Business Case: Netflix_Project

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

In [ ]:
```
!gdown https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/origi
```

```
Downloading...
From: https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/origi
nal/netflix.csv
To: /content/netflix.csv
100% 3.40M/3.40M [00:00<00:00, 129MB/s]
```

In [ ]:
```
df = pd.read_csv('/content/netflix.csv')
df.head(10)
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA
5	s6	TV Show	Midnight Mass	Mike Flanagan	Kate Siegel, Zach Gilford, Hamish Linklater, H...	NaN	September 24, 2021	2021	TV-MA
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	Vanessa Hudgens, Kimiko Glenn, James Marsden, ...	NaN	September 24, 2021	2021	PG
7	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	September 24, 2021	1993	TV-MA
8	s9	TV Show	The Great British Baking Show	Andy Devonshire	Mel Giedroyc, Sue Perkins, Mary Berry, Paul Ho...	United Kingdom	September 24, 2021	2021	TV-14
9	s10	Movie	The Starling	Theodore Melfi	Melissa McCarthy,	United States	September 24, 2021	2021	PG-13

show_id	type	title	director	cast	country	date_added	release_year	rating
				Chris O'Dowd, Kevin Kline, T...				

In [ ]:

```
df.shape
```

Out[ ]:

```
(8807, 12)
```

In [ ]:

```
df.columns
```

Out[ ]:

```
Index(['show_id', 'type', 'title', 'director', 'cast', 'country', 'date_added',
       'release_year', 'rating', 'duration', 'listed_in', 'description'],
      dtype='object')
```

In [ ]:

```
df.dtypes
```

Out[ ]:

```
show_id         object
type            object
title           object
director        object
cast            object
country         object
date_added      object
release_year     int64
rating          object
duration        object
listed_in       object
description     object
dtype: object
```

In [ ]:

```
# Converting categorical attributes to 'category' data type if necessary.
for column in df.columns:
    if df[column].dtype == 'object':
        unique_values_count = len(df[column].unique())
        total_values_count = len(df[column])
        if unique_values_count / total_values_count < 0.5:
            df[column] = df[column].astype('category')
df.dtypes
```

Out[ ]:

```
show_id           object
type            category
title             object
director          object
cast              object
country         category
date_added      category
release_year       int64
rating          category
duration        category
listed_in       category
description       object
day_added       category
year_added      category
month_added     category
dtype: object
```

In [ ]:

```
df["date_added"] = pd.to_datetime(df['date_added'])
df['day_added'] = df['date_added'].dt.day
df['year_added'] = df['date_added'].dt.year
```

```
df['month_added']=df['date_added'].dt.month
df.head(10)
```

Out[ ]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	2021-09-24	2021	TV-MA
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA
5	s6	TV Show	Midnight Mass	Mike Flanagan	Kate Siegel, Zach Gilford, Hamish Linklater, H...	NaN	2021-09-24	2021	TV-MA
6	s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	Vanessa Hudgens, Kimiko Glenn, James Marsden, ...	NaN	2021-09-24	2021	PG
7	s8	Movie	Sankofa	Haile Gerima	Kofi Ghanaba, Oyafunmike Ogunlano, Alexandra D...	United States, Ghana, Burkina Faso, United Kin...	2021-09-24	1993	TV-MA
8	s9	TV Show	The Great British Baking Show	Andy Devonshire	Mel Giedroyc, Sue Perkins, Mary Berry, Paul Ho...	United Kingdom	2021-09-24	2021	TV-14
9	s10	Movie	The Starling	Theodore Melfi	Melissa McCarthy,	United States	2021-09-24	2021	PG-13

show_id	type	title	director	cast	country	date_added	release_year	rating
				Chris O'Dowd, Kevin Kline, T...				

In [ ]:

```
df.dtypes
```

Out[ ]:

```
show_id           object
type            category
title             object
director          object
cast              object
country         category
date_added      category
release_year       int64
rating          category
duration        category
listed_in       category
description       object
day_added       category
year_added      category
month_added     category
dtype: object
```

In [ ]:

```
df.isnull().sum()
```

Out[ ]:

```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        10
release_year       0
rating             4
duration           3
listed_in          0
description        0
dtype: int64
```

In [ ]:

```
df.director.fillna("Not available",inplace=True)
df.cast.fillna("Not available",inplace=True)
df.country.fillna("Not available",inplace=True)
df.date_added.fillna("Not available",inplace=True)
df.rating.fillna("Not available",inplace=True)
df.duration.fillna("Not available",inplace=True)
df.day_added.fillna("Not available",inplace=True)
df.month_added.fillna("Not available",inplace=True)
df.year_added.fillna("Not available",inplace=True)
df.isnull().sum()
```

Out[ ]:

```
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
day_added       0
year_added      0
month_added     0
dtype: int64
```

In [ ]:

```
df.isnull().sum()
```

Out[ ]:

```
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
day_added       0
year_added      0
month_added     0
dtype: int64
```

In [ ]:

```
# Generate statistical summary for numerical attributes
df.describe()
```

Out[ ]:

	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

In [ ]:

```
df.type.value_counts()
```

Out[ ]:

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

In [ ]:

```
df.nunique()
```

```
Out[ ]: show_id         8807
        type               2
        title           8807
        director        4528
        cast            7692
        country          748
        date_added      1767
        release_year      74
        rating            17
        duration         220
        listed_in        514
        description     8775
        dtype: int64
```

```
In [ ]: df.cast.value_counts()
```

```
Out[ ]: David Attenborough
        19
        Vatsal Dubey, Julie Tejwani, Rupa Bhimani, Jigna Bhardwaj, Rajesh Kava, Mousam, Sw
        apnil
        14
        Samuel West
        10
        Jeff Dunham
        7
        David Spade, London Hughes, Fortune Feimster
        6

        ..
        Michael Peña, Diego Luna, Tenoch Huerta, Joaquin Cosio, José María Yazpik, Matt Le
        tscher, Alyssa Diaz
        1
        Nick Lachey, Vanessa Lachey
        1
        Takeru Sato, Kasumi Arimura, Haru, Kentaro Sakaguchi, Takayuki Yamada, Kendo Kobay
        ashi, Ken Yasuda, Arata Furuta, Suzuki Matsuo, Koichi Yamadera, Arata Iura, Chikak
        o Kaku, Kotaro Yoshida     1
        Toyin Abraham, Sambasa Nzeribe, Chioma Chukwuka Akpotha, Chioma Omeruah, Chiwetalu
        Agu, Dele Odule, Femi Adebayo, Bayray McNwizu, Biodun Stephen
        1
        Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanana, Manish Chaudhary, Meghna Malik, M
        alkeet Rauni, Anita Shabdish, Chittaranjan Tripathy
        1
        Name: cast, Length: 7692, dtype: int64
```

```
In [ ]: unique_actors = df['cast'].str.split(', ').explode().unique()
        unique_actors
```

```
Out[ ]: array([nan, 'Ama Qamata', 'Khosi Ngema', ..., 'Malkeet Rauni',
               'Anita Shabdish', 'Chittaranjan Tripathy'], dtype=object)
```

```
In [ ]: actor_counts = df['cast'].str.split(', ').explode().value_counts()
        actor_counts
```

```
Out[ ]: 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: cast, Length: 36439, dtype: int64
```

```
In [ ]: df.director.value_counts()
```

```
Out[ ]: Rajiv Chilaka                     19
        Raúl Campos, Jan Suter            18
        Marcus Raboy                      16
        Suhas Kadav                       16
        Jay Karas                         14
                                          ..
        Raymie Muzquiz, Stu Livingston     1
        Joe Menendez                       1
        Eric Bross                         1
        Will Eisenberg                     1
        Mozez Singh                        1
        Name: director, Length: 4528, dtype: int64
```

```
In [ ]: unique_directors = df['director'].str.split(', ').explode().unique()
        unique_directors
```

```
Out[ ]: array(['Kirsten Johnson', nan, 'Julien Leclercq', ..., 'Majid Al Ansari',
               'Peter Hewitt', 'Mozez Singh'], dtype=object)
```

```
In [ ]: director_counts = df['director'].str.split(', ').explode().value_counts()
        director_counts
```

```
Out[ ]: 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: director, Length: 4993, dtype: int64
```

```
In [ ]: unique_countries = df['country'].str.split(', ').explode().unique()
        unique_countries
```

Out[ ]:

```
array(['United States', 'South Africa', nan, 'India', 'Ghana',
       'Burkina Faso', 'United Kingdom', 'Germany', 'Ethiopia',
       'Czech Republic', 'Mexico', 'Turkey', 'Australia', 'France',
       'Finland', 'China', 'Canada', 'Japan', 'Nigeria', 'Spain',
       'Belgium', 'South Korea', 'Singapore', 'Italy', 'Romania',
       'Argentina', 'Venezuela', 'Hong Kong', 'Russia', '', 'Ireland',
       'Nepal', 'New Zealand', 'Brazil', 'Greece', 'Jordan', 'Colombia',
       'Switzerland', 'Israel', 'Taiwan', 'Bulgaria', 'Algeria', 'Poland',
       'Saudi Arabia', 'Thailand', 'Indonesia', 'Egypt', 'Denmark',
       'Kuwait', 'Netherlands', 'Malaysia', 'Vietnam', 'Hungary',
       'Sweden', 'Lebanon', 'Syria', 'Philippines', 'Iceland',
       'United Arab Emirates', 'Norway', 'Qatar', 'Mauritius', 'Austria',
       'Cameroon', 'Palestine', 'Uruguay', 'United Kingdom,', 'Kenya',
       'Chile', 'Luxembourg', 'Cambodia', 'Bangladesh', 'Portugal',
       'Cayman Islands', 'Senegal', 'Serbia', 'Malta', 'Namibia',
       'Angola', 'Peru', 'Mozambique', 'Cambodia,', 'Belarus', 'Zimbabwe',
       'Puerto Rico', 'Pakistan', 'Cyprus', 'Guatemala', 'Iraq', 'Malawi',
       'Paraguay', 'Croatia', 'Iran', 'West Germany', 'United States,',
       'Albania', 'Georgia', 'Soviet Union', 'Morocco', 'Slovakia',
       'Ukraine', 'Bermuda', 'Ecuador', 'Armenia', 'Mongolia', 'Bahamas',
       'Sri Lanka', 'Latvia', 'Liechtenstein', 'Cuba', 'Nicaragua',
       'Poland,', 'Slovenia', 'Dominican Republic', 'Samoa', 'Azerbaijan',
       'Botswana', 'Vatican City', 'Jamaica', 'Kazakhstan', 'Lithuania',
       'Afghanistan', 'Somalia', 'Sudan', 'Panama', 'Uganda',
       'East Germany', 'Montenegro'], dtype=object)
```

In [ ]:

```python
country_counts = df['country'].str.split(', ').explode().value_counts()
country_counts
```

Out[ ]:

```
United States     3689
India             1046
United Kingdom     804
Canada             445
France             393
                  ... 
Bermuda              1
Ecuador              1
Armenia              1
Mongolia             1
Montenegro           1
Name: country, Length: 127, dtype: int64
```

In [ ]:

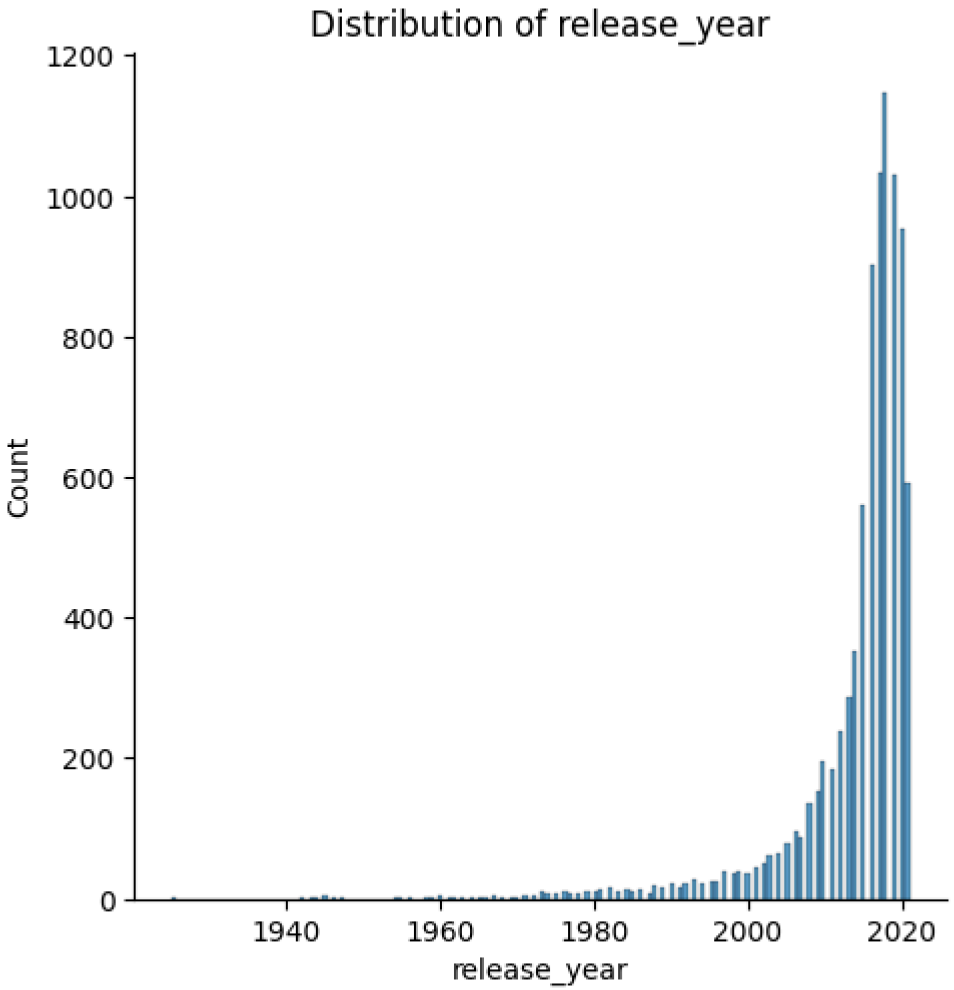
```python
from scipy.stats import zscore

# Compute z-scores for numerical columns
z_scores = df.select_dtypes(include=['float64', 'int64']).apply(zscore)

# Set a threshold for z-score (e.g., 3 standard deviations)
threshold = 3
outliers = (z_scores > threshold).any(axis=1)

# Print rows with outliers
print("Rows with outliers:")
print(df[outliers])
```

```
Rows with outliers:
Empty DataFrame
Columns: [show_id, type, title, director, cast, country, date_added, release_year,
rating, duration, listed_in, description]
Index: []
```

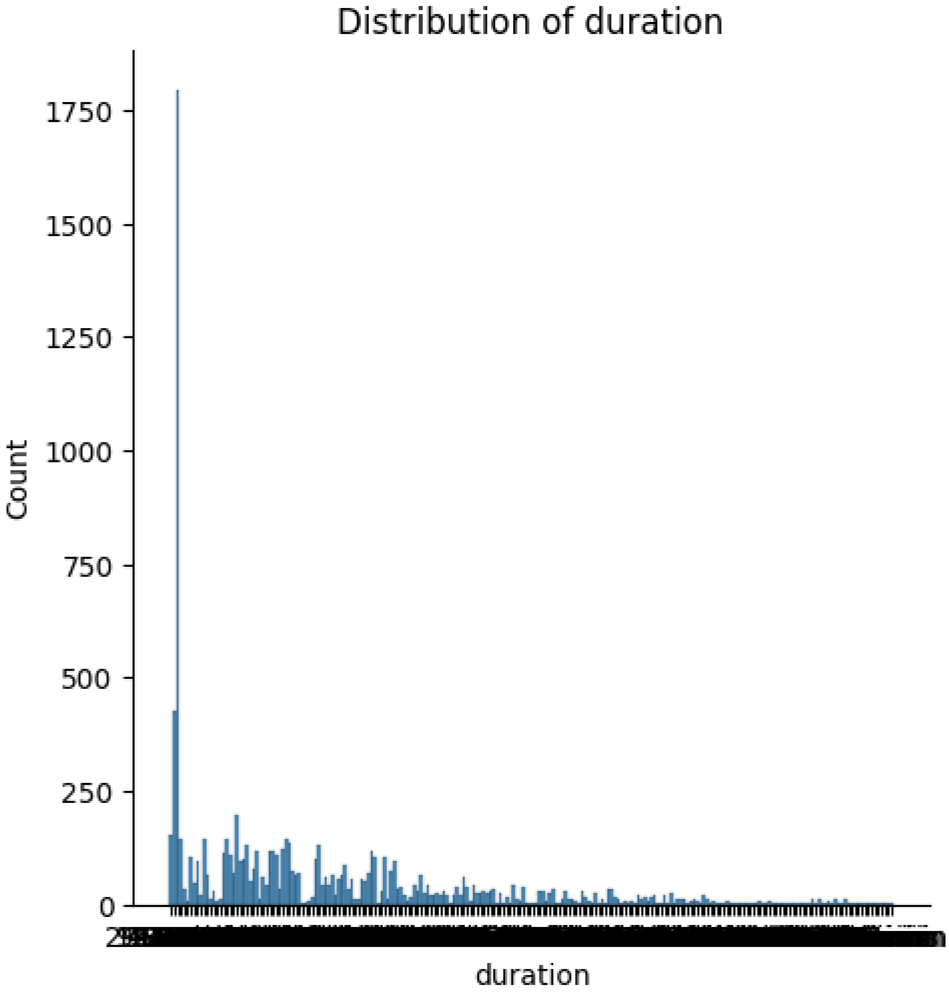
In [ ]:

```python
# Univariate analysis for continuous variables
continuous_vars = ['release_year', 'duration', 'rating']
for var in continuous_vars:
    plt.figure(figsize=(66, 56))
    # distplot
    sns.displot(df[var], kde=False)
    plt.title(f'Distribution of {var}')
    plt.xlabel(var)
    plt.ylabel('Count')
    plt.show()
```

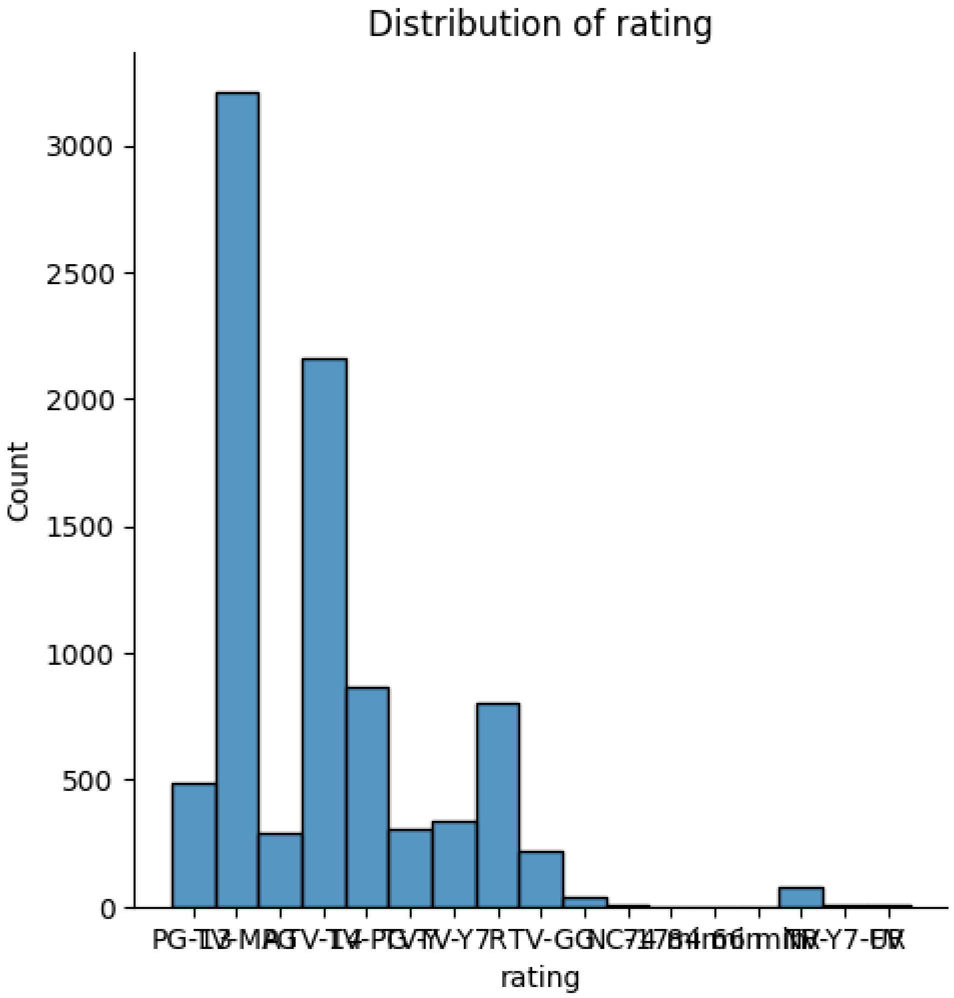
```
<Figure size 6600x5600 with 0 Axes>
```



```
<Figure size 6600x5600 with 0 Axes>
```


Distribution of duration
Count
1750
1500
1250
1000
750
500
250
0
duration

```
<Figure size 6600x5600 with 0 Axes>
```

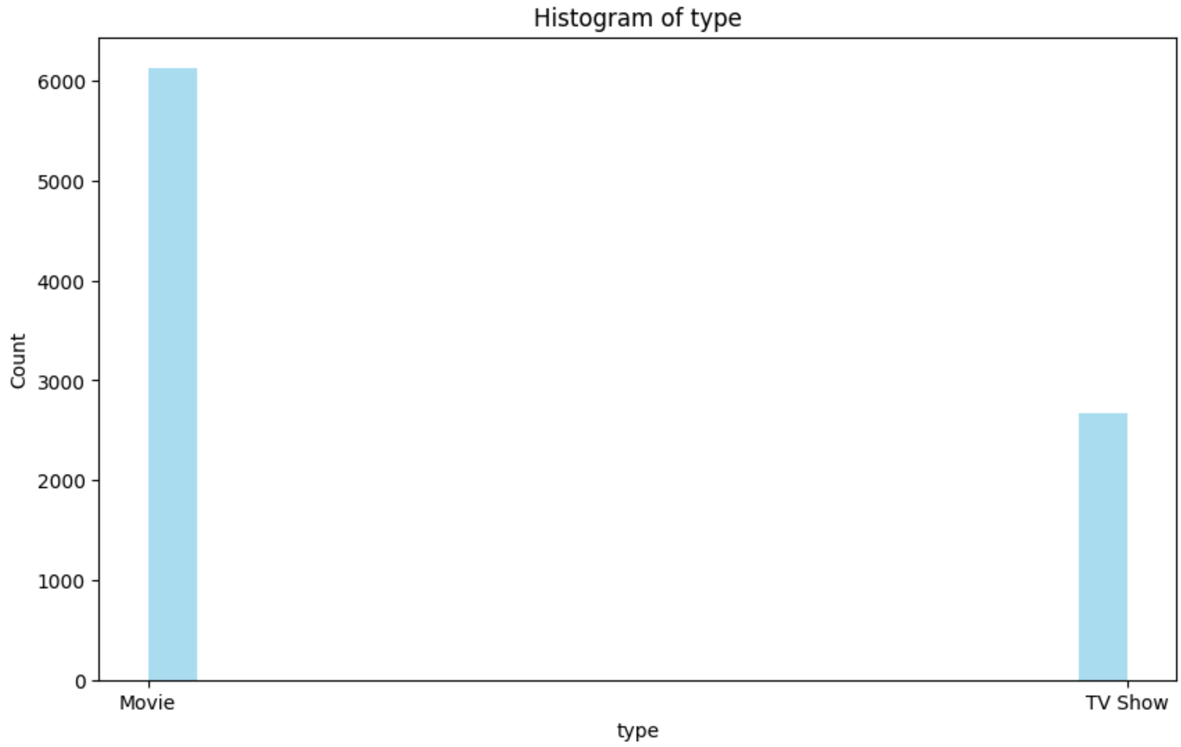


Insights:

The distribution of release years shows variations, indicating that movies have been released over a range of years. The distribution of movie durations shows that most movies fall within a certain range of durations, likely reflecting industry standards or viewer preferences. The distribution of ratings reveals the frequency of different rating categories assigned to movies.

In [ ]:

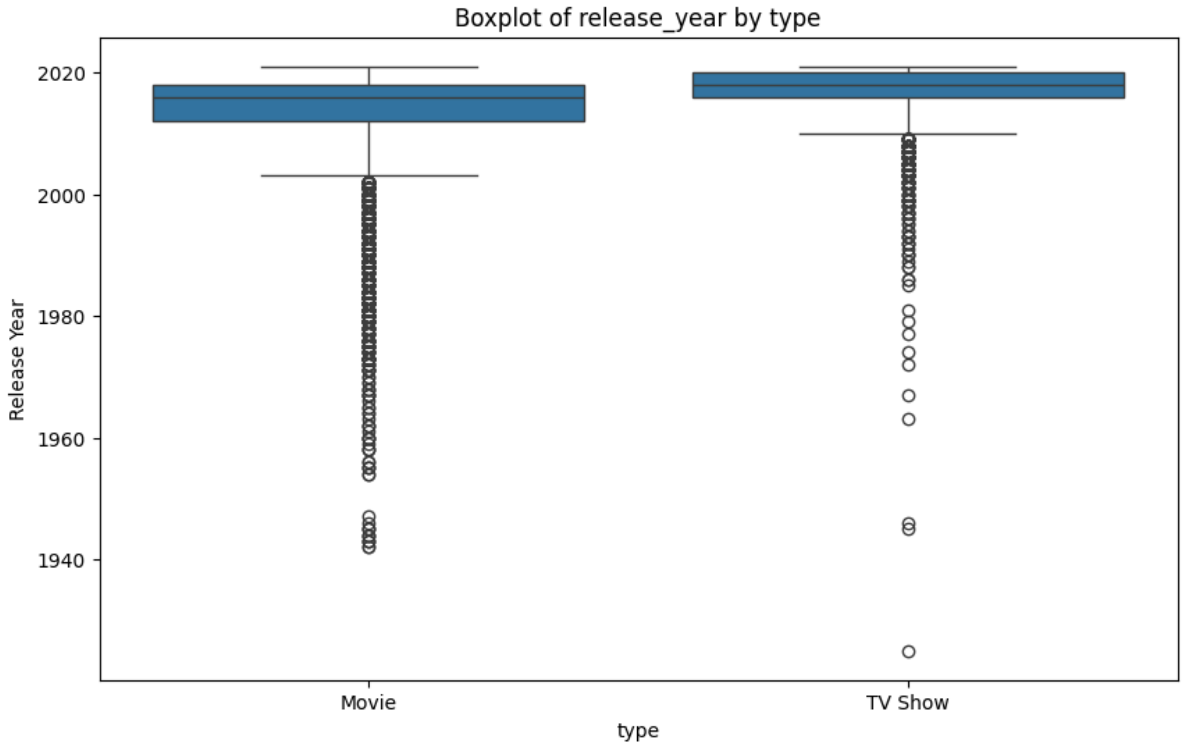
```
plt.figure(figsize=(10, 6))
plt.hist(df[var], bins=20, color='skyblue', alpha=0.7)
plt.title(f'Histogram of {var}')
plt.xlabel(str(var))  # Ensure var is converted to a string
plt.ylabel('Count')
plt.show()
```

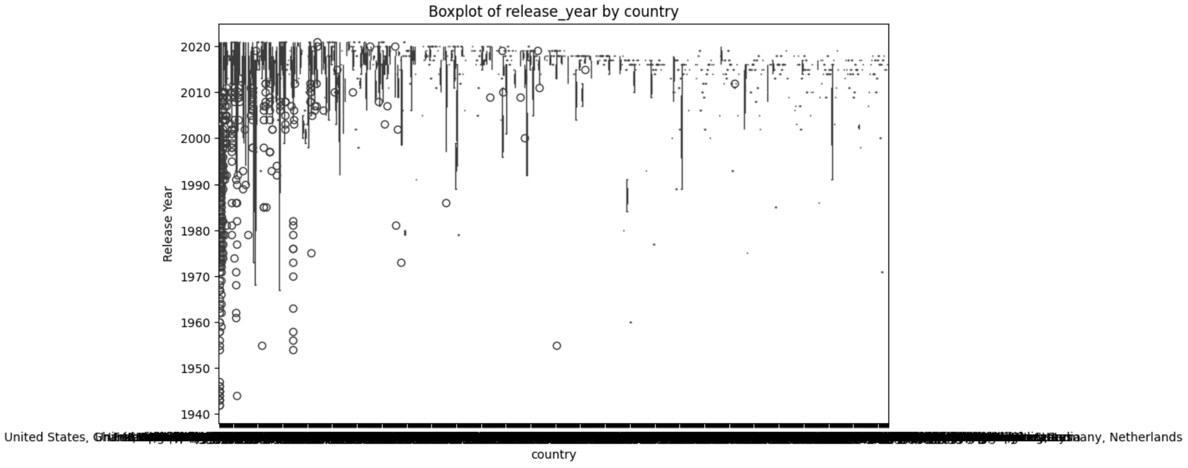


Insights:

The histogram provides insights into the distribution shape, central tendency, spread, presence of outliers, skewness, and patterns within the data for the continuous variable var. These insights help understand the data's characteristics and guide further analysis and decision-making processes.

```
In [ ]: # Conduct univariate analysis for categorical variables using boxplot
categorical_vars = ['type', 'country']
for var in categorical_vars:
    plt.figure(figsize=(10, 6))
    sns.boxplot(x=df[var], y= 'release_year', data=df)
    plt.title(f'Boxplot of release_year by {var}')
    plt.xlabel(var)
    plt.ylabel('Release Year')
    plt.show()
```



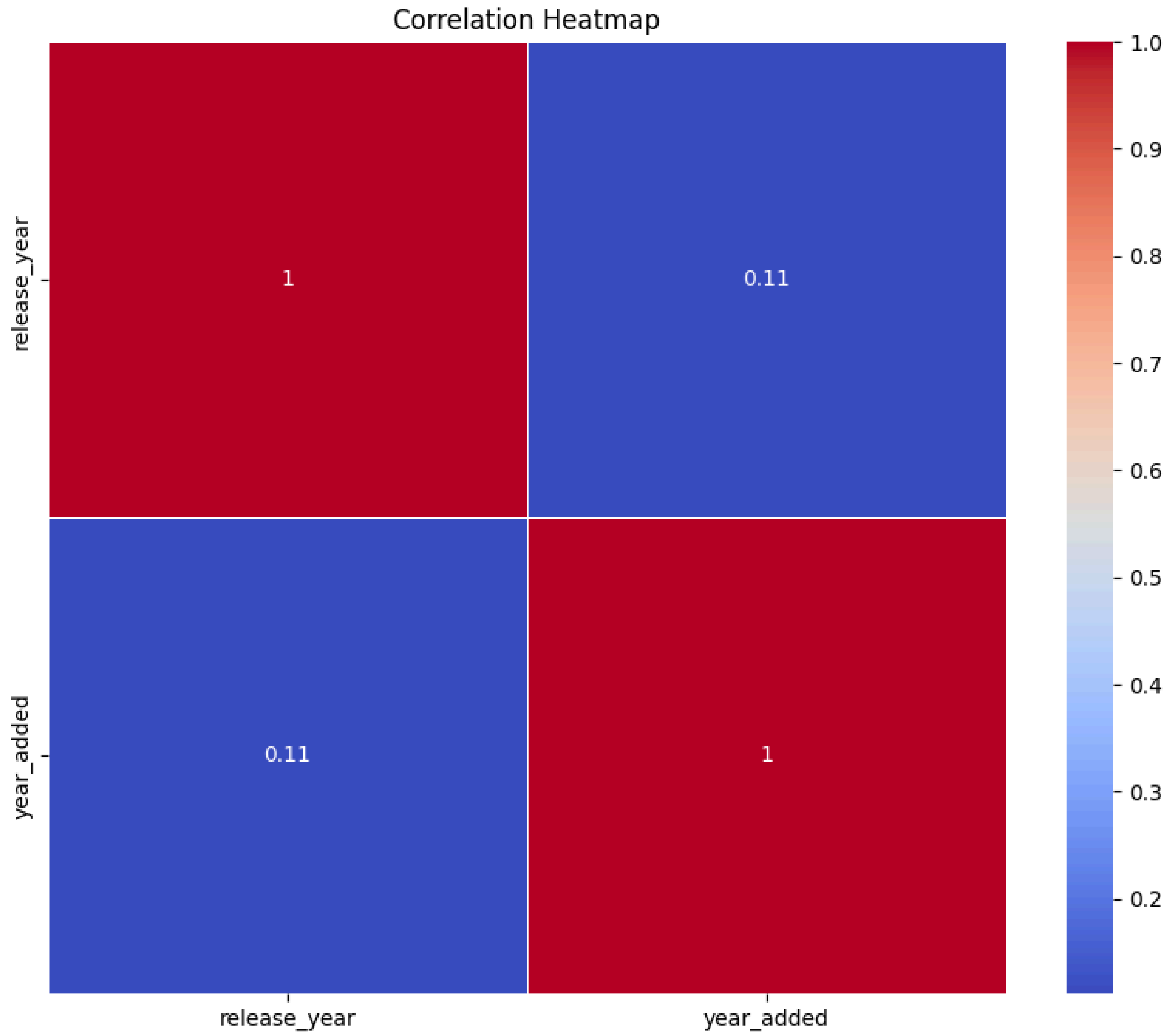


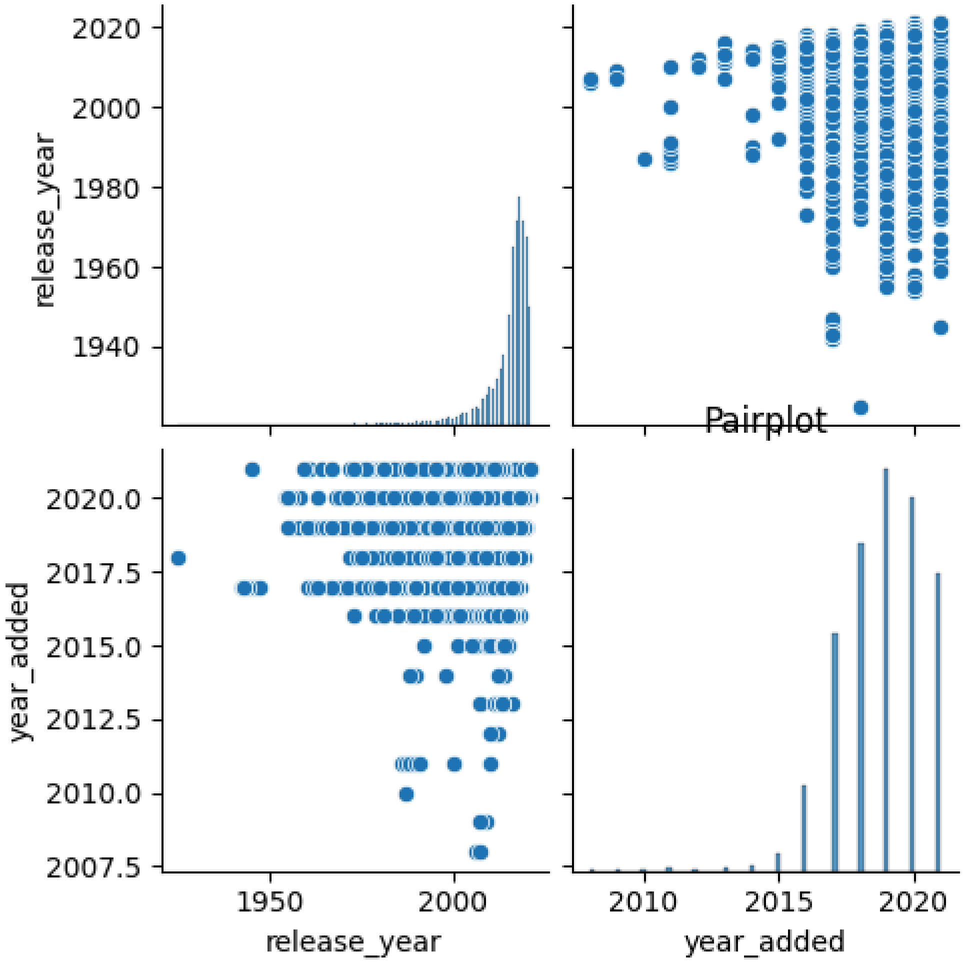
Insights:

By analyzing the boxplots of release year against categorical variables, we gain insights into how content type and country contribute to variations in the distribution of release years. These insights can inform content acquisition strategies, regional content preferences, and audience targeting efforts to enhance viewer engagement and satisfaction on the streaming platform.

```
In [ ]: # Visualize correlations using heatmaps and pairplots
        plt.figure(figsize=(10, 8))
        sns.heatmap(df.corr(), annot=True, cmap='coolwarm', linewidths=0.5)
        plt.title('Correlation Heatmap')
        plt.show()

        sns.pairplot(df)
        plt.title('Pairplot')
        plt.show()
```

```
<ipython-input-199-327cb59200a0>:3: FutureWarning: The default value of numeric_on
ly in DataFrame.corr is deprecated. In a future version, it will default to False.
Select only valid columns or specify the value of numeric_only to silence this war
ning.
  sns.heatmap(df.corr(), annot=True, cmap='coolwarm', linewidths=0.5)
```



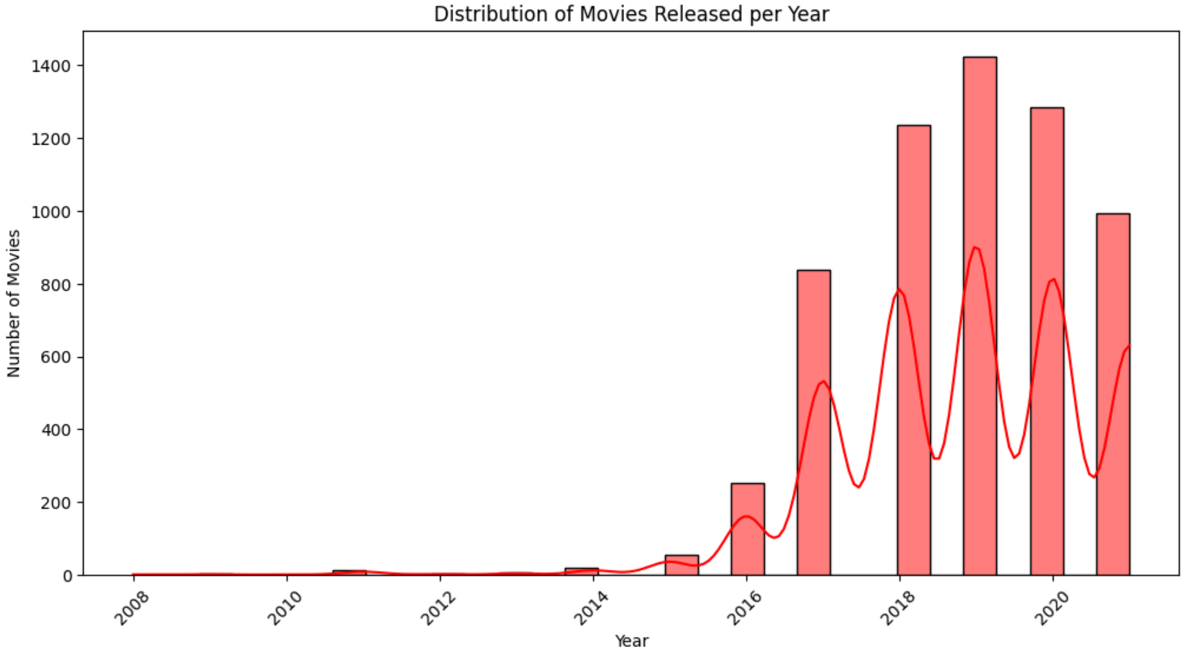


Insights:

These visualizations help in identifying patterns, trends, and relationships within the dataset. They assist in understanding how variables are related to each other and can guide further analysis and modeling decisions. Additionally, they provide insights into potential multicollinearity issues and variables that may be important predictors for modeling tasks.

***NUMBER OF MOVIES RELEASED PER YEAR CHANGED OVER THE LAST 20-30 YEARS

```
In [ ]: df['date_added'] = pd.to_datetime(df['date_added'])
        df['year_added'] = df['date_added'].dt.year
        movies_df = df[df['type'] == 'Movie']
        plt.figure(figsize=(12, 6))
        sns.histplot(data=movies_df, x='year_added', bins=30, kde=True, color='r')
        plt.title('Distribution of Movies Released per Year')
        plt.xlabel('Year')
        plt.ylabel('Number of Movies')
        plt.xticks(rotation=45)
        plt.show()
```



Insights:

The histogram shows the trend in the number of movies released per year over the available years in the dataset.The overall trend may show an increasing or decreasing pattern in movie releases over the years, indicating changes in the film industry's dynamics.

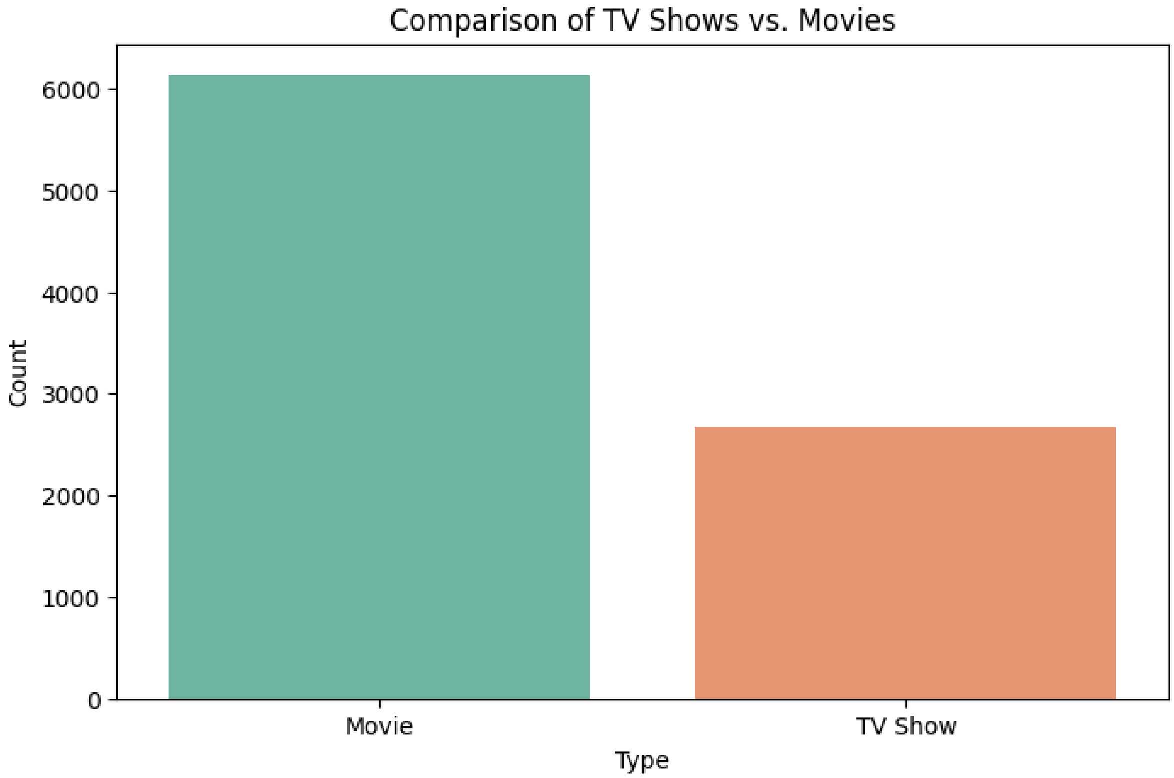
Recoomendations:

Continuously monitor trends in movie releases to stay informed about changes in viewer preferences, industry dynamics, and competitive landscape.Use insights from the distribution of movies released per year to inform content planning and scheduling decisions, ensuring a balanced and strategic approach to movie acquisitions and releases.

***COMPARISON OF TV SHOWS AND MOVIES

In [ ]:
```
plt.figure(figsize=(8, 5))
sns.countplot(data=df, x='type', palette='Set2')
plt.title('Comparison of TV Shows vs. Movies')
plt.xlabel('Type')
plt.ylabel('Count')
plt.show()
```

```
<ipython-input-171-9c5a3cbb1f29>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.
14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(data=df, x='type', palette='Set2')
```



Insights:

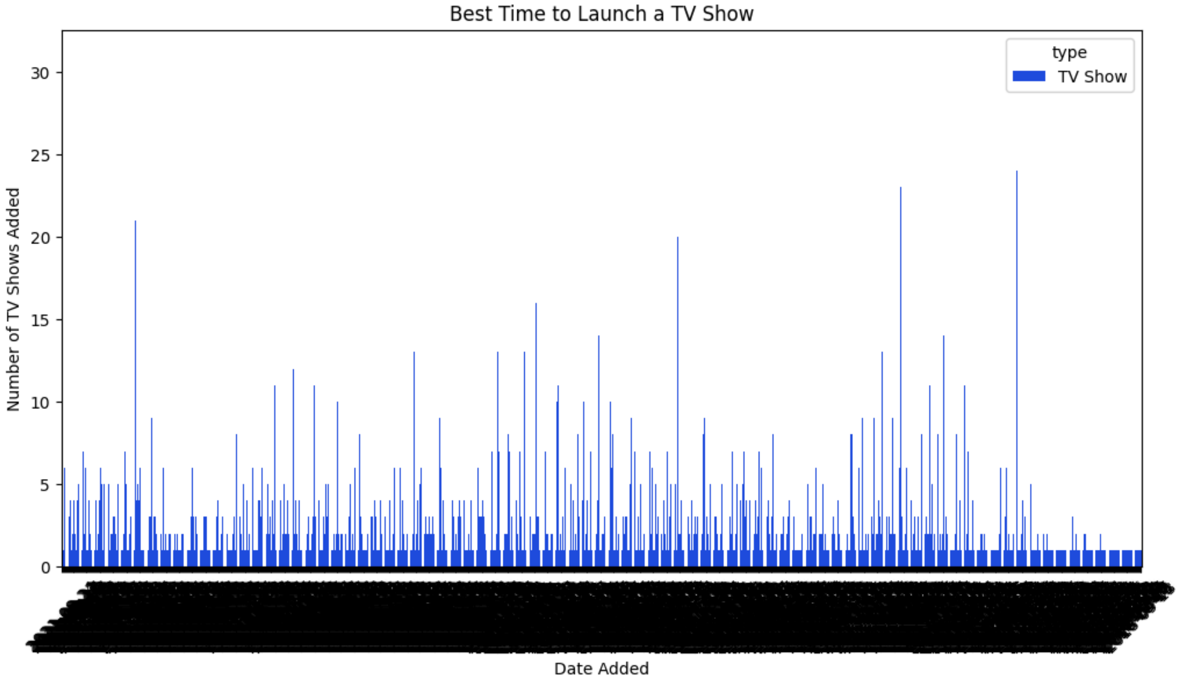
By understanding the distribution of TV shows versus movies, businesses can make informed decisions regarding content acquisition, promotion, and curation strategies to enhance viewer engagement and satisfaction on the platform.

Recoomendations:

Given the higher prevalence of movies, consider diversifying the content library by acquiring more TV shows to cater to a wider range of viewer preferences.Develop targeted promotional strategies to highlight the diversity of content offerings, emphasizing the availability of both movies and TV shows to attract and retain viewers.

***BEST TIME TO LAUNCH A TV SHOW

In [ ]:
```
tv_shows_df = df[df['type'] == 'TV Show']
plt.figure(figsize=(12, 6))
sns.countplot(data=tv_shows_df, x='date_added', hue = "type", palette='bright')
plt.title('Best Time to Launch a TV Show')
plt.xlabel('Date Added')
plt.ylabel('Number of TV Shows Added')
plt.xticks(rotation=45)
plt.show()
```



Insights:

Analyze the fluctuations in the number of TV shows added each month. Look for any recurring patterns or seasonal trends. For example, there may be peaks in TV show additions during certain months, possibly corresponding to holiday seasons or promotional periods. Monitor the overall trend of TV show additions over time. Are there noticeable increases or decreases in the number of TV shows added each month? This could provide insights into the growth trajectory of the streaming platform.

Recoomendations:

Optimal Launch Timing: Use insights from seasonal trends to identify the best time to launch a new TV show. Aim to capitalize on periods of high user engagement or demand, such as during holiday seasons or when there are fewer competing releases.Content Acquisition Strategy: Adjust the content acquisition strategy based on observed trends. For example, if certain genres consistently perform well during specific months, prioritize acquiring content in those genres during those times to attract and retain viewers.Promotional Campaigns: Plan promotional campaigns and marketing efforts around peak periods of TV show additions to maximize visibility and audience reach. This could include teaser trailers, social media campaigns, or targeted advertising to generate buzz and anticipation for new releases.

```
In [ ]: # 4. Analysis of actors/directors of different types of shows/movies.
        actors_df = df['cast'].str.split(', ').explode().reset_index(drop=True)
        actors_df
```

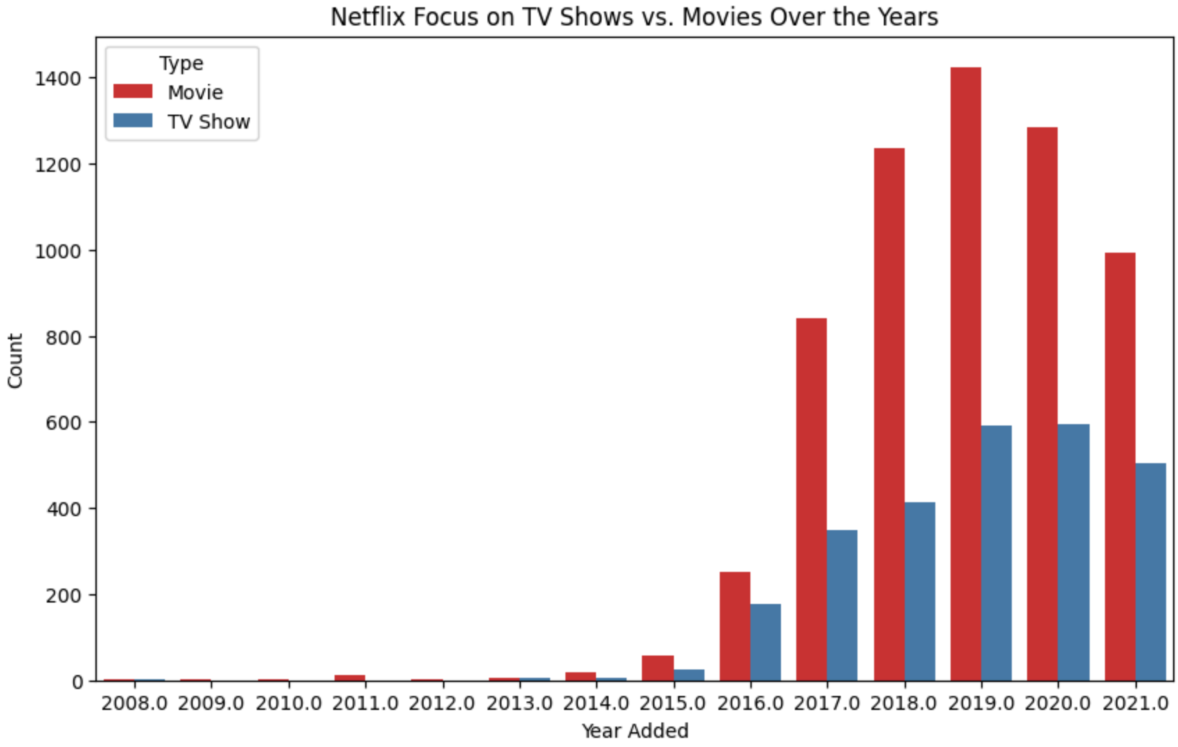
```
Out[ ]: 0                           NaN
        1                   Ama Qamata
        2                  Khosi Ngema
        3                Gail Mabalane
        4               Thabang Molaba
                             ...
        64946          Manish Chaudhary
        64947              Meghna Malik
        64948             Malkeet Rauni
        64949            Anita Shabdish
        64950      Chittaranjan Tripathy
        Name: cast, Length: 64951, dtype: object
```

```
In [ ]: directors_df = df['director'].str.split(', ').explode().reset_index(drop=True)
        directors_df
```

```
Out[ ]: 0       Kirsten Johnson
        1                   NaN
        2       Julien Leclercq
        3                   NaN
        4                   NaN
                      ...
        9607      David Fincher
        9608                NaN
        9609    Ruben Fleischer
        9610       Peter Hewitt
        9611        Mozez Singh
        Name: director, Length: 9612, dtype: object
```

***NETFLIX FOCUS ON TV SHOWS VS MOVIES OVER THE YEARS

```
In [ ]: plt.figure(figsize=(10, 6))
        sns.countplot(data=df, x='year_added', hue='type', palette='Set1')
        plt.title('Netflix Focus on TV Shows vs. Movies Over the Years')
        plt.xlabel('Year Added')
        plt.ylabel('Count')
        plt.legend(title='Type')
        plt.show()
```



Insights: & Recoomendations:

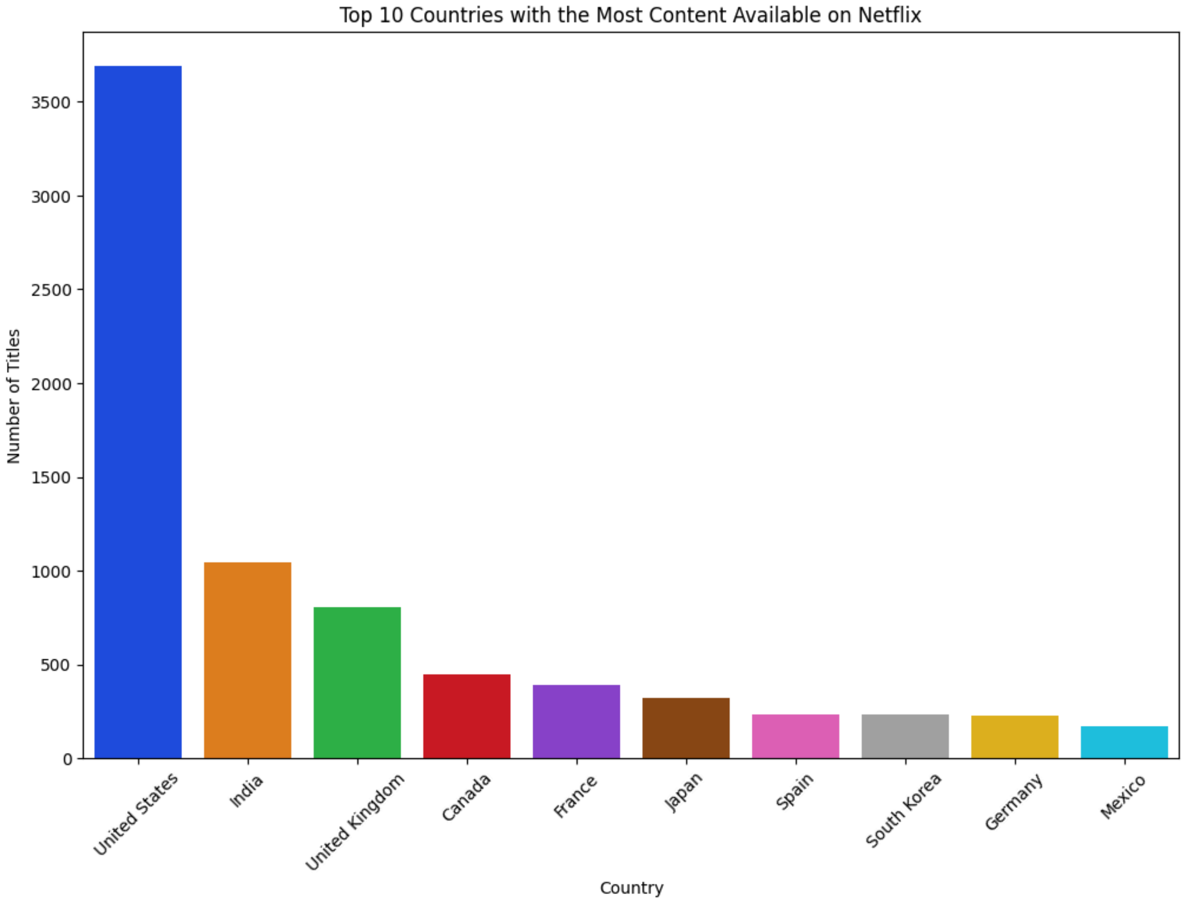
"In the ever-expanding realm of streaming entertainment, platforms like Netflix are continuously refining their content strategies to cater to diverse audience preferences. A glance at the historical data reveals intriguing insights into Netflix's focus on TV shows versus movies over the years. Through meticulous analysis of the count of TV shows and movies added annually, patterns emerge, reflecting the platform's dynamic approach to content acquisition and audience engagement. By leveraging these insights, Netflix can craft tailored strategies to optimize its content portfolio, ensuring a compelling and immersive viewing experience for subscribers worldwide.

***COUNTRIES WITH MOST CONTENT AVAILABLE ON NETFLIX

In [ ]:

```
plt.figure(figsize=(12, 8))
top_countries = df['country'].str.split(', ').explode().value_counts().head(10)
sns.barplot(x=top_countries.index, y=top_countries.values, palette='bright')
plt.title('Top 10 Countries with the Most Content Available on Netflix')
plt.xlabel('Country')
plt.ylabel('Number of Titles')
plt.xticks(rotation=45)
plt.show()
```

```
<ipython-input-192-1bdcb4130915>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=top_countries.index, y=top_countries.values, palette='bright')
```



Insights: & Recoomendations:

"The battle for supremacy in the realm of streaming entertainment rages on, with Netflix emerging as a global powerhouse. A closer look at the data unveils the top 10 countries contributing to Netflix's vast content library. The United States leads the charge, boasting a rich tapestry of titles, followed closely by other entertainment hubs like the United Kingdom and India. This diverse array of content reflects Netflix's commitment to catering to a worldwide audience, transcending geographical boundaries to deliver captivating stories and compelling narratives. As Netflix continues to expand its global footprint, these insights serve as a testament to its unwavering dedication to providing unparalleled entertainment experiences to subscribers around the globe."

THE END

```
In [ ]:
```