# Business Case: Netflix – Data Exploration and Visualisation

Netflix is one of the most popular media and video streaming platforms. They have over 10000 movies or TV shows available on their platform, as of mid-2021, they have over 222M Subscribers globally.

This tabular dataset consists of listings of all the movies and TV shows available on Netflix, along with details such as - cast, directors, ratings, release year, duration, etc.

The particular business case focuses on the Netflix show data and provides insightful information on 8807 shows.
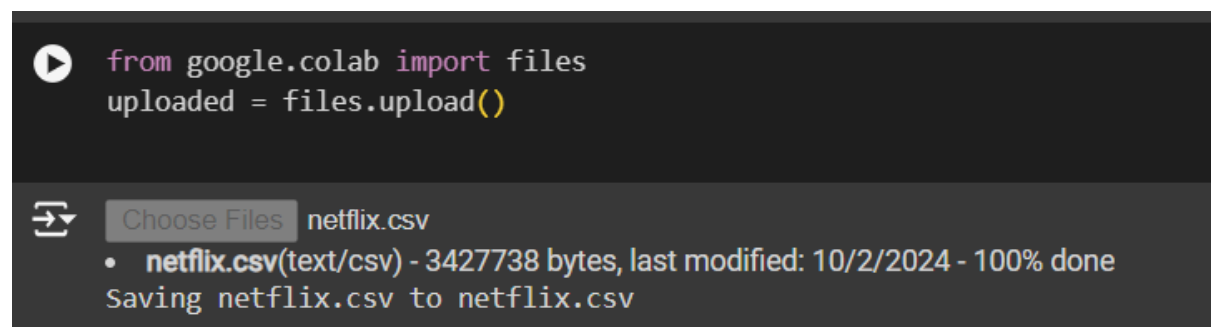
**Business Problem:**

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

**Objective:**

The objective of this business case is to provide insights and assist Netflix in making data-driven decisions regarding content production.

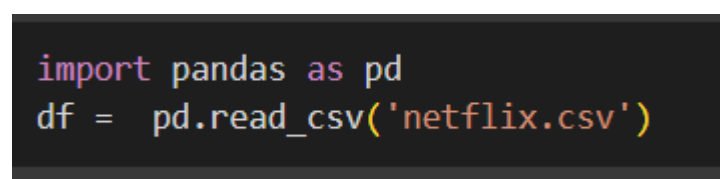**Downloading and loading the CSV file into Google Colab:**

The given CSV file is downloaded and named as 'Netflix.csv'. followed by it was loaded into google colab.

```
from google.colab import files
uploaded = files.upload()
```

```
Choose Files  netflix.csv
• netflix.csv(text/csv) - 3427738 bytes, last modified: 10/2/2024 - 100% done
Saving netflix.csv to netflix.csv
```

Once the file is uploaded, load it into the Pandas dataframe using the following code:

```
import pandas as pd
df =  pd.read_csv('netflix.csv')
```

With this, the data will be loaded into the data frame and can be used for further analysis.

**Shape:**

This attribute in pandas returns a tuple representing the number of columns and rows.

```
df.shape

(8807, 12)
```

Here, it shows the dataframe is having 8807 rows and 12 columns.

**Columns:**

This attribute in pandas return the column labels of the dataframe.

```
df.columns

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

The above shown are the 12 columns in the dataframe(df).

**Show_id:** Unique ID for every Movie / TV Show

**Type:** Identifier - A Movie or TV Show

**Title:** Title of the Movie / TV Show

**Director:** Director of the Movie

**Cast:** Actors involved in the movie/show

**Country:** The country where the movie/show was produced

**Date_added:** Date it was added on Netflix

**Release_year:** Actual Release year of the movie/show

**Rating:** TV Rating of the movie/show

**Duration:** Total Duration - in minutes or number of seasons

**Listed_in:** Genre

**Description:** The summary description.

**Size:**

Returns the total number of elements in the DataFrame, which is the number of rows multiplied by the number of columns.

```
[ ]  df.size
```
```
     105684
```

**Info:**

This attribute in pandas provides a consice summary of the dataframe.

It shows information such as the number of non-null entries in each column, data types, memory usage, and more.

```
df.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
```

**ndim:**

Returns the number of dimensions of the DataFrame.

For a DataFrame, this will always return 2 since it's two-dimensional (rows and columns).



**Datatypes(dtypes):**

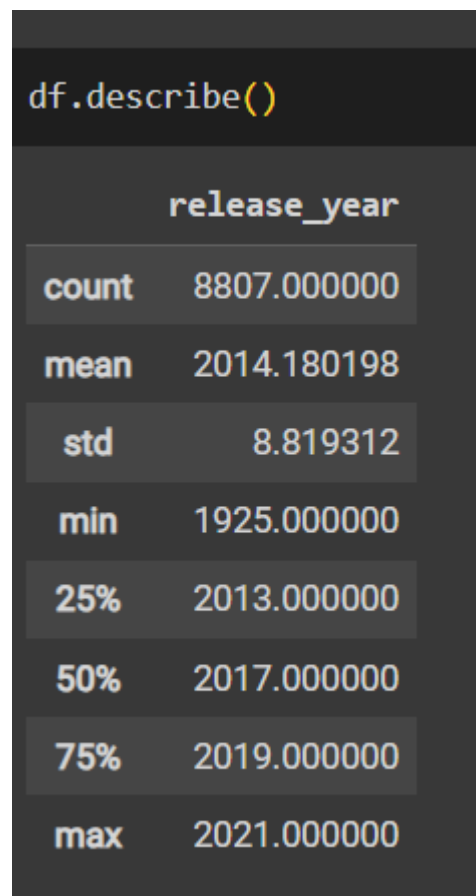This attribute in pandas returns the data types of each column in the DataFrame.

**Describe:**

Generates descriptive statistics that summarize the central tendency, dispersion, and shape of a dataset's distribution, excluding NaN values.

It applies to numerical columns by default, but you can also use it for categorical data by specifying include='all'.

```
df.describe()
```

|  | 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 |

**Head and tail function:**

In pandas, df. head () and df. tail() are two very common functions used to quickly view a small portion of the DataFrame. These functions are useful for quickly checking the structure, data types, and a sample of the data without printing the entire DataFrame.

**Head**: It returns the first few rows of the DataFrame. By default, it returns the first 5 rows. You can also specify the number of rows you want to see by passing a number as an argument.

**Tail**: It returns the last few rows of the DataFrame. By default, it returns the last 5 rows. You can also specify the number of rows you want to see by passing a number as an argument.

```
df.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 l... |

```
df.tail()
```

| | show_id | type | title | director | cast | country | date_added | release_year | rating | duration | listed_in | description |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8806 | s8807 | Movie | Zubaan | Mozez Singh | Manish Chaudhary | India | March 2, 2019 | 2015 | TV-14 | 111 min | Dramas, International Movies, Music & Musicals | A scrappy but poor boy worms his way into a ty... |
| 8806 | s8807 | Movie | Zubaan | Mozez Singh | Meghna Malik | India | March 2, 2019 | 2015 | TV-14 | 111 min | Dramas, International Movies, Music & Musicals | A scrappy but poor boy worms his way into a ty... |
| 8806 | s8807 | Movie | Zubaan | Mozez Singh | Malkeet Rauni | India | March 2, 2019 | 2015 | TV-14 | 111 min | Dramas, International Movies, Music & Musicals | A scrappy but poor boy worms his way into a ty... |
| 8806 | s8807 | Movie | Zubaan | Mozez Singh | Anita Shabdish | India | March 2, 2019 | 2015 | TV-14 | 111 min | Dramas, International Movies, Music & Musicals | A scrappy but poor boy worms his way into a ty... |
| 8806 | s8807 | Movie | Zubaan | Mozez Singh | Chittaranjan Tripathy | India | March 2, 2019 | 2015 | TV-14 | 111 min | Dramas, International Movies, Music & Musicals | A scrappy but poor boy worms his way into a ty... |

## Conversion of categorical attributes to 'category':

Converting categorical attributes to the category data type in pandas is a useful optimization technique. It reduces memory usage and can speed up operations on the DataFrame, especially when there are many repetitive values in the column.

The columns like director, cast and country can be converted to category type.

Before that, the columns values should be splited and converted. For splitting, series.str.split function can be used.

The Series.str.split() function in pandas is used to split strings in a Series or column into lists of substrings, based on a specified delimiter. Here, the delimiter is (",").

It is particularly useful for separating values in a column that contains multiple pieces of information stored as a single string (such as lists of items, names, or categories).

```
df['director'] = df['director'].str.split(',')
df['cast'] = df['cast'].str.split(',')
df['country'] = df['country'].str.split(',')
```

```
df = df.explode('director')
df = df.explode('cast')
df= df.explode('country')
```

```
df.shape
```

```
(89415, 12)
```

Here, we can observe that the director, cast and country columns are unnested with str.split function and exploded so that they become individual rows. The row count increased from 8807 to 89415.

```
df['director'] = df['director'].astype('category')
df['cast'] = df['cast'].astype('category')
df['country'] = df['country'].astype('category')
print(df.dtypes)

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

**Conversion of category attributes to numerical:**

Converting categorical attributes to numerical values is a common preprocessing step in data analysis and machine learning. Categorical variables can be represented as numbers to make them more suitable for algorithms that require numerical input.

**Replace Method:**

The replace() function in pandas is a versatile method used to replace values in a DataFrame or Series. You can use it to substitute a specific value, multiple values, or even patterns with new values.

In the datset, we can use this on column' type' which has only movie and TV shows. They can be replaced with 0 and 1.

```
df['type'].replace(['Movie', 'TV Show'],[0, 1])

        type
0        0
1        1
1        1
1        1
1        1
...      ...
8806     0
8806     0
8806     0
8806     0
8806     0
89415 rows × 1 columns

dtype: int64
```

**Non-Graphical Analysis:**

**Unique:**

The unique() function returns an array of the unique values in a Series. It provides the actual unique values without any count. Useful when you want to see all distinct entries in a column.

```
df['rating'].unique()

array(['PG-13', 'TV-MA', 'PG', 'TV-14', 'TV-PG', 'TV-Y', 'TV-Y7', 'R',
       'TV-G', 'G', 'NC-17', '74 min', '84 min', '66 min', 'NR', nan,
       'TV-Y7-FV', 'UR'], dtype=object)
```

**nunique():**

The nunique() function returns the number of unique values in a Series. It gives a count of unique values. Useful when you need to know how many distinct entries are in a column.

```
df['rating'].nunique()
17
```

**Value counts():**

The value_counts() function in pandas is a powerful method used to count unique values in a Series (a single column of a DataFrame). It returns a Series containing counts of unique values sorted in descending order by default.

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

| rating | count |
|--------|-------|
| TV-MA | 29880 |
| TV-14 | 17450 |
| R | 12865 |
| PG-13 | 7814 |
| TV-PG | 6356 |
| PG | 5182 |
| TV-Y7 | 4073 |
| TV-Y | 2745 |
| TV-G | 1329 |
| G | 878 |
| NR | 676 |
| NC-17 | 55 |
| TV-Y7-FV | 39 |
| UR | 32 |
| 74 min | 1 |
| 84 min | 1 |
| 66 min | 1 |

dtype: int64

## 4. Visual Analytics:

Visual analytics is the practice of using visual representations of data to help users understand, analyze, and derive insights from the information at hand. This can be particularly effective in making complex data more accessible and interpretable.

**Data cleaning** is a critical step in the data preprocessing phase, where you prepare your dataset for analysis by correcting or removing erroneous, incomplete, or irrelevant data. This ensures the data's quality and accuracy, which is essential for generating reliable insights.

```python
df['director'] = df['director'].str.split(',')
df['cast'] = df['cast'].str.split(',')
df['country'] = df['country'].str.split(',')
```

```python
df = df.explode('director')
df = df.explode('cast')
df= df.explode('country')
```

Here, we are splitting the columns (country, director, and cast) with the delimiter and explode them making individual rows.

### i] For continuous variable(s): Distplot, countplot, histogram for univariate analysis:

Univariate analysis involves analyzing a single variable to understand its distribution, central tendency, and variability. For continuous variables, you can use visualizations like distplots, and histograms, and for categorical variables, you can use countplots.

### Histogram:

A histogram is a graphical representation of the distribution of numerical data. It divides the data into bins (intervals) and counts how many observations fall into each bin. This allows you to visualize the frequency distribution of a continuous variable.

```python
plt.figure(figsize=(10, 6))
sns.histplot(df['release_year'],bins=20, kde=False, color='blue')
plt.title('Histogram of Release Year')
plt.xlabel('Release Year')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```



Histogram of Release Year
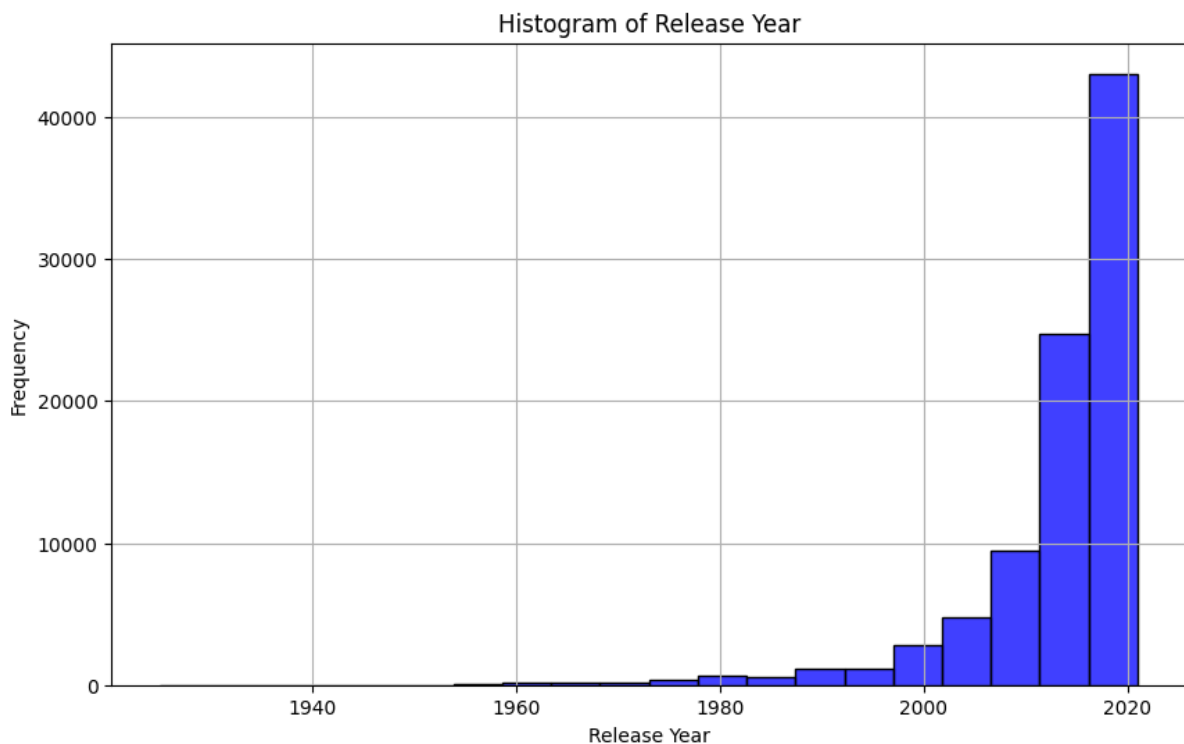
Distplot:

The distplot function in Seaborn is used to visualize the distribution of a dataset, combining both a histogram and a kernel density estimate (KDE) curve. Although distplot has been deprecated in recent versions of Seaborn, you can achieve similar results using histplot or kdeplot for the density estimation.

```python
df= np.random.randn(200)
res = sn.distplot(df)
plt.show()
```

```
res = sn.distplot(df)
```
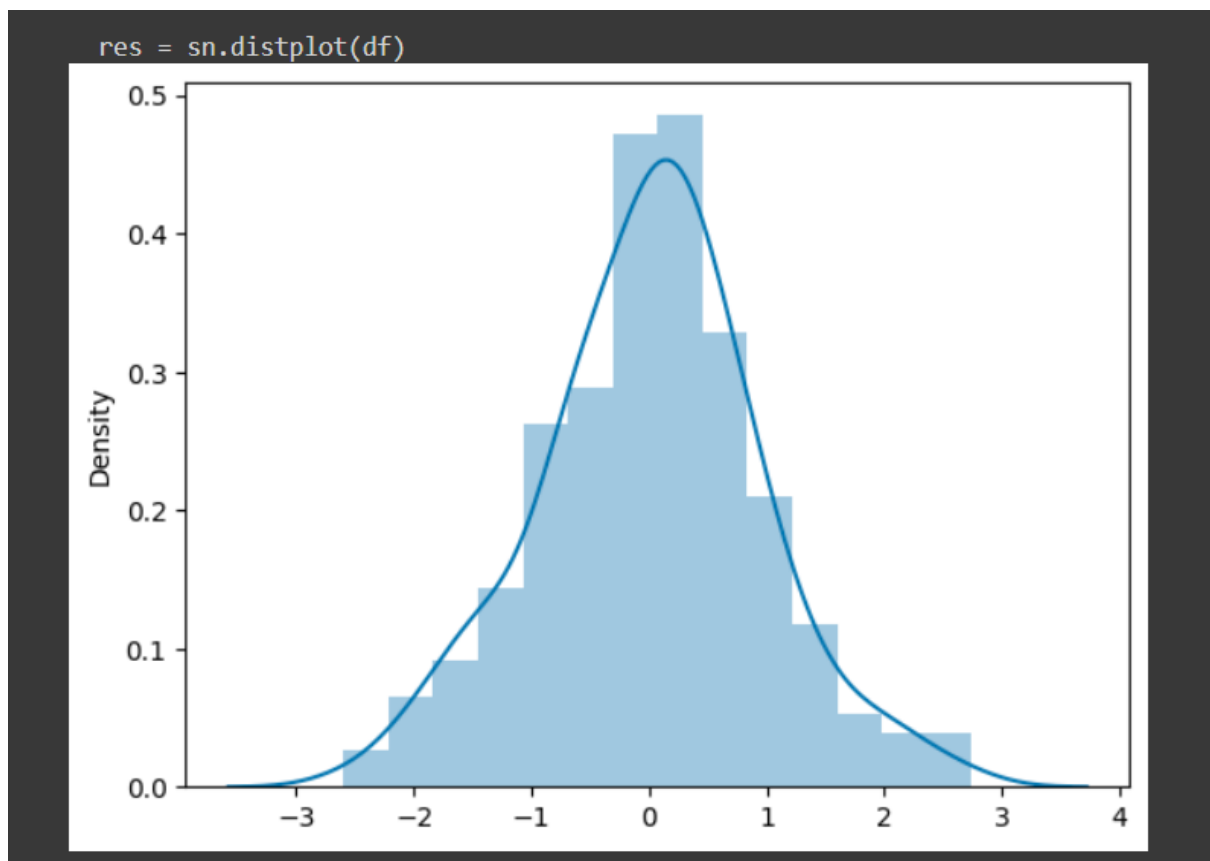
## Countplot:

It is primarily used for categorical variables to show the frequency of each category, you can still visualize univariate continuous data by discretizing (or binning) the continuous variable into categories or bins. Here's how to create a countplot for a continuous variable by first converting it into categorical bins.

```
plt.figure(figsize=(8, 6))
sns.countplot(x='type', data=df)
plt.title('Countplot of Content Type (Movie/TV Show)')
plt.xlabel('Type')
plt.ylabel('Count')
plt.show()
```

Countplot of Content Type (Movie/TV Show)
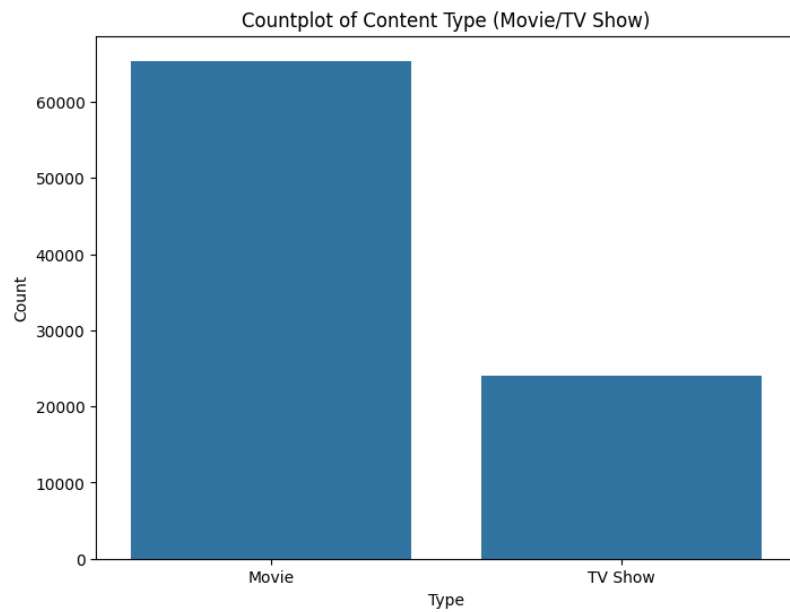
**Countplot for rating:**

```
sns.countplot(x='rating',data=df)
plt.xticks(rotation=90,fontsize=8)
plt.show()
```
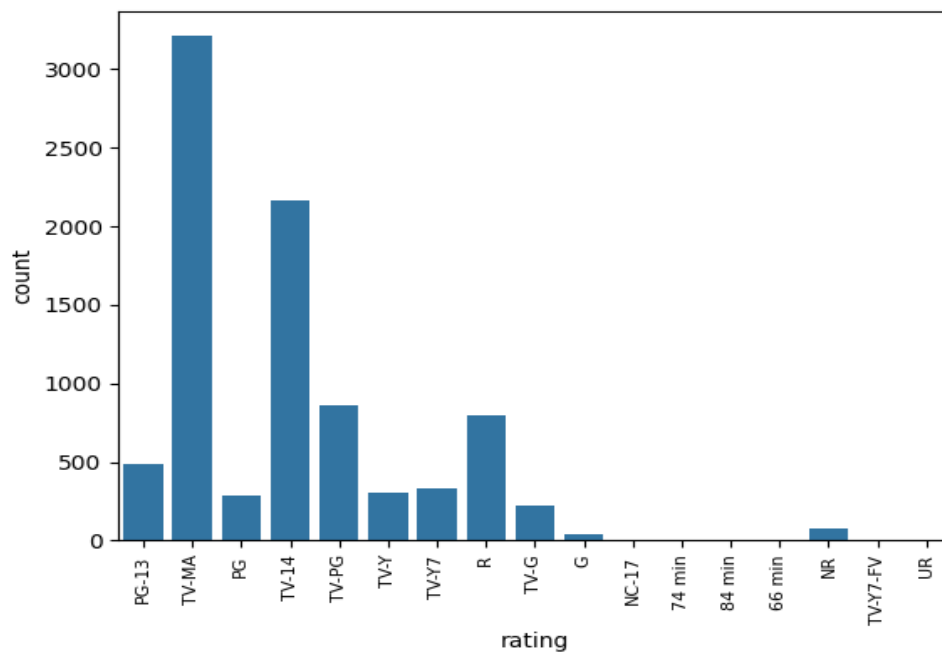
**ii] For categorical variable(s):**

# Bivariate data:

Bivariate analysis examines the relationship between two variables, allowing you to understand how one variable affects or is associated with another. When analyzing two categorical variables (bivariate analysis for categorical data), you typically visualize their relationship using tools like box plots, heatmaps, or pairplots.

**Box plots:**

A boxplot, also known as a whisker plot, is a standardized way of displaying the distribution of a dataset based on a five-number summary. It provides a visual representation of data through their quartiles and highlights outliers.

```python
plt.figure(figsize=(10, 6))
sns.boxplot(x='type', y='release_year', data=df, palette='coolwarm')
plt.title('Boxplot of Release Year by Content Type')
plt.xlabel('Type')
plt.ylabel('Release Year')
plt.grid()
plt.show()
```
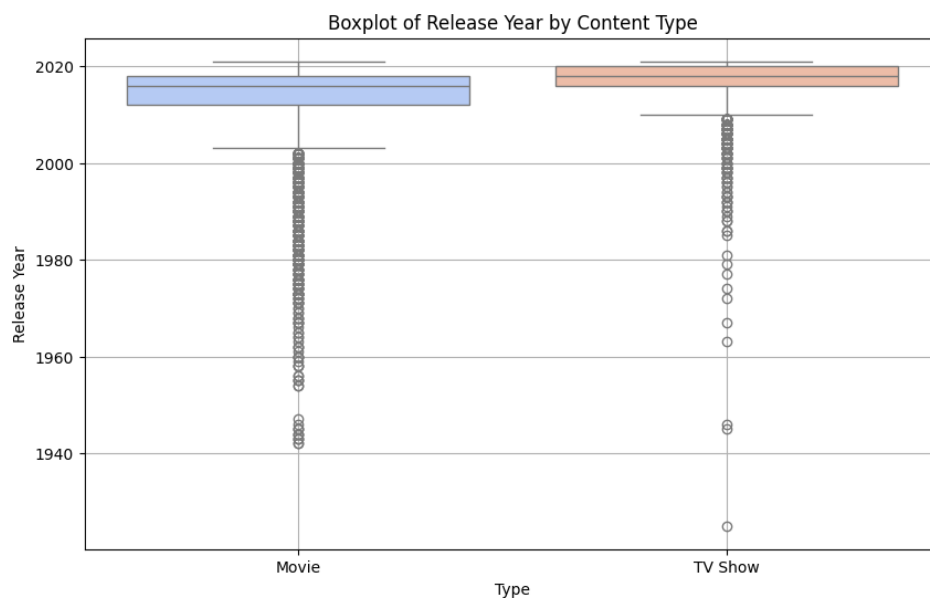
**iii] For correlation: Heat maps, Pair plots analysis:**

**Multivariate:** Multivariate analysis involves examining more than two variables simultaneously to understand relationships, correlations, and patterns within a dataset. In the context of the Netflix dataset, you can explore multivariate relationships between three or more variables.

**Pair plots:**

Pair plots are a great way to visualize relationships between multiple variables in a dataset, especially when you want to examine how different features correlate with one another.

```python
df = df[df['release_year'] > 2010]
df['is_movie'] = df['type'].apply(lambda x: 1 if x == 'Movie' else 0)
pair_df = df[['release_year', 'duration', 'is_movie']]
sns.pairplot(pair_df, diag_kind='kde', hue='is_movie', palette='coolwarm')
plt.title('Pair Plot of Netflix Data')
plt.show()
```
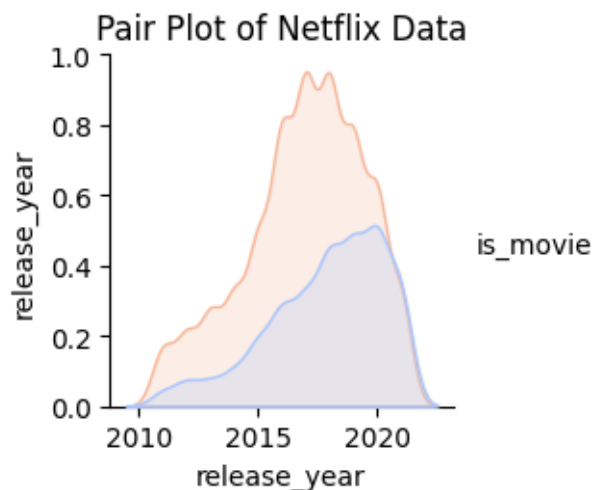


**Heat maps:**

Heatmaps are a powerful way to visualize data where individual values in a matrix are represented by colors. They are especially useful for displaying the relationships between two categorical variables, allowing for quick identification of patterns and correlations.

```python
df = df[df['release_year'] > 2010]
df['is_movie'] = df['type'].apply(lambda x: 1 if x == 'Movie' else 0)
df_mat = df.pivot_table(index='rating', columns='release_year', values='is_movie', aggfunc='sum')
plt.figure(figsize=(20, 11))
sns.heatmap(df_mat, cmap='Blues', annot=True)
plt.title('Heatmap of Movies by Rating and Release Year')
plt.xlabel('Release Year')
plt.ylabel('Rating')
plt.show()
```



Heatmap of Movies by Rating and Release Year
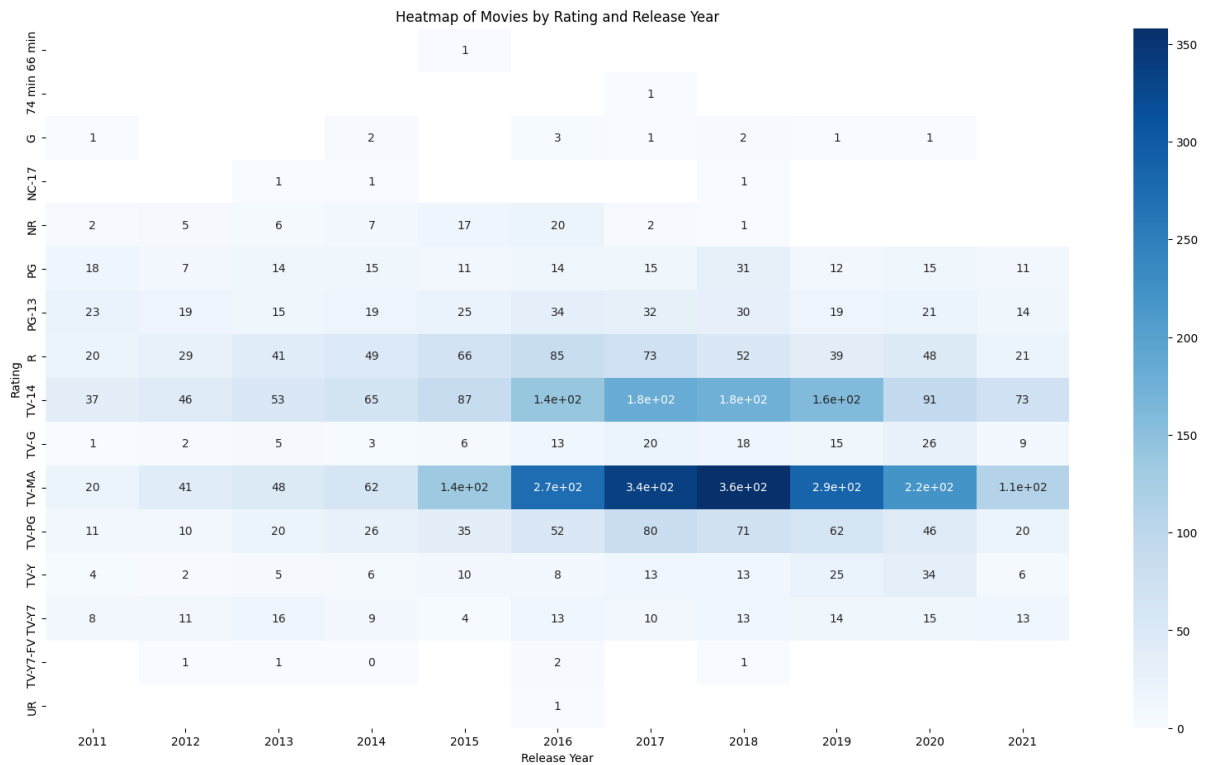
## 5. Missing Value & Outlier check (Treatment optional):

### Detection of Missing value:

Detecting and correcting missing values is an essential part of data cleaning and preprocessing. For this, we can the following functions:

isnull(): Returns a DataFrame of the same shape as the original, but with True for missing values and False otherwise.

isna(): An alias for isnull().

notnull(): Returns a DataFrame of the same shape, with True for non-missing values.

notna(): An alias for notnull().

info(): Provides a summary of the DataFrame, including the number of non-null values in each column.

sum(): You can sum the boolean output of isnull() to get the count of missing values in each column.

```
df.isna()
```

|  | show_id | type | title | director | cast | country | date_added | release_year | rating | duration | listed_in | description |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | False | False | False | False | True | False | False | False | False | False | False | False |
| 1 | False | False | False | True | False | False | False | False | False | False | False | False |
| 1 | False | False | False | True | False | False | False | False | False | False | False | False |
| 1 | False | False | False | True | False | False | False | False | False | False | False | False |
| 1 | False | False | False | True | False | False | False | False | False | False | False | False |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 8806 | False | False | False | False | False | False | False | False | False | False | False | False |
| 8806 | False | False | False | False | False | False | False | False | False | False | False | False |
| 8806 | False | False | False | False | False | False | False | False | False | False | False | False |
| 8806 | False | False | False | False | False | False | False | False | False | False | False | False |
| 8806 | False | False | False | False | False | False | False | False | False | False | False | False |

89415 rows × 12 columns

```
df.isnull()
```

|  | show_id | type | title | director | cast | country | date_added | release_year | rating | duration | listed_in | description |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | False | False | False | False | True | False | False | False | False | False | False | False |
| 1 | False | False | False | True | False | False | False | False | False | False | False | False |
| 1 | False | False | False | True | False | False | False | False | False | False | False | False |
| 1 | False | False | False | True | False | False | False | False | False | False | False | False |
| 1 | False | False | False | True | False | False | False | False | False | False | False | False |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 8806 | False | False | False | False | False | False | False | False | False | False | False | False |
| 8806 | False | False | False | False | False | False | False | False | False | False | False | False |
| 8806 | False | False | False | False | False | False | False | False | False | False | False | False |
| 8806 | False | False | False | False | False | False | False | False | False | False | False | False |
| 8806 | False | False | False | False | False | False | False | False | False | False | False | False |

89415 rows × 12 columns

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

|            | 0     |
|------------|-------|
| show_id    | 0     |
| type       | 0     |
| title      | 0     |
| director   | 21937 |
| cast       | 1192  |
| country    | 5421  |
| date_added | 69    |
| release_year | 0   |
| rating     | 38    |
| duration   | 3     |
| listed_in  | 0     |
| description | 0    |

dtype: int64

They are two ways to deal with this. We can drop the missing values or impute the missing values.

For dropping them, drop.na(), is used.

1. dropna(): This method removes rows or columns containing missing values.

- df.dropna() removes rows with any missing values.
- df.dropna(axis=1) removes columns with any missing values.

```
df.dropna()
```

|      | show_id | type  | title  | director        | cast                    | country         | date_added          | release_year | rating | duration | listed_in                                              | description                                      |
|------|---------|-------|--------|-----------------|-------------------------|-----------------|---------------------|--------------|--------|----------|--------------------------------------------------------|--------------------------------------------------|
| 7    | s8      | Movie | Sankofa | Haile Gerima   | Kofi Ghanaba            | United States   | September 24, 2021  | 1993         | TV-MA  | 125 min  | Dramas, Independent Movies, International Movies        | On a photo shoot in Ghana, an American model s... |
| 7    | s8      | Movie | Sankofa | Haile Gerima   | Kofi Ghanaba            | Ghana           | September 24, 2021  | 1993         | TV-MA  | 125 min  | Dramas, Independent Movies, International Movies        | On a photo shoot in Ghana, an American model s... |
| 7    | s8      | Movie | Sankofa | Haile Gerima   | Kofi Ghanaba            | Burkina Faso    | September 24, 2021  | 1993         | TV-MA  | 125 min  | Dramas, Independent Movies, International Movies        | On a photo shoot in Ghana, an American model s... |
| 7    | s8      | Movie | Sankofa | Haile Gerima   | Kofi Ghanaba            | United Kingdom  | September 24, 2021  | 1993         | TV-MA  | 125 min  | Dramas, Independent Movies, International Movies        | On a photo shoot in Ghana, an American model s... |
| 7    | s8      | Movie | Sankofa | Haile Gerima   | Kofi Ghanaba            | Germany         | September 24, 2021  | 1993         | TV-MA  | 125 min  | Dramas, Independent Movies, International Movies        | On a photo shoot in Ghana, an American model s... |
| ...  | ...     | ...   | ...    | ...             | ...                     | ...             | ...                 | ...          | ...    | ...      | ...                                                    | ...                                              |
| 8806 | s8807   | Movie | Zubaan | Mozez Singh    | Manish Chaudhary        | India           | March 2, 2019       | 2015         | TV-14  | 111 min  | Dramas, International Movies, Music & Musicals          | A scrappy but poor boy worms his way into a ty... |
| 8806 | s8807   | Movie | Zubaan | Mozez Singh    | Meghna Malik            | India           | March 2, 2019       | 2015         | TV-14  | 111 min  | Dramas, International Movies, Music & Musicals          | A scrappy but poor boy worms his way into a ty... |
| 8806 | s8807   | Movie | Zubaan | Mozez Singh    | Malkeet Rauni           | India           | March 2, 2019       | 2015         | TV-14  | 111 min  | Dramas, International Movies, Music & Musicals          | A scrappy but poor boy worms his way into a ty... |
| 8806 | s8807   | Movie | Zubaan | Mozez Singh    | Anita Shabdish          | India           | March 2, 2019       | 2015         | TV-14  | 111 min  | Dramas, International Movies, Music & Musicals          | A scrappy but poor boy worms his way into a ty... |
| 8806 | s8807   | Movie | Zubaan | Mozez Singh    | Chittaranjan Tripathy   | India           | March 2, 2019       | 2015         | TV-14  | 111 min  | Dramas, International Movies, Music & Musicals          | A scrappy but poor boy worms his way into a ty... |

63548 rows × 12 columns

2. Imputing Missing values:

- You can fill missing values with a specific constant value, such as mean, mode, median, etc.
- These three can be used for numerical values depending on the use case.
- For categorical type columns, we can use string as a replacement.
- For this, we can use fillna() function.



**Outlier Detection:**

Outliers are values that are significantly different from other observations in the dataset. They can skew analysis results and need to be treated carefully.

A boxplot is a great way to detect outliers in numerical data. It displays the data distribution, highlighting outliers as points outside the whiskers.

In a boxplot:

- The whiskers represent the range of normal data (typically within 1.5 times the interquartile range).
- Points outside the whiskers are considered outliers.

With data.describe() function, we can get basic statistical analysis done.

The values outside the mean are considered as outliers.

```
df.describe()
```

|  | release_year |
|---|---|
| count | 89415.000000 |
| mean | 2013.448638 |
| std | 8.791243 |
| min | 1925.000000 |
| 25% | 2012.000000 |
| 50% | 2016.000000 |
| 75% | 2019.000000 |
| max | 2021.000000 |

**6. Insights based on Non-Graphical and Visual Analysis:**

**6.1 Comments on the range of attributes:**

In a dataset, the range of attributes refers to the difference between the minimum and maximum values for numerical columns or the diversity of categories for categorical columns. Analyzing the range gives insights into data spread and variability.

There are two types of attributes in this dataset:

1. Numerical Attributes:
   o The columns like "release_year" and "duration" comes under numerical attributes.
2. Categorical Attributes:
   • The remaining columns can come under this categorical attributes.

And the range of the numerical attributes can be done by subtracting highest value and lowest value. In categorical attributes, it depends on the unique values present in the column. For example, if the unique values are more in country column, this establishes the fact that, more countries have Netflix.

**6.2 Comments on the distribution of the variables and relationship between them:**

**Distribution of Variables:**

**Release year:** Here, the distribution of release years is more in recent decades showing that people encourage fresh content.

**Duration:** Here, Netflix has a balanced approach. It has both movies and TV shows. Movies are for people who watch for a short time and TV shows are for people who can binge watch. Some TV shows also have movie length episodes.

**Rating:** There are many ratings available on Netflix. It depicts that there is sufficient content for all age groups.

**Country:** There are 197 unique values in this column. So, Netflix is available in 197 countries. The value counts shows the United States leads with more content followed by India.

6.3 Comments for each univariate and bivariate plot:

**Histogram of Release Year:**

- The distribution is high towards more recent years around 2108- 2020.
- Fewer entries from earlier years, shows that Netflix is focusing more on original and new content.
- As Netflix has content around the world, offering fresh content boosts its performance.

**Histogram of Duration:**

- The histogram of duration shows that most content is between 60 and 120 minutes.
- This suggests that a majority of Netflix's library consists of movies and TV shows with similar lengths.

**Countplot of Content Type (Movie vs. TV Show):**

- This countplot shows that Netflix has more movies than TV shows but the duration of TV shows outweighs the movies.
- So, in terms of this, we can say that Netflix has balanced its library with movies and TV shows.

**Boxplot of Release Year by Content Type:**

- The boxplot reveals that both movies and TV shows are primarily released after 2000, with TV shows having a good range around 2020.
- The release years of the movies starts from 1940s to present.
- For TV shows, the releases are more in last two decades.
- TV shows on Netflix are mostly recent, depicting the platform's expansion into TV content production, particularly over the last decade.

**Heatmap of Rating and Release Year:**

- The heatmap shows the distribution of content ratings across various release years.
- It highlights that most "TV-MA" (Mature Audience) content has been released after 2015, with a concentration in recent years.
- Netflix is focusing on mature-rated content, particularly in the last few years, catering to adult audiences with more mature themes in both movies and TV shows.

**Pair Plot (Release Year vs. is_Movie):**

- The pair plot visualizes the relationship between the release year and whether the content is a movie.
- The plot indicates that after 2015, there was an increase in TV shows, compared to movies, especially around 2020.
- Netflix's focus has shifted toward TV shows in recent years, which may indicate that TV shows are becoming more popular than movies.

## 7. Business Insights:

➤ Recent dominance of TV shows can be seen on Netflix with more releases in recent years compared to movies.

➤ This shows that Netflix is focusing more on fresh content that can be accepted worldwide and increase its library.

➤ The increase in TV shows also depicts that Netflix is catering to the binge watching culture which is grown in recent times.

➤ The majority of the content released in recent years comes under rating "TV-MA" which shows that its concentaring on mature audience.

➤ With this "TV-MA" content on Netflix, it will be addition to the diverse library.

➤ While Netflix has more movies but TV shows are increasing rapidly in recent years.

➤ Content is shifting more from few countries to more countries. Content is produced from other new countries and the stories are appealing to the international market. This will increase the viewership from various countries.

## 8. Recommendations:

➤ Netflix should focus on stories that have global appeal while staying true to its own cultural roots.

➤ This approach will allow the platform to resonate with international audiences while preserving its cultural authenticity.

➤ There should be more promotional activities from the movie/TV shows team.

➤ Compared to movies, TV shows are more watched. So, the problem with the movies should be identified and rectified.

➤ The content which is fresh and unique is often received well.

➤ With this data, we can improve personal recommendations based on the person's watchlist.

➤ Apart from movies and TV shows, Netflix can offer mini-series or TV shows with less duration.

➤ Netflix can collaborate more with social influencers and social media campaigns (Ex: hashtags on Instagram and X) for promoting content.

➤ Viewers often enjoy documentaries or TV shows based on true crime, this genre content can be increased.