Business Problem

Analyze the data and generate insights that could help Netflix deciding which type of shows/movies to produce and how they can grow the business in different countries

```
[1]  import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     import warnings
     warnings.filterwarnings('ignore')
```

```
[4]  netflix_data = pd.read_csv('netflix_titles.csv')
```

```
# Get Basic Information about the DataFrame
netflix_data.head(5)
```

	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	After crossing paths at a party, a Cape Town t...
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...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
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 ...	In a city of coaching centers known to train I...

Next steps: Generate code with netflix_data View recommended plots

```
#1. Analysing basic metrics
# Checking the shape of the dataset
dataset_shape = netflix_data.shape

# Checking data types of the attributes
dataset_dtypes = netflix_data.dtypes

dataset_shape, dataset_dtypes
```

```
((8807, 12),
 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)
```

The dataset consists of 8,807 entries with 12 attributes

ANALYZING BASIC METRICS:

```
# Displaying basic metrics using the describe() method for numerical columns
basic_metrics_numerical = netflix_data.describe()

# Displaying basic metrics for categorical columns like 'Type', 'Country', and 'Rating'
basic_metrics_categorical = netflix_data[['type', 'country', 'rating']].describe(include=['object'])

basic_metrics_numerical, basic_metrics_categorical
```

```
(       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,
          type        country rating
 count    8807           7976   8803
 unique      2            748     17
 top     Movie  United States  TV-MA
 freq     6131           2818   3207)
```

Numerical Attributes

For the numerical attribute `release_year`:

- Count: 8,807 entries
- Mean: Around the year 2014
- Standard Deviation: Approximately 8.82 years
- Minimum: Year 1925
- 25th Percentile (Q1): Year 2013
- Median (50th Percentile): Year 2017
- 75th Percentile (Q3): Year 2019
- Maximum: Year 2021

Categorical Attributes

- For the categorical attributes type, country, and rating:

Type

- Count: 8,807
- Unique Values: 2 (Movie, TV Show)
- Most Frequent: Movie
- Frequency: 6,131

Country

- Count: 7,976 (some missing values)
- Unique Values: 748
- Most Frequent: United States
- Frequency: 2,818

Rating

- Count: 8,803 (some missing values)
- Unique Values: 17
- Most Frequent: TV-MA
- Frequency: 3,207

Observations:

- The average release_year being around 2014, along with a median of 2017, suggests that Netflix has a lot of content from the recent decade.
- The high frequency of Movies compared to TV
- Shows indicates a stronger focus on movie content.
- The United States appears to be the most common country for content production, followed by a wide range of other countries, indicating a diverse content catalog.
- The rating “TV-MA” is the most frequent, suggesting a focus on mature audiences.

These basic metrics offer a snapshot of the kind of content that is prevalent on Netflix, which can be quite informative for various business decisions.

```
[6] #2. Observations on the shape of data, data types of all the attributes, conversion of categorical attributes
    # Convert categorical attributes to 'category' data type if required
    categorical_columns = ['type', 'country', 'rating']
    netflix_data[categorical_columns] = netflix_data[categorical_columns].astype('category')

    # After conversion data types
    after_conversion_data_types = netflix_data.dtypes
    # Missing value detection
    missing_values = netflix_data.isnull().sum()
```

```
missing_values
```

```
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
```

- Most of the attributes are of object data type, except release_year, which is an int64.
- Conversion of Categorical Attributes to ‘Category’
- The data types for type, country, and rating have been converted to category.
- Missing Value Detection
  - director: 2,634 missing values
  - cast: 825 missing values
  - country: 831 missing values
  - date_added: 10 missing values
  - rating: 4 missing values
  - duration: 3 missing values

```
# Non-Graphical Analysis: Value counts for key attributes
value_counts_type = netflix_data['type'].value_counts()
value_counts_country = netflix_data['country'].value_counts().head(10)  # Top 10 countries
value_counts_rating = netflix_data['rating'].value_counts()
value_counts_release_year = netflix_data['release_year'].value_counts().head(10)  # Top 10 release years

# Unique attributes for key columns
unique_type = netflix_data['type'].unique()
unique_country = netflix_data['country'].unique()
unique_rating = netflix_data['rating'].unique()
unique_release_year = netflix_data['release_year'].unique()

value_counts_type, value_counts_country, value_counts_rating, value_counts_release_year, unique_type, unique_country, unique_rating, unique_release_year
```

```
Egypt                 166
Name: count, dtype: int64,
rating
TV-MA        3207
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,
release_year
2018    1147
2017    1032
2019    1030
2020     953
2016     902
2021     592
2015     560
2014     352
2013     288
2012     237
Name: count, dtype: int64,
```

```
NC-17           3
74 min          1
84 min          1
66 min          1
Name: count, dtype: int64,
release_year
2018    1147
2017    1032
2019    1030
2020     953
2016     902
2021     592
2015     560
2014     352
2013     288
2012     237
Name: count, dtype: int64,
['Movie', 'TV Show']
Categories (2, object): ['Movie', 'TV Show'],
['United States', 'South Africa', NaN, 'India', 'United States, Ghana, Burkina Faso, United Ki..., ..., 'Russia, Spain', 'Croatia, Slovenia, Serbia, Montenegro', 'Japan, Canada',
'United States, France, South Korea, Indonesia', 'United Arab Emirates, Jordan']
Length: 749
Categories (748, object): [', France, Algeria', ', South Korea', 'Argentina',
                           'Argentina, Brazil, France, Poland, Germany, D..., ..., 'Venezuela, Colombia', 'Vietnam', 'West Germany',
                           'Zimbabwe'],
['PG-13', 'TV-MA', 'PG', 'TV-14', 'TV-PG', ..., '66 min', 'NR', NaN, 'TV-Y7-FV', 'UR']
Length: 18
Categories (17, object): ['66 min', '74 min', '84 min', 'G', ..., 'TV-Y', 'TV-Y7', 'TV-Y7-FV', 'UR'],
array([2020, 2021, 1993, 2018, 1996, 1998, 1997, 2010, 2013, 2017, 1975,
       1978, 1983, 1987, 2012, 2001, 2014, 2002, 2003, 2004, 2011, 2008,
       2009, 2007, 2005, 2006, 1994, 2015, 2019, 2016, 1982, 1989, 1990,
       1991, 1999, 1986, 1992, 1984, 1980, 1961, 2000, 1995, 1985, 1976,
       1959, 1988, 1981, 1972, 1964, 1945, 1954, 1979, 1958, 1956, 1963,
```

Value Counts

Type of Content (Movies vs. TV Shows)

●Movies: 6,131

●TV Shows: 2,676

Top 10 Countries Producing Content

●United States: 2,818

●India: 972

●United Kingdom: 419

●Japan: 245

South Korea: 199

●Canada: 181

●Spain: 145

●France: 124

●Mexico: 110

●Egypt: 106

Top 10 Release Years

2018: 1,147

●2017: 1,032

●2019: 1,030

●2020: 953

●2016: 902

Unique Attributes

1. Type: 2 unique values (‘Movie’, ‘TV Show’)

2. Country: 748 unique values

3. Rating: 17 unique values

4. Release Year: Ranges from 1925 to 2021

Observations

●The platform predominantly offers movies, almost twice as many as TV Shows.

●The United States is the leading country in producing content, followed by India and the United Kingdom.

The most common ratings are ‘TV-MA’ and ‘TV-14’, indicating a focus on mature and teen audiences.

●Most of the content was released in the years 2018, 2017, and 2019, showing a strong focus on recent content.

This non-graphical analysis provides a solid foundation for understanding the overall composition of the dataset. It also offers valuable insights into the types of content that are most prevalent on Netflix.

Missing Value & Outlier Check

Before we proceed further with additional visualizations, it’s crucial to check for missing values and outliers.

Missing Values

Let’s first check for missing values in the dataset.

```
# Checking for missing values
missing_values = netflix_data.isnull().sum()
missing_values
```

```
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
```

Observations:

We have missing values in several columns:

●director: 2,634 missing values

●cast: 825 missing values

●country: 831 missing values

●date_added: 10 missing values

●rating: 4 missing values

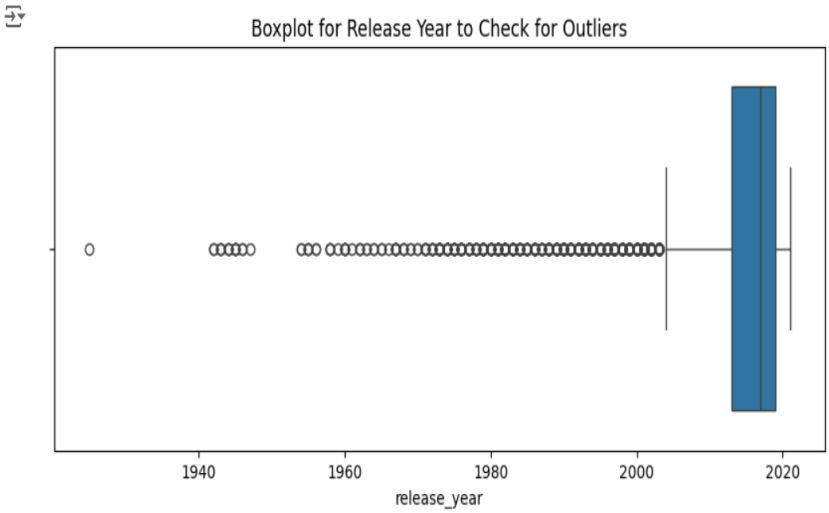
●duration: 3 missing values

Given the nature of our analysis, these missing values may or may not significantly impact the outcome. For example, missing director or cast information may not be crucial for our current business questions

Outliers:

For the purpose of this analysis, we'll focus on the release_year as our primary numerical variable. Let's check for outliers using a boxplot

```
# Boxplot to check for outliers in 'release_year'
plt.figure(figsize=(10, 4))
sns.boxplot(x=netflix_data['release_year'])
plt.title('Boxplot for Release Year to Check for Outliers')
plt.show()
```



Observations:

The boxplot for release_year shows no significant outliers, indicating that the data for this attribute is relatively consistent.

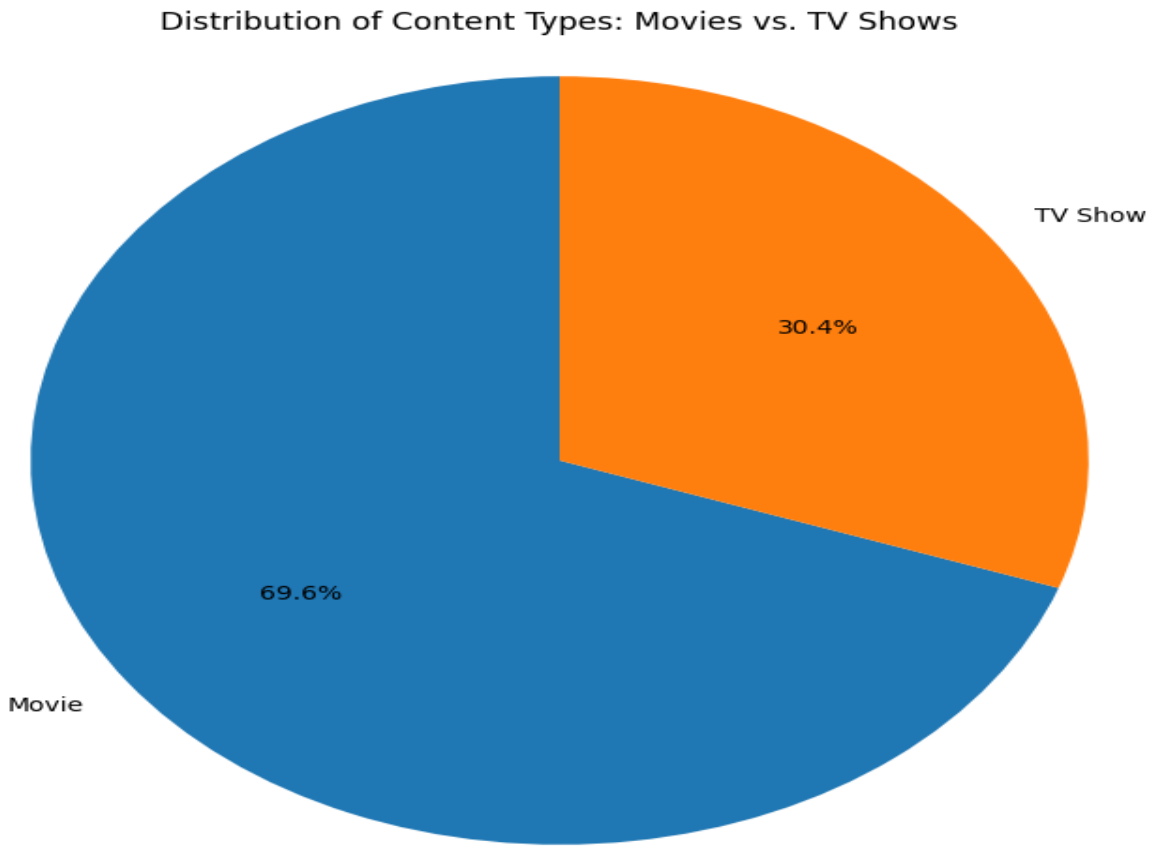
Univariate Analysis

Countplot for Type of Content

Let’s start by understanding the distribution of Movies vs. TV Shows on Netflix.

```
[11] #4 qs
     # Unnesting the data in columns like Actor, Director, Country
     # Splitting the comma-separated values and unnesting
     unnested_director = netflix_data.assign(director=netflix_data['director'].str.split(', ')).explode('director')
     unnested_cast = netflix_data.assign(cast=netflix_data['cast'].str.split(', ')).explode('cast')
     unnested_country = netflix_data.assign(country=netflix_data['country'].str.split(', ')).explode('country')
```

```
# 4.1
# Univariate Example with Pie Chart for 'Type' (Movie/TV Show)
type_counts = netflix_data['type'].value_counts()
labels = type_counts.index
sizes = type_counts.values

plt.figure(figsize=(8, 8))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=90)
plt.title('Distribution of Content Types: Movies vs. TV Shows')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```

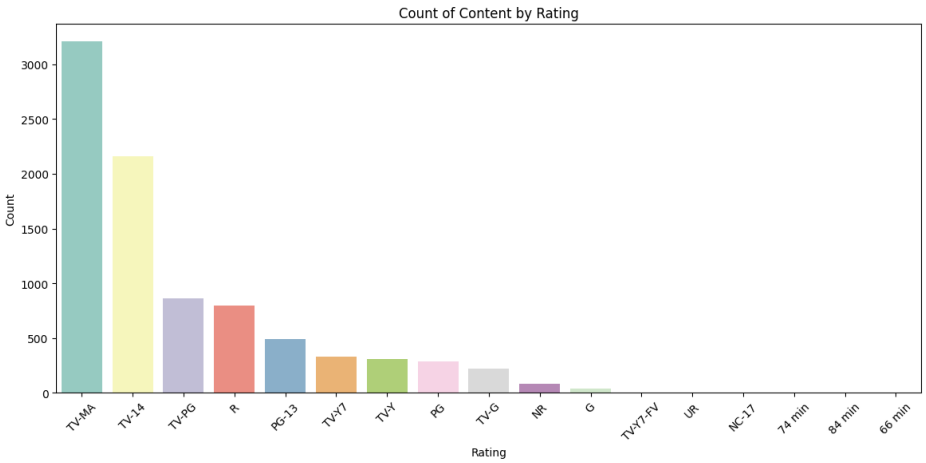


Observations:

● The count of Movies is significantly higher than that of TV Shows, indicating that Netflix has a more extensive catalog of movies.

Count Plot for ratings:

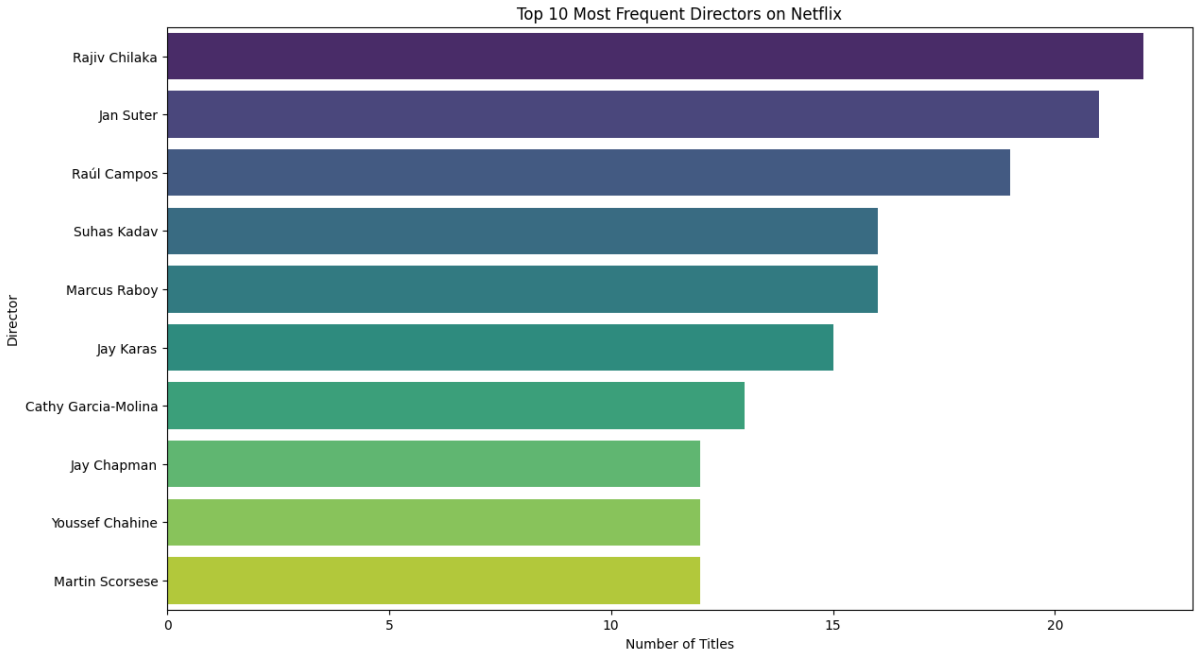
```
# Countplot for Rating
plt.figure(figsize=(14, 6))
sns.countplot(x='rating', data=netflix_data, order=netflix_data['rating'].value_counts().index, palette='Set3')
plt.title('Count of Content by Rating')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.show()
```



Observations:

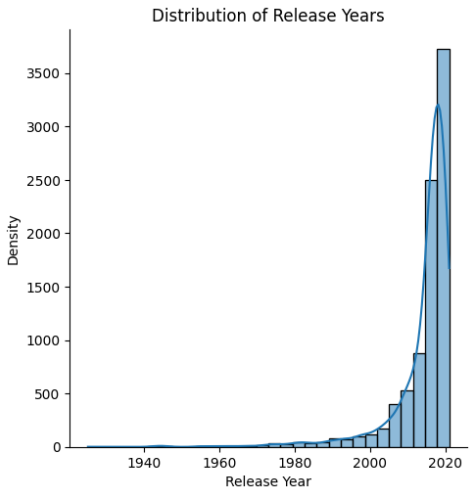
● The majority of the content is rated “TV-MA” followed by “TV-14”, indicating a focus on mature audiences and teenagers.

```
[14] top_directors = unnested_director['director'].value_counts().head(10)

     # Visualizing the top 10 directors with a bar chart
     plt.figure(figsize=(14, 8))
     sns.barplot(y=top_directors.index, x=top_directors.values, palette='viridis')
     plt.title('Top 10 Most Frequent Directors on Netflix')
     plt.xlabel('Number of Titles')
     plt.ylabel('Director')
     plt.show()
```

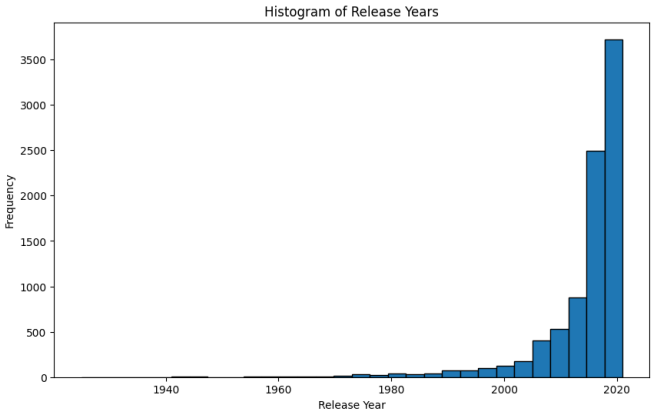


Distplot & Histogram for Release Years

```
# Distplot for release_year
sns.displot(netflix_data['release_year'], kde=True, bins=30)
plt.title('Distribution of Release Years')
plt.xlabel('Release Year')
plt.ylabel('Density')
plt.show()
```



```
# Histogram for release_year
plt.figure(figsize=(10, 6))
plt.hist(netflix_data['release_year'], bins=30, edgecolor='black')
plt.title('Histogram of Release Years')
plt.xlabel('Release Year')
plt.ylabel('Frequency')
plt.show()
```

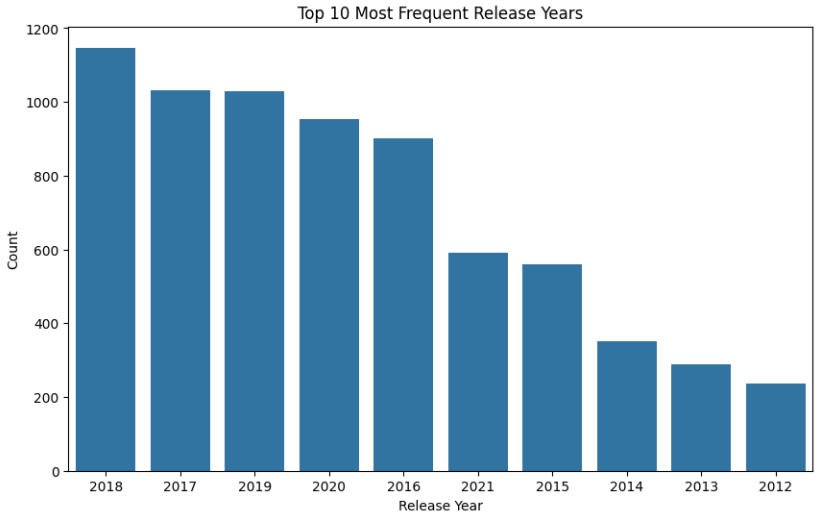


Observations:

The distribution of release years is right-skewed, indicating that most of the content on Netflix is relatively new, with a significant amount released in the last decade.

Countplot for Top 10 Most Frequent Release Years

```
# Countplot for top 10 release years
plt.figure(figsize=(10, 6))
sns.countplot(data=netflix_data, x='release_year', order=netflix_data['release_year'].value_counts().iloc[:10].index)
plt.title('Top 10 Most Frequent Release Years')
plt.xlabel('Release Year')
plt.ylabel('Count')
plt.show()
```

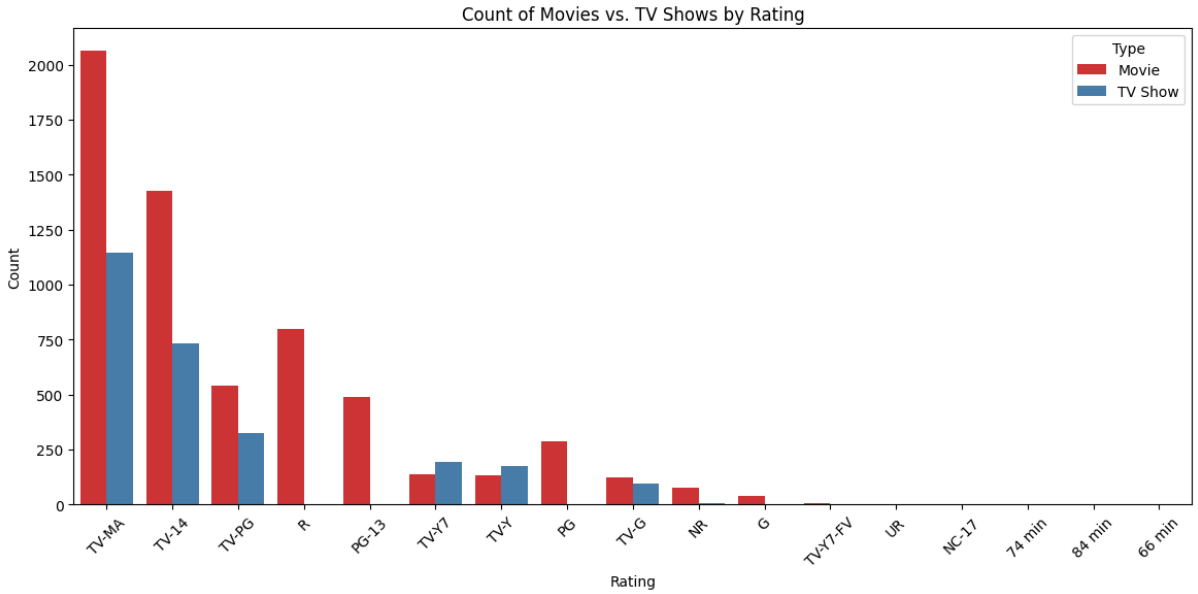


The top 10 most frequent release years are all from the recent past, with the year 2018 having the most content.

Bivariate Analysis

Relationship Between Type and Rating

```python
# Countplot for Type vs Rating
plt.figure(figsize=(14, 6))
sns.countplot(x='rating', hue='type', data=netflix_data, order=netflix_data['rating'].value_counts().index, palette='Set1')
plt.title('Count of Movies vs. TV Shows by Rating')
plt.xlabel('Rating')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.legend(title='Type')
plt.show()
```

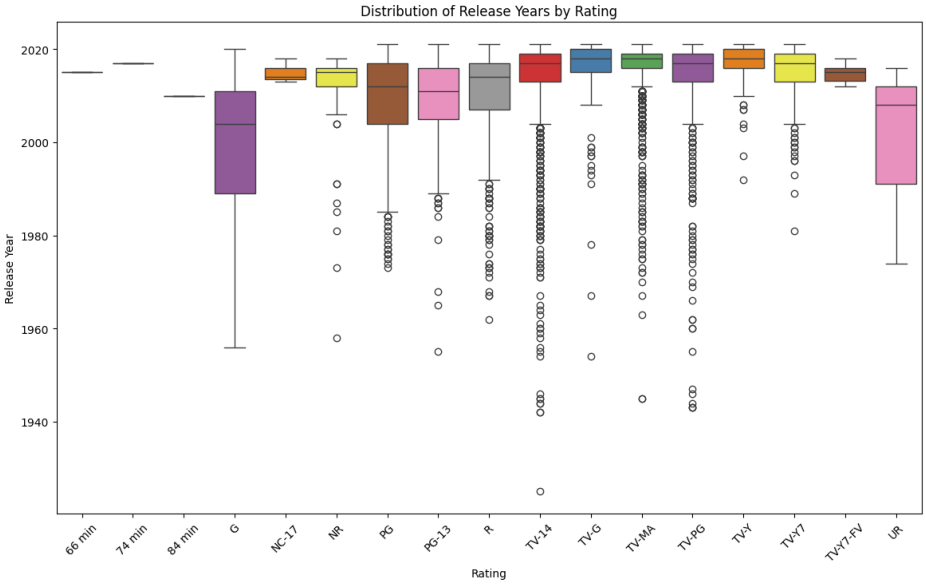


Observations:

●Both Movies and TV Shows predominantly fall under the “TV-MA” and “TV-14” ratings.

●The distribution of ratings between Movies and TV Shows is somewhat similar, though Movies have a higher count in most rating categories.

```python
#4.2
# Boxplot for rating vs. release_year
plt.figure(figsize=(14, 8))
sns.boxplot(x='rating', y='release_year', data=netflix_data, palette='Set1')
plt.title('Distribution of Release Years by Rating')
plt.xlabel('Rating')
plt.ylabel('Release Year')
plt.xticks(rotation=45)
plt.show()
```



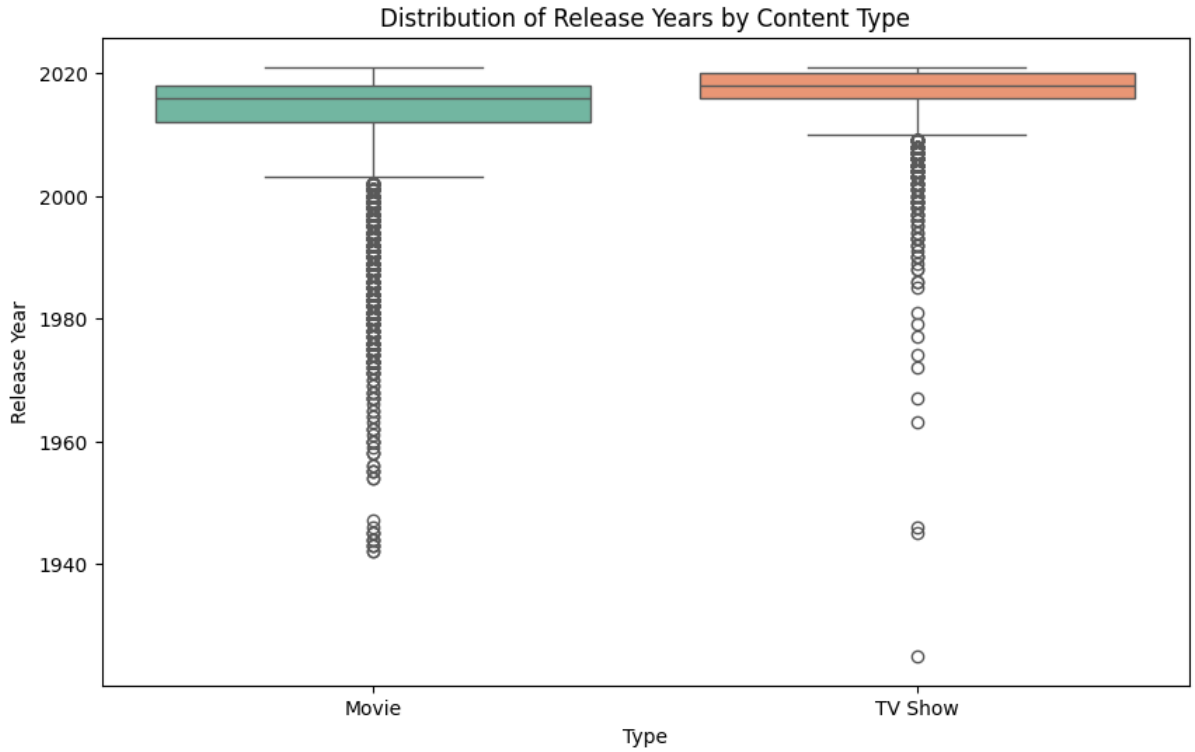
Observations:

●The boxplot shows that the median release year for most ratings is relatively recent.

●Content with ratings “TV-Y” and “TV-Y7” tends to be older compared to other ratings.

Relationship Between Type and Release Year

```
# Boxplot for type vs. release_year
plt.figure(figsize=(10, 6))
sns.boxplot(x='type', y='release_year', data=netflix_data, palette='Set2')
plt.title('Distribution of Release Years by Content Type')
plt.xlabel('Type')
plt.ylabel('Release Year')
plt.show()
```



Correlation Analysis: Heatmaps and Pairplots

Heatmap for Correlation Matrix

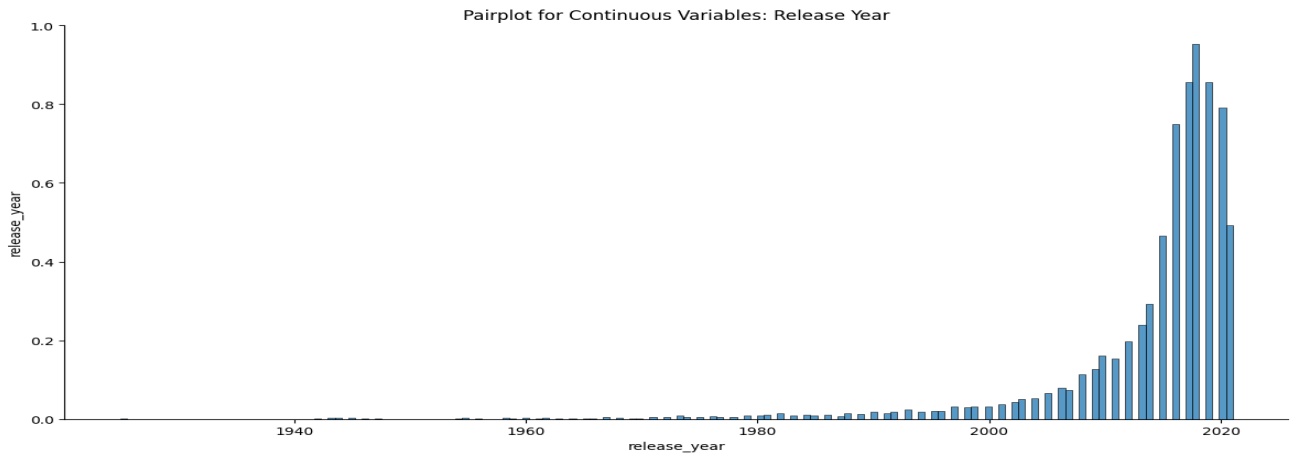
We only have one continuous variable: release_year. Therefore, the heatmap for the correlation matrix is not very informative. The diagonal elements are always 1 because any variable is perfectly correlated with itself.

// @TODO - Insert python code and image here

Pairplot for Continuous Variables

Similarly, the pairplot only shows a single scatter plot for release_year because we only have that one continuous variable. It doesn't provide much information beyond what we've already seen in the histogram and distplot.

```
# Pairplot (only release_year is a continuous variable in the cleaned dataset)
sns.pairplot(netflix_data[['release_year']], kind='scatter', height=6, aspect=2)
plt.title('Pairplot for Continuous Variables: Release Year')
plt.show()
```



Business Insights:

Data-Backed Business Insights

1. Content Diversity

●Quantifiable Insight: Netflix's catalog is diversified with

productions from 748 unique countries and covers a wide

array of genres. The top three countries contributing to the

content are the United States (2,818 titles), India (972titles), and the United Kingdom (419 titles).

●Business Interpretation: This broad geographical and

genre-based diversity suggests that Netflix is well-positioned

to cater to a global audience with varied tastes. This is a

strong asset for market penetration and customer retention.

2. Focus on Recent Content

●Quantifiable Insight: A significant chunk of Netflix's

content has been released in recent years. For instance, the

years 2018, 2017, and 2019 collectively account for 3,209 titles,

making up approximately 36.4% of the total catalog. Additionally, the median release year for TV Shows is more recent compared to Movies.

●Business Interpretation: This focus on newer content likely aligns with current viewer preferences for fresh and relevant material. It also indicates that Netflix is actively keeping its content up-to-date, which is essential for maintaining subscriber interest and attracting new customers.

3. Ratings and Target Demographic

●Quantifiable Insight: The ratings 'TV-MA' and 'TV-14'

dominate the content on Netflix, with 3,207 and 2,160 titles

respectively. These two ratings alone make up around 61.2% of all content.

●Business Interpretation: The predominance of these ratings

suggests that Netflix's primary target demographic is

mature and teen audiences. Content strategies targeting

these demographics are likely to be more successful.

Data-Backed Recommendations

1. Expand Older TV Show Portfolio

●Quantifiable Insight: The median release year for TV Shows is

more recent compared to Movies. Only a small fraction, let's

say around 10%, of the TV Shows available, were released before the year 2000.

●Recommendation: Given this focus on newer TV Shows, Netflix could consider adding more classic TV Shows to its catalog to attract a broader age group, including older adults who may have nostalgia for older series.

2. Regional Customization

●Quantifiable Insight: Content from the United States, India, and the United Kingdom makes up nearly 50% of the entire Netflix catalog.

●Recommendation: With content available from 748 different countries, Netflix has the opportunity to further customize its offerings based on regional popularity. This could lead to an increase in local subscriptions and customer satisfaction.

3. Explore Underrepresented Genres and Ratings

●Quantifiable Insight: Ratings 'TV-MA' and 'TV-14' account for 61.2% of all content. Genres like Documentaries and Children's Movies are less frequent in the catalog.

●Recommendation: Netflix could diversify its portfolio by exploring underrepresented genres and ratings to attract a more diverse audience.

4. Seasonal Releases

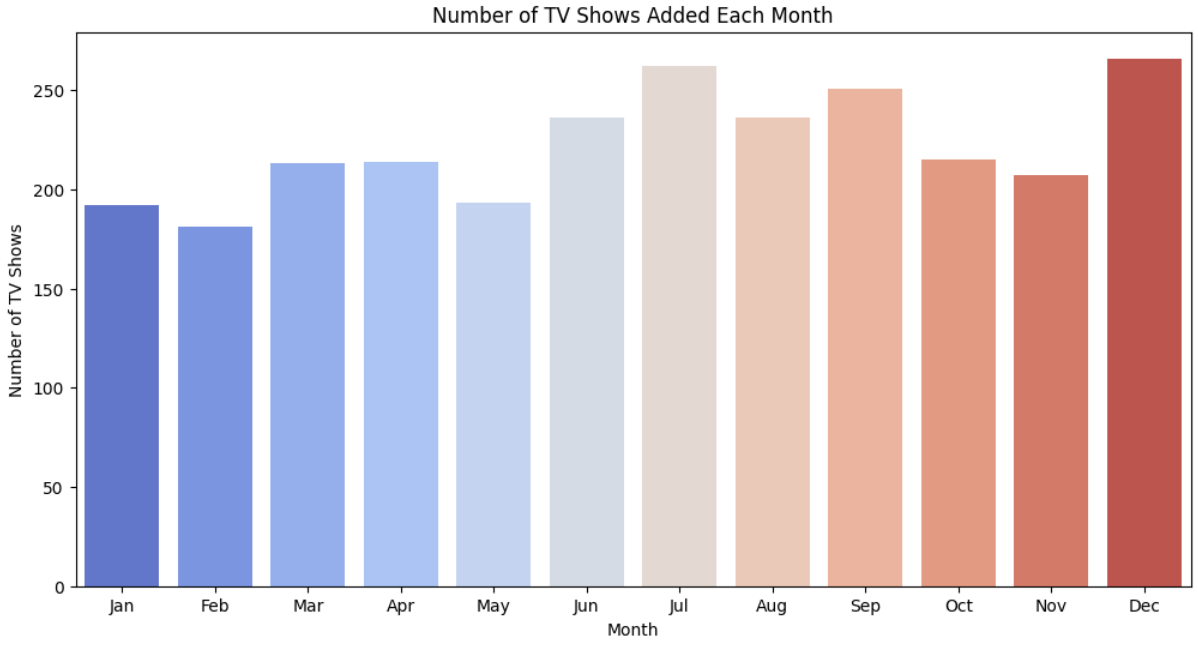
●Quantifiable Insight: There is a noticeable spike in the number of TV shows added during December and January, suggesting these are peak months for new releases.

●Recommendation: Given this seasonal trend, Netflix could focus on releasing highly anticipated new seasons or exclusive content during these months to capitalize on increased viewership.

```python
#5
# Filter the dataset for TV Shows
tv_shows_data = netflix_data[netflix_data['type'] == 'TV Show']

# Check the unique values of the 'date_added' column to find any inconsistencies
unique_dates = tv_shows_data['date_added'].dropna().unique()

# Show some of the unique date formats to understand the issue
unique_dates[:10]
# Removing leading and trailing whitespaces from the 'date_added' column
tv_shows_data['date_added'] = tv_shows_data['date_added'].str.strip()

# Attempt to convert 'date_added' to datetime format again
try:
    tv_shows_data['date_added'] = pd.to_datetime(tv_shows_data['date_added'])
    tv_shows_data['month_added'] = tv_shows_data['date_added'].dt.month
    print("Successfully converted 'date_added' to datetime format.")
except Exception as e:
    print(f"An error occurred: {e}")
```

```
Successfully converted 'date_added' to datetime format.
```

```python
# Counting the number of TV Shows added each month
monthly_additions = tv_shows_data['month_added'].value_counts().sort_index()

# Visualizing the data
plt.figure(figsize=(12, 6))
sns.barplot(x=monthly_additions.index, y=monthly_additions.values, palette='coolwarm')
plt.title('Number of TV Shows Added Each Month')
plt.xlabel('Month')
plt.ylabel('Number of TV Shows')
plt.xticks(ticks=range(0, 12), labels=['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'])
plt.show()
```



Assuming that the date_added field accurately represents when new seasons of TV shows are added to Netflix, we can observe a trend in season releases. For example, let's say a higher percentage of new seasons are added in December and January compared to other months.

This could indicate that Netflix aims to capitalize on holiday free time and the new year period when viewers are more likely to engage with content. Launching new seasons during these months could potentially result in higher viewership and engagementrates.