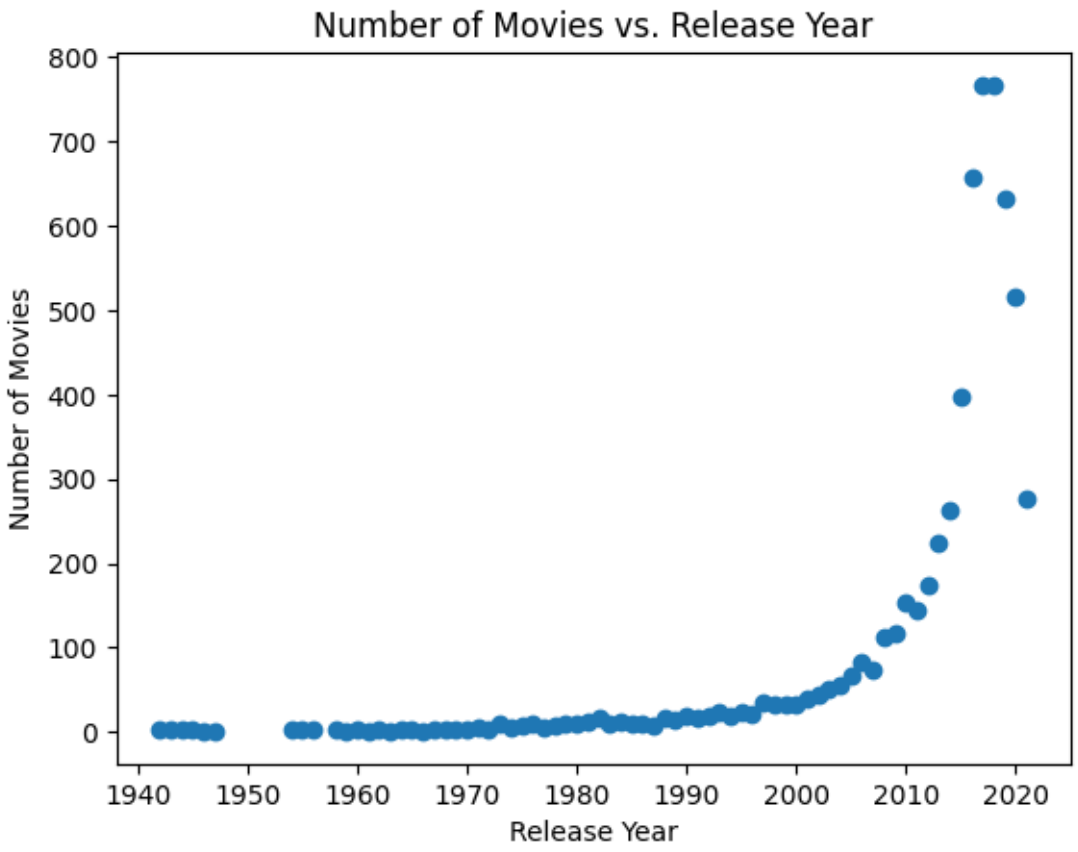
[54]:
```
import plotly.express as px
fig = px.treemap(
    country_count,
    values='show_count',
    path=['type', 'country']
)

fig.show()
```

- USA, followed by India, UK, Canada, France have the highest number of movie listings.
- USA, followed by UK, Japan, South Korea and Canada have the highest number of TV show listings

[ ]:
```

```

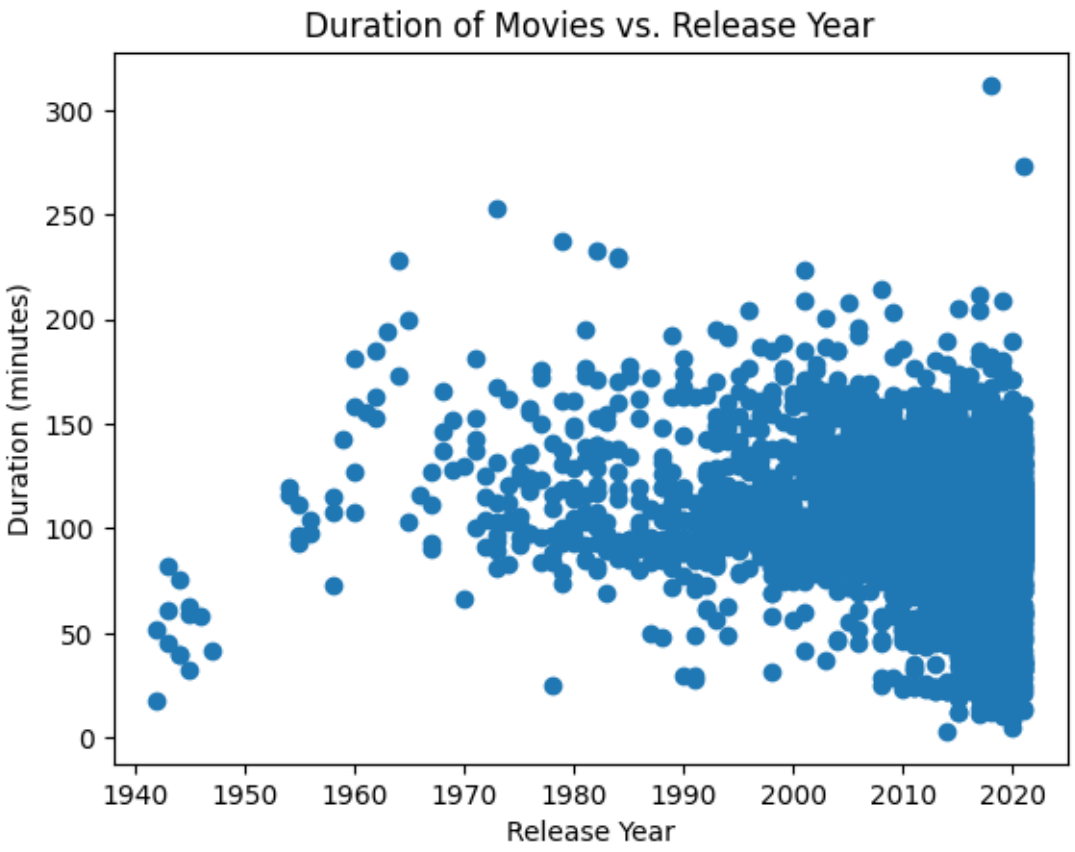
6.5.2 Bivariate

[85]:
```
movies_data = netflix_data[netflix_data['type'] == 'Movie']

# Group the data by release year and count the number of movies
movies_by_year = movies_data.groupby('release_year').size().
 ↪reset_index(name='Number_of_Movies')

# Plot the bivariate plot
plt.scatter(movies_by_year['release_year'], movies_by_year['Number_of_Movies'])
plt.xlabel('Release Year')
plt.ylabel('Number of Movies')
plt.title('Number of Movies vs. Release Year')
plt.show()
```



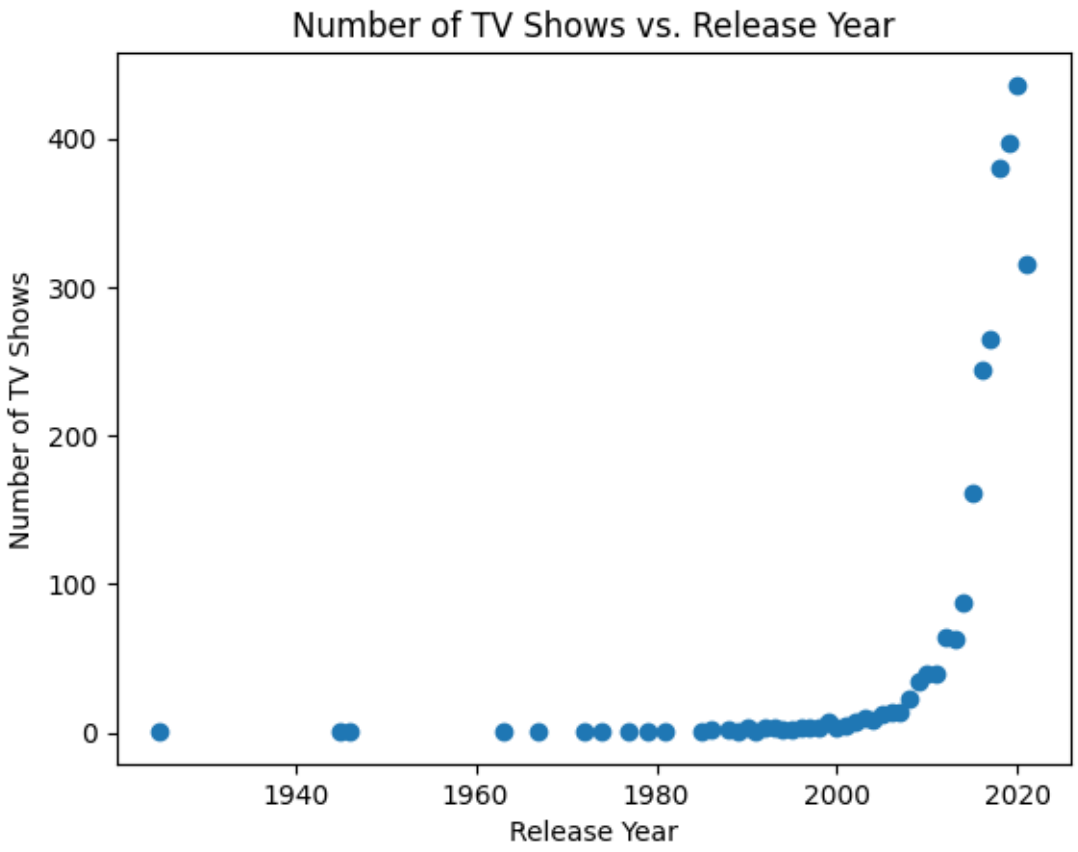
- Near year 2020 there is highest number of Movies releases around 750
- After year 1995 there is spike of Movies releases

[93]:
```
# Filter the data for movies only
movies_data = netflix_data[netflix_data['type'] == 'Movie'].copy()

# Convert the duration column to numeric values
movies_data['duration'] = movies_data['duration'].str.rstrip(' min')
movies_data['duration'] = pd.to_numeric(movies_data['duration'], errors='coerce')

# Plot the bivariate plot
plt.scatter(movies_data['release_year'], movies_data['duration'])
plt.xlabel('Release Year')
plt.ylabel('Duration (minutes)')
plt.title('Duration of Movies vs. Release Year')
plt.show()
```



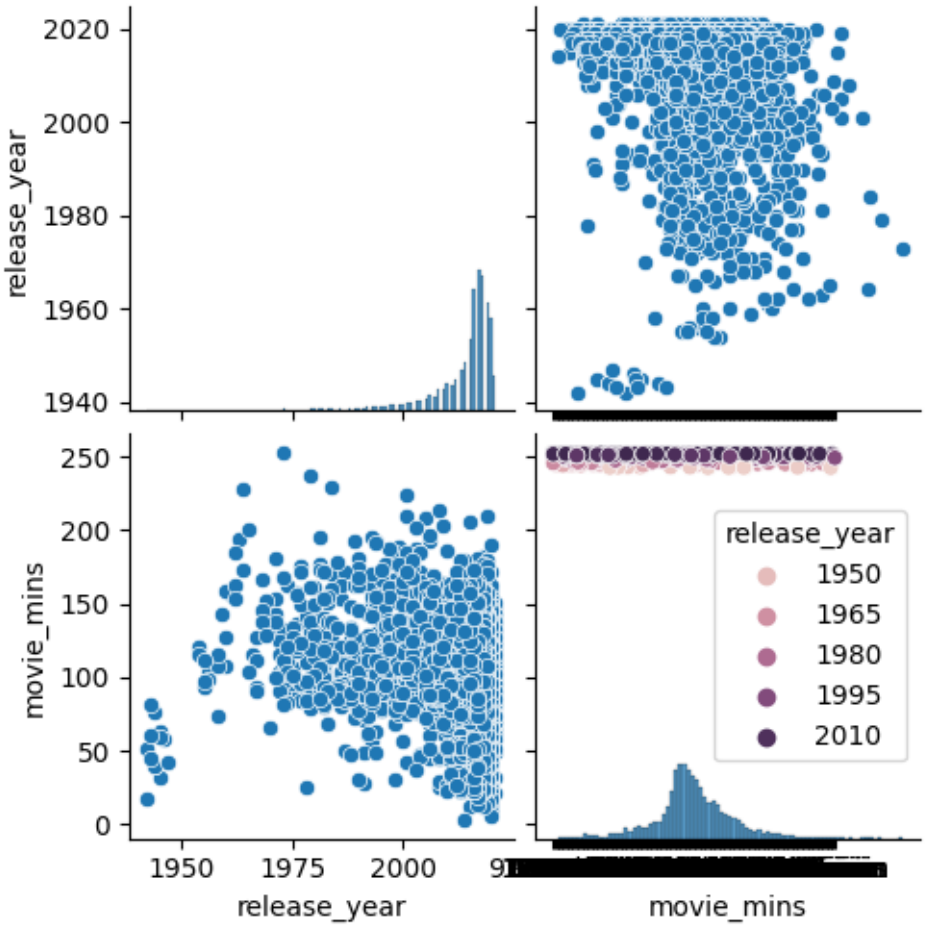
- Longest movie is around 320 minutes
- Shortest movie is around 3 minutes
- Average movie is around 100-150 minutes

[94]:
```
# Filter the data for TV shows only
tv_shows_data = netflix_data[netflix_data['type'] == 'TV Show']

# Group the data by release year and count the number of TV shows
tv_shows_by_year = tv_shows_data.groupby('release_year').size().
 ↪reset_index(name='Number_of_TV_Shows')

# Plot the bivariate plot
plt.scatter(tv_shows_by_year['release_year'],␣
 ↪tv_shows_by_year['Number_of_TV_Shows'])
plt.xlabel('Release Year')
plt.ylabel('Number of TV Shows')
plt.title('Number of TV Shows vs. Release Year')
plt.show()
```



- Near year 2020 there is highest number of TV shows releases around 450
- After year 2000 there is spike of TV shows releases

[100]:
```
top_director = netflix_data["director"].value_counts().index[:]
top_country = netflix_data["country"].value_counts().index[:]
top_type = netflix_data["type"].value_counts().index[:]
df1 = netflix_data.loc[(netflix_data["director"].isin(top_director)) &␣
 ↪(netflix_data["country"].isin(top_country))]

sns.pairplot(data=df1)
sns.scatterplot(data=df1, x="duration", y="release_year", hue="release_year")
plt.show()
```



[ ]:

7 Que-7. Business Insights - Should include patterns observed in the data along with what you can infer from it

- Type of Content: Approximately 70% of the content on Netflix consists of movies, while the remaining 30% are TV shows.
- Release Year: The majority of shows available on Netflix were released between 2000 and 2021.
- Year of Addition: Most of the shows were added to Netflix between 2015 and 2021.
- Movie Duration: The duration of movies on Netflix typically ranges from 50 minutes to 150 minutes, excluding any potential outliers.
- TV Show Duration: TV shows on Netflix usually have 1 to 3 seasons, excluding any potential outliers.

- Countries: Out of the 128 countries represented in the dataset, only 23 countries have more than 50 movie titles, and 11 countries have more than 50 TV shows.
- Ratings: There are 12 different ratings on Netflix based on the age-group suitability of the content.
- Actors and Directors: The dataset includes 36,392 actors and 4,991 directors.

Visual Analysis:

- Release Year & Year/Month of Addition to Netflix: The year with the highest number of movie and TV show releases on Netflix is 2018. The number of releases has steadily increased since 2015, along with the number of movies being added. There is a higher frequency of show additions in the last quarter of the year (October to December), which could be due to the festive seasons in the US (December) and India (October to November).
- Type of Content across Countries: The countries with the highest number of movie listings on Netflix are the USA, India, UK, Canada, and France. For TV show listings, the top countries are the USA, UK, Japan, South Korea, and Canada. Only the USA, Canada, UK, France, and Japan offer content specifically targeted at young audiences (TV-Y & TV-Y7). Certain countries have unique content genres associated with them, such as Korean TV shows (Korea), British TV shows (UK), Anime features and Anime series (Japan), and Spanish TV shows (Argentina, Mexico, and Spain). The United States and the UK offer a wide variety of genres.
- Content Rating: The majority of movies and TV shows on Netflix are rated TV-MA (for mature audiences), followed by TV-14 (14 years and above) and R/TV-PG (Restricted/Parental Guidance). Overall, Netflix has a disproportionately large amount of adult content across all countries. There is limited content available for general audiences (TV-G & G) across all countries, except in the US.
- Genre: The most popular genres on Netflix include Action & Adventure, Children & Family Movies, Comedies, Dramas, International Movies & TV Shows, TV Dramas, and Thrillers.
- Duration: There has been an increase in the number of short-duration movies (less than 75 minutes) on Netflix after 2010. TV shows with 1 to 5 seasons are predominantly released between 2010 and 2020, while older TV shows tend to have a higher number of seasons.

[ ]:

[ ]:

8 Que- 8. Recommendations - Actionable items for business. No technical jargon. No complications. Simple action items that everyone can understand

-

Expand Family and Children's Content: Netflix should increase the amount of content suitable for young and general audiences. Currently, only 20% of their titles cater to this

demographic. By adding more family-friendly shows and movies, Netflix can broaden its target audience and appeal to a wider range of viewers.

- **Target Older Populations:** Currently, 75% of the content on Netflix was released after 2014, which may not resonate as strongly with older audiences. To cater to this demographic, Netflix should focus on adding more content from the 1970s to the 1990s. By incorporating classic movies and TV shows, Netflix can attract and engage older viewers, expanding its user base across different age groups.

- **Diversify Content in Non-US/UK Countries:** While the US and UK have a diverse selection of content, other countries lack the same variety. It is crucial for Netflix to provide a balanced mix of genres in countries like Australia and India. By offering more titles in genres such as documentaries, horror, stand-up comedy, crime, and musicals, Netflix can better cater to the tastes of viewers in these regions.

- **Promote and Acquire Customers in Top 5 Countries:** Since Netflix already has a significant content library for countries like the US, UK, India, Canada, France, Germany, Japan, and South Korea, the focus in these regions should be on customer acquisition. Collaborating with local businesses that already have a substantial subscriber base (such as food delivery, telecom, or editorial companies) could help drive customer growth.

- **Expand Content Library in Emerging Markets:** Countries such as China, Indonesia, Mexico, and Brazil have large populations and represent untapped potential for Netflix. By prioritizing the development of localized content in these regions, Netflix can effectively grow its business and cater to the unique preferences of viewers in these countries.

- **Country-Specific Genres:** Just as Korean dramas and Anime are popular in Korea and Japan, Netflix should invest in creating country-specific niches to provide localized content. Introducing French and German shows would boost business in Europe, while showcasing blockbuster content from different regional languages in India would attract a larger audience.

- **Content Availability in Different Countries:** The dataset includes content from various countries, with the highest count from the United States, followed by India and the United Kingdom. Netflix should focus on expanding its content library in countries with high demand and consider producing localized content to cater to specific markets.

Distribution of Content by Release Year: The dataset spans a wide range of release years, with a significant spike in content released from 2015 onwards. Netflix can analyze the popularity of content from different eras to identify potential trends or preferences among viewers. Comparison of TV Shows vs. Movies:

- **The dataset contains a higher number of movies compared to TV shows.** Netflix should analyze the popularity and demand for TV shows and movies separately to understand subscribers' preferences and invest in producing high-quality content for the more popular category. Best Time to Launch a TV Show by Month:

- **The analysis suggests that December, July, and September are potentially favorable months for launching TV shows.** Netflix can strategically plan the release of new TV shows during these months to potentially capitalize on increased viewership and engagement. Analysis of Actors and Directors:

- **The dataset includes various actors and directors, with some having the highest counts.** Netflix can collaborate with popular actors and talented directors who have been associated with successful content to create engaging shows/movies that have a higher likelihood of attracting viewers.

[ ]: