About:

Netflix is a global streaming service that offers a wide variety of TV shows, movies, documentaries, and original content across a range of genres and languages. Launched in 1997 as a DVD rental service, it revolutionized the entertainment industry by introducing the concept of streaming in 2007. With over 200 million subscribers worldwide, Netflix has become synonymous with binge-watching culture and is known for its innovative approach to content creation and distribution.

Business Case Study:

- **Market Expansion Strategy:** As **Netflix** continues to expand its global footprint, understanding regional preferences becomes paramount. With over 200 countries in its reach, tailoring content to specific demographics and cultural nuances is essential for sustained growth.
- **Competitive Landscape:** In a competitive streaming landscape, characterized by the emergence of new players and the evolving strategies of existing ones, **Netflix** must differentiate itself by offering compelling and diverse content. This necessitates a deep dive into audience preferences and consumption patterns across various geographies.
- **Content Personalization:** With the proliferation of streaming platforms, viewers have come to expect personalized recommendations and content curation. Leveraging data insights, **Netflix** can refine its recommendation algorithms to not only suggest relevant content but also anticipate emerging trends, thereby enhancing user engagement and retention.
- **Risk Mitigation:** Producing original content entails significant financial investment and risk. By leveraging data analytics to forecast audience demand and gauge the potential success of different genres and formats, **Netflix** can mitigate risk and optimize its content investment strategy, ensuring a higher return on investment.
- **Cultural Relevance:** The success of a show or movie often hinges on its cultural relevance and resonance with the target audience. Understanding cultural sensitivities, societal trends, and local preferences is crucial for creating content that resonates with viewers across different regions, fostering a deeper connection and loyalty to the platform.

Content:

1. Defining Problem Statement and Analyzing basic metrics.
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.
3. Non-Graphical Analysis: Value counts and unique attributes.
4. Visual Analysis - Univariate, Bivariate after pre-processing of the data.Pre-processing involves unnesting of the data in columns like Actor, Director, Country.
    - 4.1. For continuous variable(s): Distplot, countplot, histogram for univariate analysis.
    - 4.2. For categorical variable(s): Boxplot.
    - 4.3. For correlation: Heatmaps, Pairplots.
5. Missing Value & Outlier check (Treatment optional).
6. Insights based on Non-Graphical and Visual Analysis.
    - 6.1. Comments on the range of attributes.
    - 6.2. Comments on the distribution of the variables and relationship between them.
    - 6.3. Comments for each univariate and bivariate plot.
7. Business Insights.
8. Recommendations.

1. Defining Problem Statement and Analyzing basic metrics.

Import Libraries:

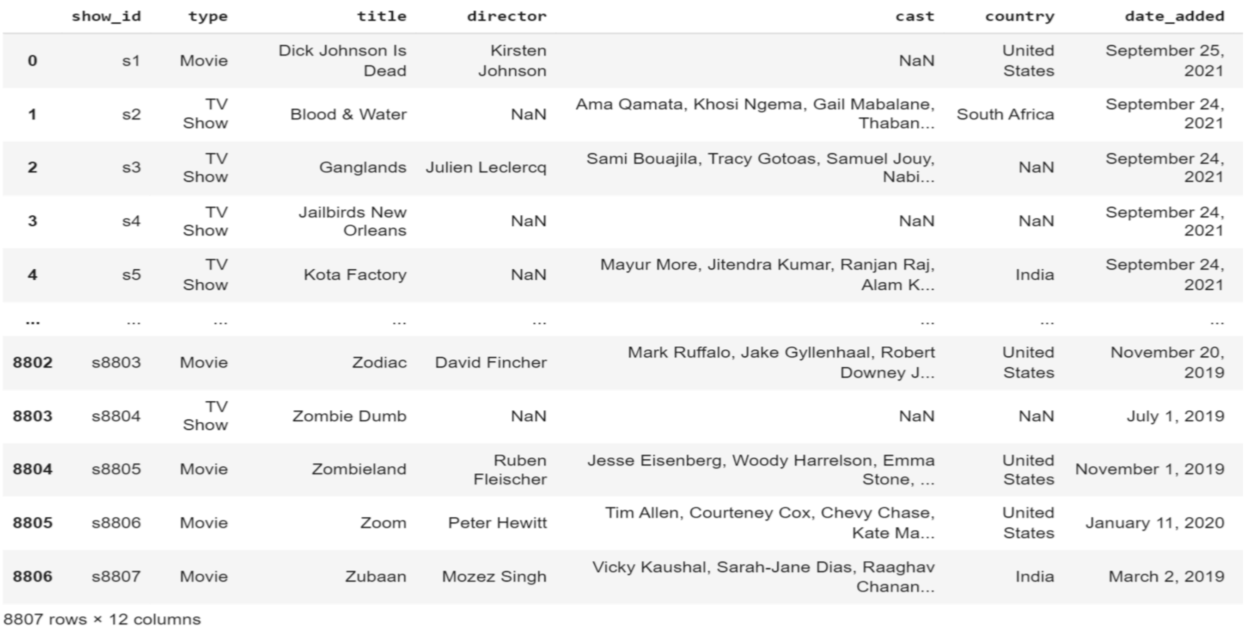
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

Importing Raw Data:

```
netflix = pd.read_csv('/content/netflix_titles.csv')
```

Inspect the Overview of the Data:

```
netflix
```



	show_id	type	title	director	cast	country	date_added
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021
...	...	...	...	...	...	...	...
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019

8807 rows × 12 columns

Dataset:

The dataset provided to you consists of a list of all the TV shows/movies available:

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: 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

Insights:

- The dataset contains 8,807 rows and 12 columns, which include some NaN (Not a Number) values and missing data.

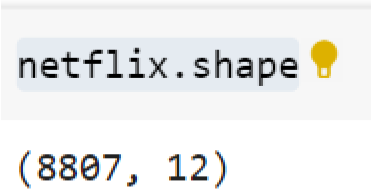
2. Observations on the shape of data, data types of all the attributes, conversion of categorical attributes to 'category' (If required), missing value detection, statistical summary.

The Shape of the Data:

Input:

```
netflix.shape
```

Output :



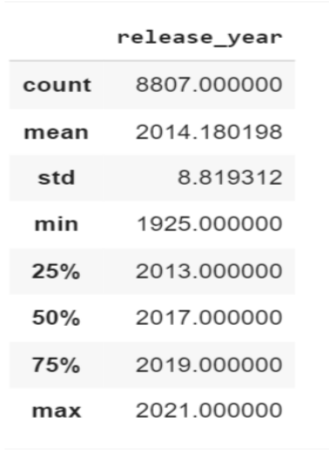
```
netflix.shape
(8807, 12)
```

Statical Summary Before Data Cleaning:

Input:

```
netflix.describe()
```

Output:



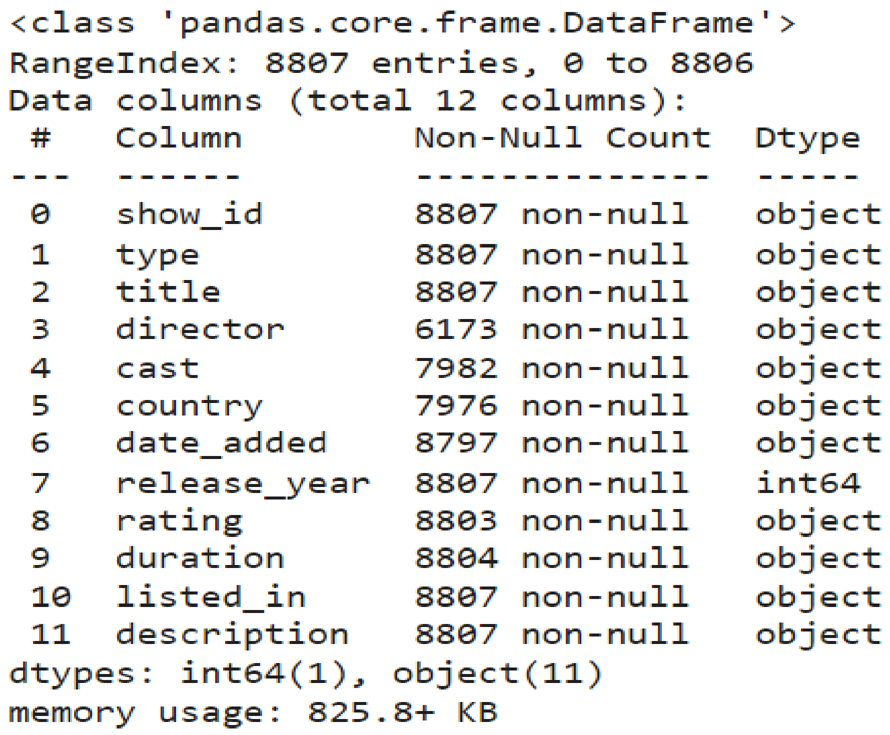
	release_year
count	8807.000000
mean	2014.180198
std	8.819312
min	1925.000000
25%	2013.000000
50%	2017.000000
75%	2019.000000
max	2021.000000

The Data type of all the Attributes:

Input:

```
    netflix.info()
```

Output:



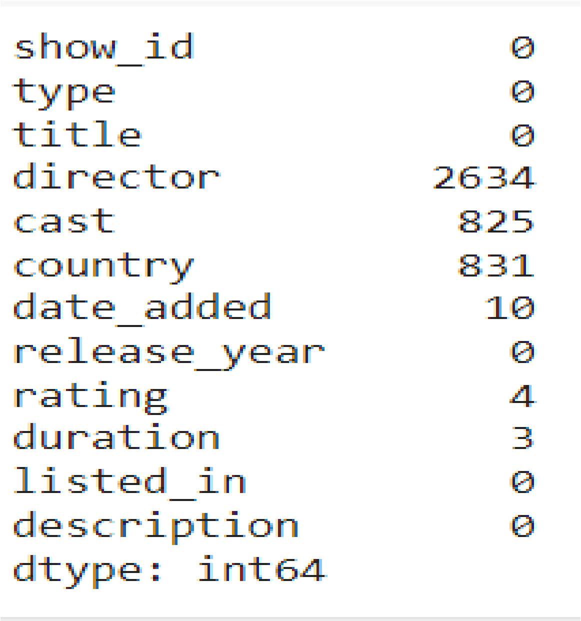
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
 2   title         8807 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
```

Missing Value Detection :

Input:

```
# Calculate the number of missing values in each column of the netflix
missing_values = netflix.isnull().sum()
# Display the number of missing values for each column
missing_values
```

Output:



```
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
```

Insights:

- Data cleaning involves identifying and rectifying incorrect, incomplete, inaccurate, or irrelevant data. It's a fundamental aspect of data science. In our analysis, we have 8,807 entries and 12 columns.

- Several columns contain null values: "director," "cast," "country," "date_added," "rating," and "duration." Specifically, there are 4,307 null values across the dataset, with 2,634 in "director," 825 in "cast," 831 in "country," 11 in "date_added," 4 in "rating," and 3 in "duration." We must address these null values

Data Cleaning:

- First, we need to split the 'Director' and 'Cast' columns, then merge them to ensure accurate data representation. Similarly, we need to split and then merge the 'Country' and 'Listed In' columns.

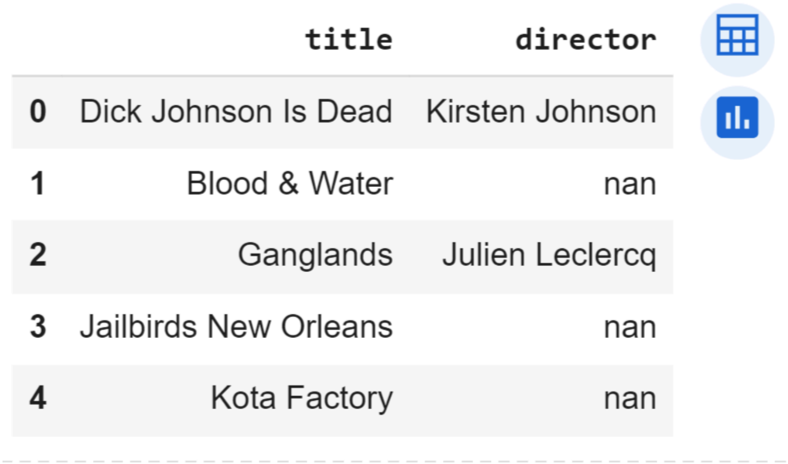
- Next, we must address the null values in the 'Date Added' column by replacing them with the mode of the dates for the respective release year. We will correct the 'Duration' values that were mistakenly shifted to the 'Rating' column using conditional statements.

- Additionally, we need to replace null values in the 'Country' column with the country of origin for the respective director or cast members. Finally, these cleaned and merged columns will be integrated back into the original dataset. This data preparation step is crucial before proceeding with Exploratory Data Analysis (EDA) and further analysis.

Input:

```
# Split the 'director' column by commas and create a DataFrame where each
title has multiple rows, one for each director
netflix_director_raw = pd.DataFrame(netflix['director'].apply(lambda x:
str(x).split(',')).tolist(), index=netflix['title'])

# Convert the DataFrame to a Series, where each title-director pair is a
separate row, then reset the index to turn the multi-index into columns
netflix_director = netflix_director_raw.stack().reset_index()

# Drop the unnecessary level_1 column which was created during the
stacking process
netflix_director.drop('level_1', axis=1, inplace=True)

# Rename the column with director information from 0 to 'director' for
better readability
netflix_director.rename(columns={0: 'director'}, inplace=True)

# Display the first few rows of the transformed DataFrame
netflix_director.head()
```

Output:

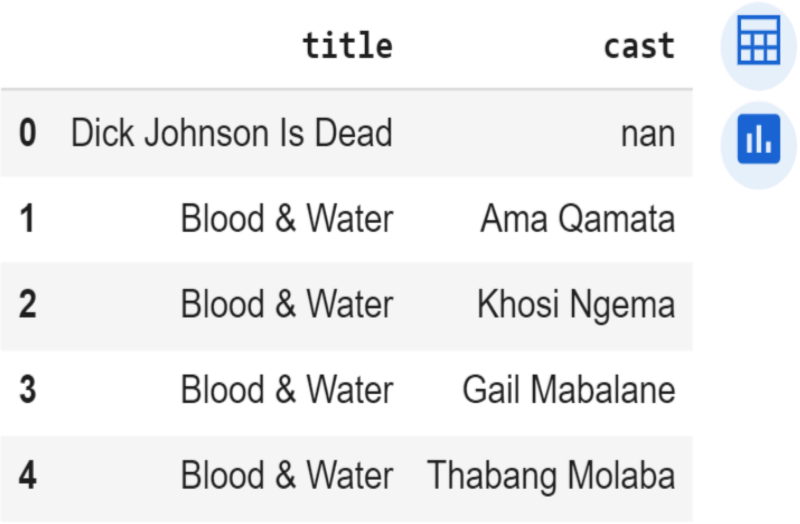


	title	director
0	Dick Johnson Is Dead	Kirsten Johnson
1	Blood & Water	nan
2	Ganglands	Julien Leclercq
3	Jailbirds New Orleans	nan
4	Kota Factory	nan

Input:

```
# Split the 'cast' column by commas and create a DataFrame where each title has multiple rows, one for each cast member
netflix_cast_raw = pd.DataFrame(netflix['cast'].apply(lambda x: str(x).split(',')).tolist(), index=netflix['title'])

# Convert the DataFrame to a Series, where each title-cast pair is a separate row, then reset the index to turn the multi-index into columns
netflix_cast = netflix_cast_raw.stack().reset_index()

# Drop the unnecessary level_1 column which was created during the stacking process
netflix_cast.drop('level_1', axis=1, inplace=True)

# Rename the column with cast information from 0 to 'cast' for better readability
netflix_cast.rename(columns={0: 'cast'}, inplace=True)

# Display the first few rows of the transformed DataFrame
netflix_cast.head()
```

Output:

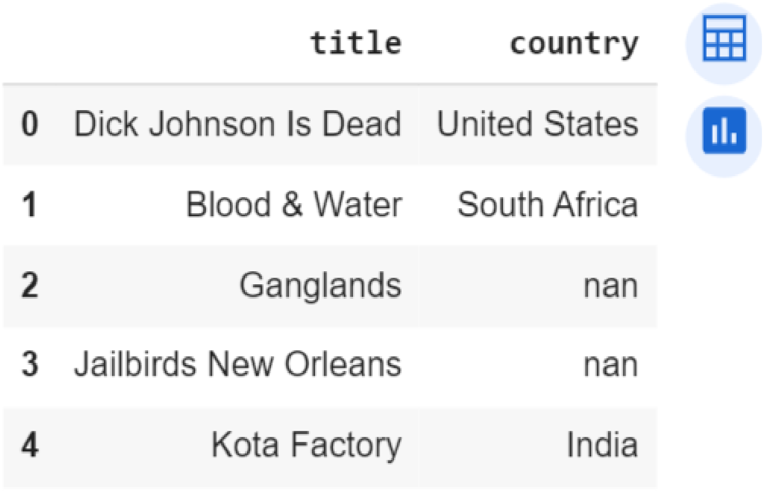


	title	cast
0	Dick Johnson Is Dead	nan
1	Blood & Water	Ama Qamata
2	Blood & Water	Khosi Ngema
3	Blood & Water	Gail Mabalane
4	Blood & Water	Thabang Molaba

Input:

```
# Split the 'country' column by commas and create a DataFrame where each title has multiple rows, one for each country
netflix_country_raw = pd.DataFrame(netflix['country'].apply(lambda x: str(x).split(',')).tolist(), index=netflix['title'])

# Convert the DataFrame to a Series, where each title-country pair is a separate row, then reset the index to turn the multi-index into columns
netflix_country = netflix_country_raw.stack().reset_index()

# Drop the unnecessary level_1 column which was created during the stacking process
netflix_country.drop('level_1', axis=1, inplace=True)

# Rename the column with country information from 0 to 'country' for better readability
netflix_country.rename(columns={0: 'country'}, inplace=True)

# Display the first few rows of the transformed DataFrame
netflix_country.head()
```

Output:

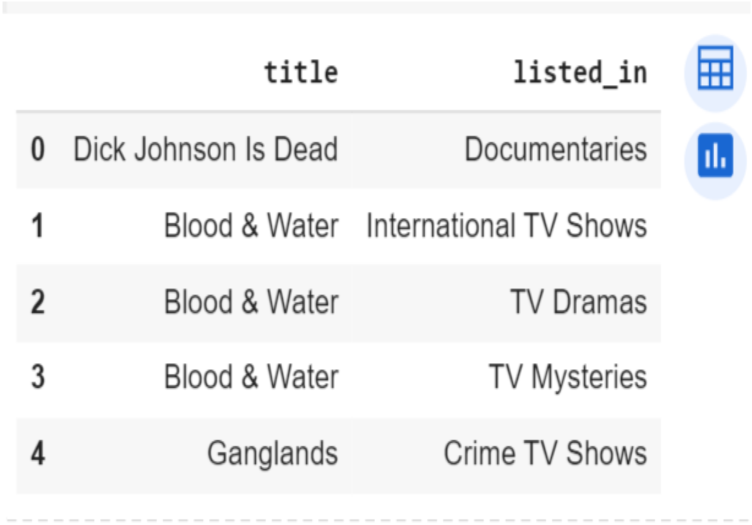


	title	country
0	Dick Johnson Is Dead	United States
1	Blood & Water	South Africa
2	Ganglands	nan
3	Jailbirds New Orleans	nan
4	Kota Factory	India

Input:

```
# Split the 'listed_in' column by commas and create a DataFrame where each title has multiple rows, one for each genre/category
netflix_listed_in_raw = pd.DataFrame(netflix['listed_in'].apply(lambda x: str(x).split(',')).tolist(), index=netflix['title'])

# Convert the DataFrame to a Series, where each title-genre pair is a separate row, then reset the index to turn the multi-index into columns
netflix_listed_in = netflix_listed_in_raw.stack().reset_index()

# Drop the unnecessary level_1 column which was created during the stacking process
netflix_listed_in.drop('level_1', axis=1, inplace=True)

# Rename the column with genre/category information from 0 to 'listed_in' for better readability
netflix_listed_in.rename(columns={0: 'listed_in'}, inplace=True)

# Display the first few rows of the transformed DataFrame
netflix_listed_in.head()
```

Output:



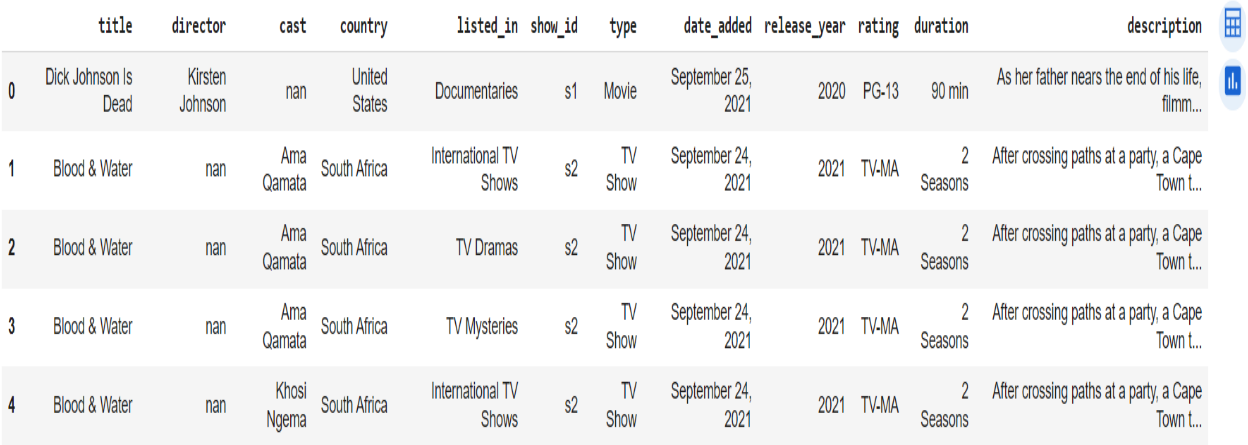
	title	listed_in
0	Dick Johnson Is Dead	Documentaries
1	Blood & Water	International TV Shows
2	Blood & Water	TV Dramas
3	Blood & Water	TV Mysteries
4	Ganglands	Crime TV Shows

Merging the Dataset Together:

Input:

```python
# Merge netflix_director with netflix_cast on the 'title' column
netflix_new1 = netflix_director.merge(netflix_cast, on='title')

# Merge netflix_country with netflix_listed_in on the 'title' column
netflix_new2 = netflix_country.merge(netflix_listed_in, on='title')

# Now merge the resulting datasets on the 'title' column to combine
them all
netflix_combined = netflix_new1.merge(netflix_new2, on='title')

# Now merge the resulting datasets on the 'title' column to combine
them all with the original data set netflix
netflix_final = netflix_combined.merge(netflix[['show_id',
'type','title','date_added',
       'release_year', 'rating', 'duration','description']],
on='title')
netflix_final.head()
```

Output:



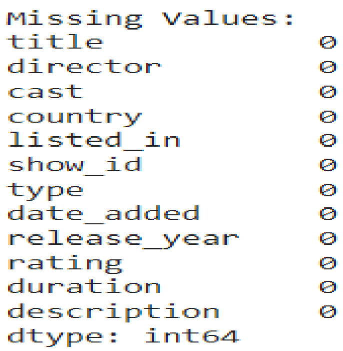
	title	director	cast	country	listed_in	show_id	type	date_added	release_year	rating	duration	description
0	Dick Johnson Is Dead	Kirsten Johnson	nan	United States	Documentaries	s1	Movie	September 25, 2021	2020	PG-13	90 min	As her father nears the end of his life, filmm...
1	Blood & Water	nan	Ama Qamata	South Africa	International TV Shows	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...
2	Blood & Water	nan	Ama Qamata	South Africa	TV Dramas	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...
3	Blood & Water	nan	Ama Qamata	South Africa	TV Mysteries	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...
4	Blood & Water	nan	Khosi Ngema	South Africa	International TV Shows	s2	TV Show	September 24, 2021	2021	TV-MA	2 Seasons	After crossing paths at a party, a Cape Town t...

Replacing the NaN values with a specific value that makes sense for our dataset.

Input:

```
# Replace NaN values in cast as Unknown Actor
netflix_final['cast'].replace(['nan'],['Unknown Actor'], inplace = True)

# Replace NaN values in cast as Unknown Director
netflix_final['director'].replace(['nan'],['Unknown Director'], inplace =
True)

# Calculate mode of release_year
release_year_mode = netflix_final['release_year'].mode()[0]

# Replace NaN values in date_added with mode of release_year
netflix_final['date_added'] =
netflix_final['date_added'].fillna(release_year_mode)

# Replacing NaN values in rating with Not rated
netflix_final['rating'].fillna('Not Rated', inplace=True)

# Replace NaN values in 'duration' with corresponding 'rating' values
netflix_final['duration'] = netflix_final.apply(lambda row: row['rating']
if pd.isna(row['duration']) else row['duration'],axis=1)

netflix_final.isnull().sum()
```

Output:



```
Missing Values:
title           0
director        0
cast            0
country         0
listed_in       0
show_id         0
type            0
date_added      0
release_year    0
rating          0
duration        0
description     0
dtype: int64
```

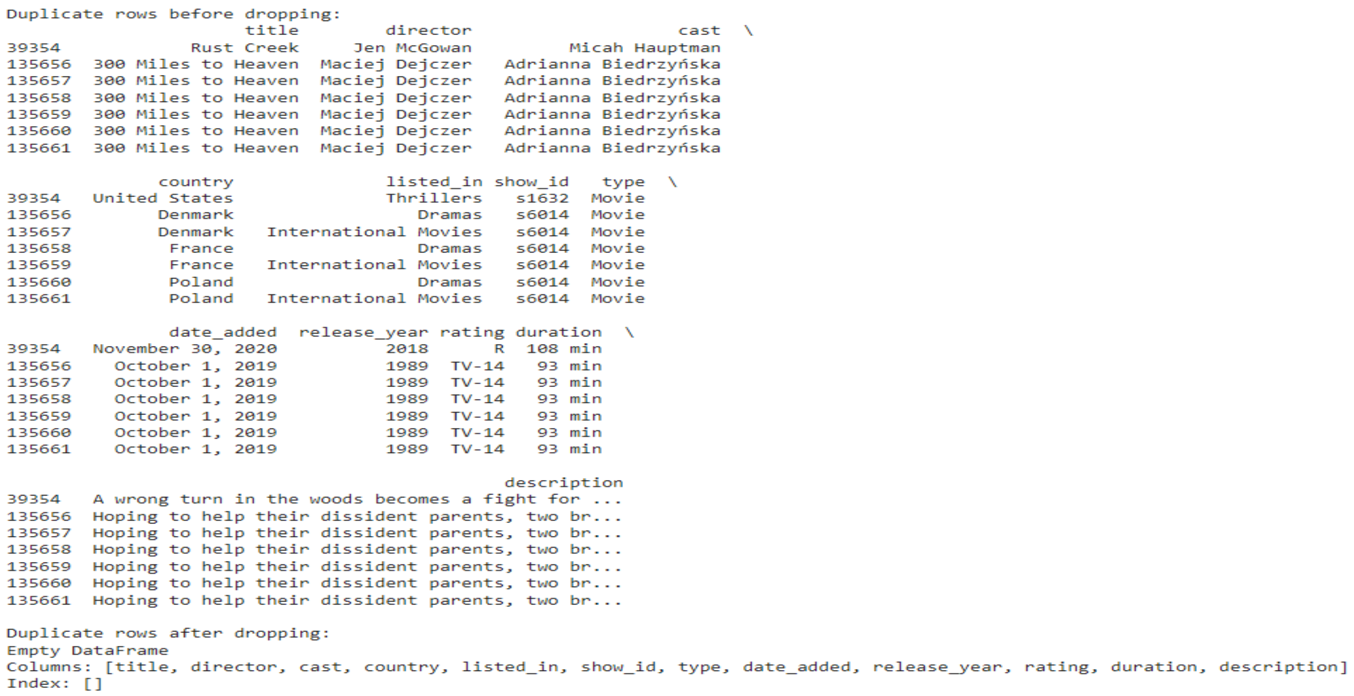
Duplicates:

- After cleaning the data, it's crucial to address duplicate values, as they can significantly impact the exploratory data analysis (EDA) and subsequent modeling. Removing duplicates and retaining the first instance ensures greater accuracy and reduces potential errors in forecasting.

Input:

```python
# Check for duplicates
print("Duplicate rows before dropping:")
print(netflix_final[netflix_final.duplicated()])

# Drop pure duplicates, keeping the first occurrence
netflix_final.drop_duplicates(inplace=True, keep='first')

# Check for duplicates again to verify
print("\nDuplicate rows after dropping:")
print(netflix_final[netflix_final.duplicated()])
```

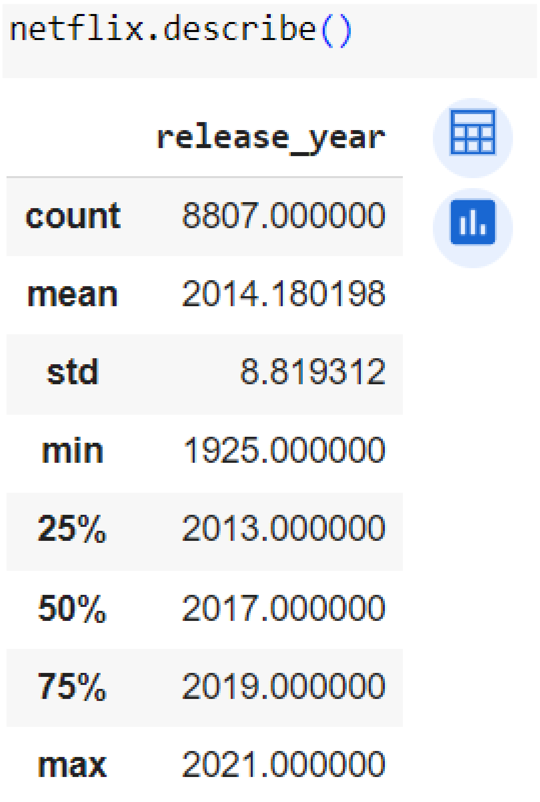
Output:



```
Duplicate rows before dropping:
                      title        director                  cast  \
39354            Rust Creek     Jen McGowan        Micah Hauptman   
135656  300 Miles to Heaven  Maciej Dejczer  Adrianna Biedrzyńska   
135657  300 Miles to Heaven  Maciej Dejczer  Adrianna Biedrzyńska   
135658  300 Miles to Heaven  Maciej Dejczer  Adrianna Biedrzyńska   
135659  300 Miles to Heaven  Maciej Dejczer  Adrianna Biedrzyńska   
135660  300 Miles to Heaven  Maciej Dejczer  Adrianna Biedrzyńska   
135661  300 Miles to Heaven  Maciej Dejczer  Adrianna Biedrzyńska   

              country             listed_in show_id   type  \
39354   United States             Thrillers   s1632  Movie   
135656        Denmark                Dramas   s6014  Movie   
135657        Denmark  International Movies   s6014  Movie   
135658         France                Dramas   s6014  Movie   
135659         France  International Movies   s6014  Movie   
135660         Poland                Dramas   s6014  Movie   
135661         Poland  International Movies   s6014  Movie   

               date_added  release_year rating duration  \
39354   November 30, 2020          2018      R  108 min   
135656    October 1, 2019          1989  TV-14   93 min   
135657    October 1, 2019          1989  TV-14   93 min   
135658    October 1, 2019          1989  TV-14   93 min   
135659    October 1, 2019          1989  TV-14   93 min   
135660    October 1, 2019          1989  TV-14   93 min   
135661    October 1, 2019          1989  TV-14   93 min   

                                             description  
39354   A wrong turn in the woods becomes a fight for ...  
135656  Hoping to help their dissident parents, two br...  
135657  Hoping to help their dissident parents, two br...  
135658  Hoping to help their dissident parents, two br...  
135659  Hoping to help their dissident parents, two br...  
135660  Hoping to help their dissident parents, two br...  
135661  Hoping to help their dissident parents, two br...  

Duplicate rows after dropping:
Empty DataFrame
Columns: [title, director, cast, country, listed_in, show_id, type, date_added, release_year, rating, duration, description]
Index: []
```

Statistical Summary Before Data Cleaning / / After Data Cleaning

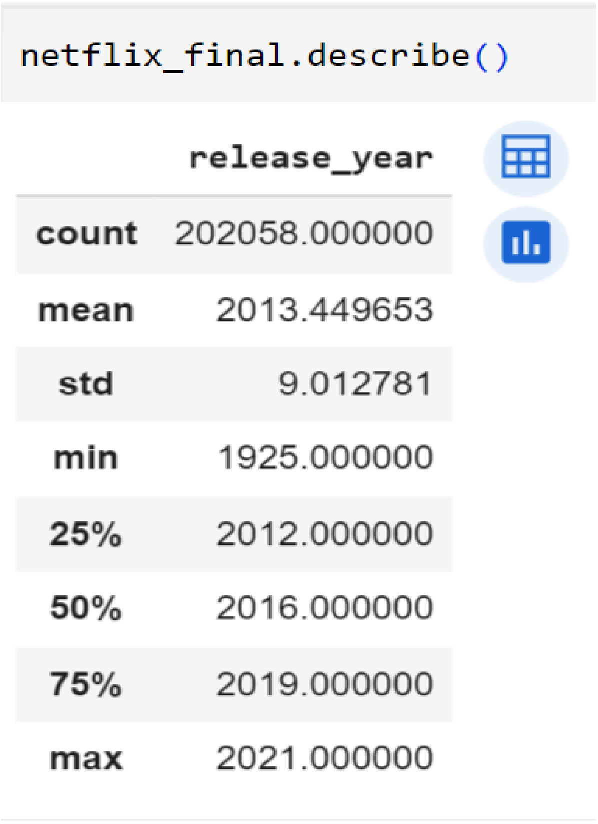
Before:



```
netflix.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

After:



```
netflix_final.describe()
```

	release_year
count	202058.000000
mean	2013.449653
std	9.012781
min	1925.000000
25%	2012.000000
50%	2016.000000
75%	2019.000000
max	2021.000000

Insights:

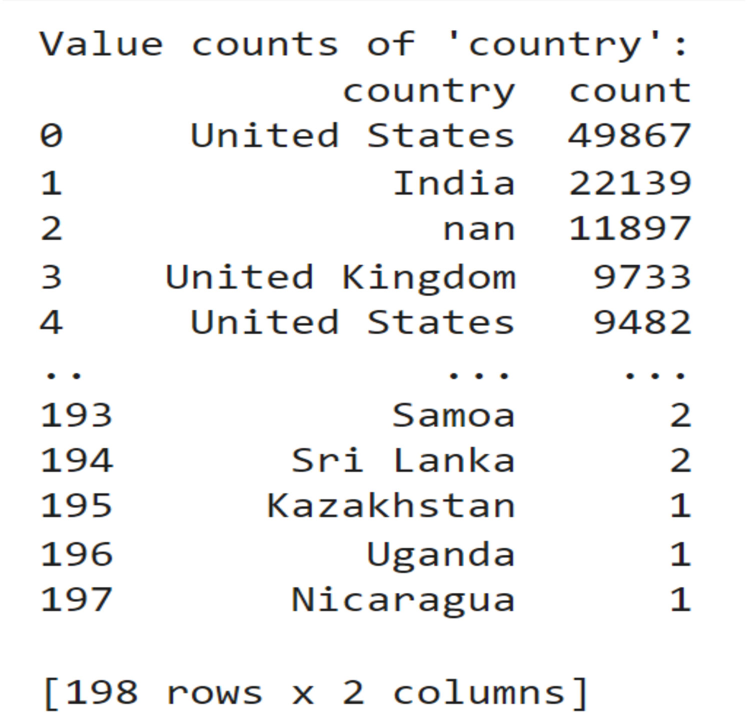
- Based on the data before and after cleaning, we can derive several insights. The dataset expanded significantly after cleaning, growing from 8,807 to 202,058 entries. Despite the increase in size, the mean release year remained relatively stable, decreasing marginally from 2014.18 to 2013.45. This suggests that the newly added data includes content from earlier years.
- The standard deviation also remained consistent around 9 years, indicating a similar spread of release years. The quartile ranges (25th, 50th, and 75th percentiles) did not change significantly, with the median release year (50th percentile) remaining at 2016. Overall, while the dataset saw substantial growth after cleaning, the distribution and characteristics of release years remained relatively unchanged, with older content being added alongside newer releases.

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

Input:

```
# Value counts for a specific column
country_counts =
netflix_final['country'].value_counts().reset_index()
country_counts.columns = ['country', 'count']
```

Output:

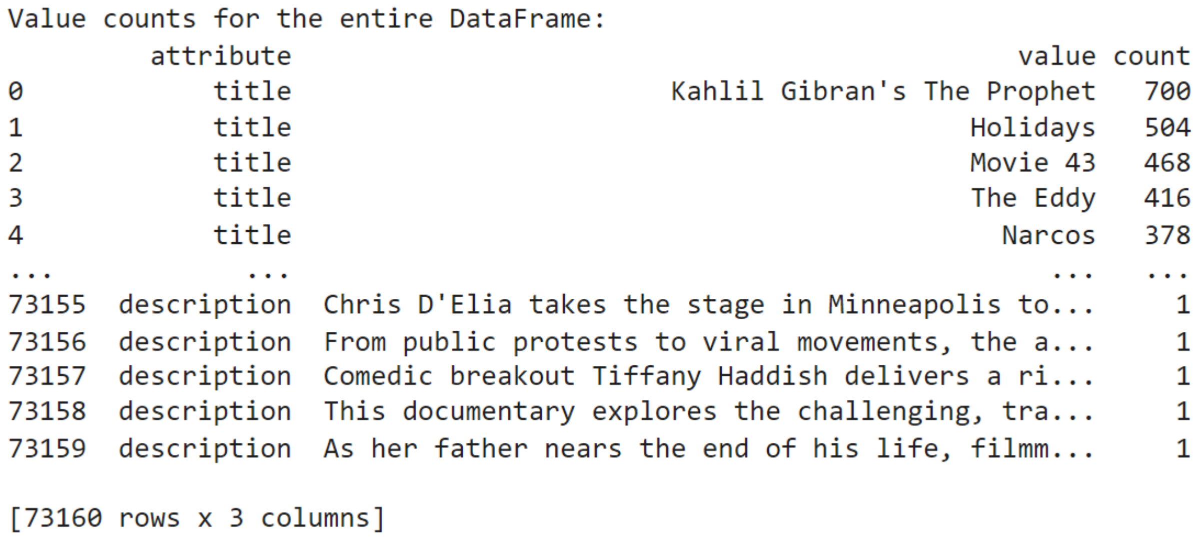


```
Value counts of 'country':
              country  count
0       United States  49867
1               India  22139
2                 nan  11897
3      United Kingdom   9733
4       United States   9482
..                ...    ...
193             Samoa      2
194         Sri Lanka      2
195        Kazakhstan      1
196            Uganda      1
197         Nicaragua      1

[198 rows x 2 columns]
```

Input:

```
# Value counts for the entire DataFrame
value_counts_df = pd.DataFrame(columns=['attribute', 'value',
'count'])

for col in netflix_final.columns:
    value_counts = netflix_final[col].value_counts().reset_index()
    value_counts.columns = ['value', 'count']
    value_counts['attribute'] = col
    value_counts_df = pd.concat([value_counts_df, value_counts],
ignore_index=True)

print("Value counts of 'country':")
print(country_counts)
print("\nValue counts for the entire DataFrame:")
print(value_counts_df)
```

Output:



```
Value counts for the entire DataFrame:
         attribute                                            value count
0            title                      Kahlil Gibran's The Prophet   700
1            title                                         Holidays   504
2            title                                         Movie 43   468
3            title                                         The Eddy   416
4            title                                          Narcos   378
...            ...                                              ...   ...
73155  description  Chris D'Elia takes the stage in Minneapolis to...     1
73156  description  From public protests to viral movements, the a...     1
73157  description  Comedic breakout Tiffany Haddish delivers a ri...     1
73158  description  This documentary explores the challenging, tra...     1
73159  description  As her father nears the end of his life, filmm...     1

[73160 rows x 3 columns]
```

Insights:

- Netflix's catalog is heavily dominated by content from the United States, India, and the United Kingdom, which collectively account for a substantial portion of its titles. The United States leads with 49,867 entries, highlighting Netflix's focus on American content and its origin.

- India follows closely with 22,139 titles, reflecting Netflix's strategic investment in the Indian market to expand its subscriber base. The United Kingdom rounds out the top three with 9,733 titles, indicating significant representation of British content, which caters to English-speaking audiences worldwide.

- This distribution underscores Netflix's strategy of providing diverse content to appeal to global viewers, while also targeting specific regional markets with localized programming.

- Conversely, some countries have minimal representation in Netflix's catalog. Nicaragua, Uganda, and Kazakhstan each have only one title available, suggesting limited content availability from these regions. This may stem from smaller local production industries or lower international demand for their content.

- Netflix's challenge lies in balancing its global content strategy with localized offerings to appeal to diverse audiences worldwide. Understanding these dynamics is crucial for Netflix to optimize content acquisition strategies and enhance its global market presence effectively.

- Title Frequency: The high frequency of titles like "Kahlil Gibran's The Prophet," "Holidays," and "Movie 43" suggests these are prominent entries in the Netflix dataset, possibly indicating their popularity or multiple versions available.

- Description Variability: The unique nature of descriptions such as those featuring Chris D'Elia, public protests, and Tiffany Haddish suggests diverse content offerings, including specials, documentaries, and unique programming.

- These insights provide a glimpse into the diversity and distribution of content within the Netflix dataset, highlighting popular titles and unique content offerings. Understanding these patterns can guide content curation and user engagement strategies for Netflix.

4. Visual Analysis - Univariate, Bivariate after pre-processing of the data.Pre-processing involves unnesting of the data in columns like Actor, Director, Country. (Please Refer to Page 8)

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

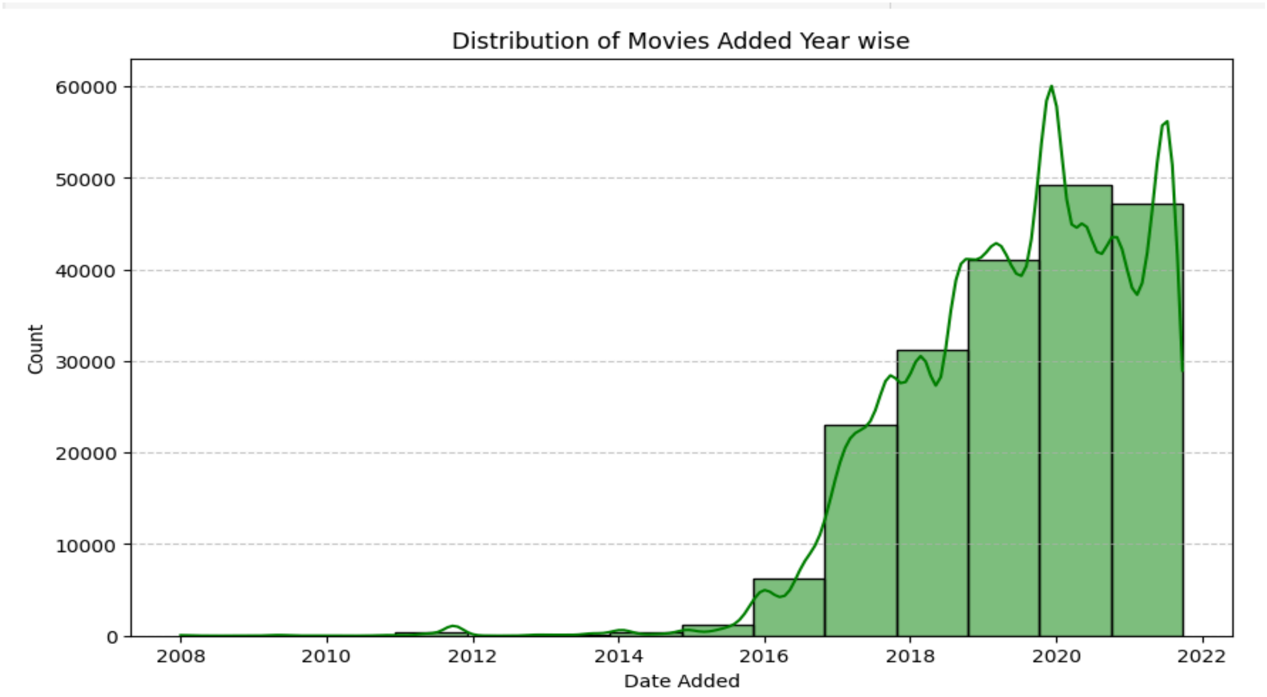
Distplot:

- A Dist Plot, short for distribution plot, is a type of plot that displays the distribution of a continuous variable. It is a combination of a histogram and a kernel density plot (KDE) that provides a visual representation of the data distribution.

Input:

```
# Convert date_added to datetime format and extract year
netflix_final['date_added'] =
pd.to_datetime(netflix_final['date_added'], errors='coerce')
netflix_final['date_added_year'] =
netflix_final['date_added'].dt.year

# Distplot (for continuous variable)
plt.figure(figsize=(10, 6))
sns.histplot(netflix_final['date_added'].dropna(), kde=True,
color='green', binwidth=365)  # Increase binwidth to 1 year
plt.title('Distribution of Movies Added Year wise')
plt.xlabel('Date Added')
plt.ylabel('Count')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

Output:



Insights:

- Netflix has seen a remarkable increase in content additions since it gained significant traction after 2013. By analyzing the data, it's clear that Netflix's library has been expanding rapidly each year.

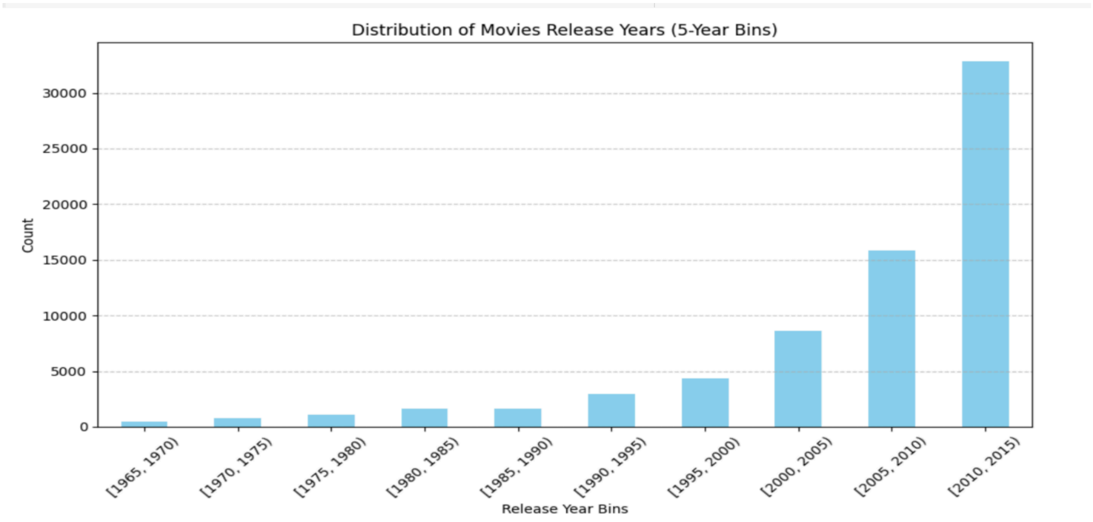
- The introduction of the "year_added" column reveals that the platform has been consistently adding new titles since its inception. Notably, the growth in the number of movies has outpaced that of TV shows. In both 2018 and 2019, approximately 1,300 new movies were added, indicating a strong focus on expanding its movie catalog.

- This trend suggests that Netflix is increasingly prioritizing movie content over TV shows in recent years, aligning its strategy with evolving viewer preferences and market demands.

Input:

```python
# Extract release years
release_years = netflix_final['release_year']

# Calculate the distribution
year_counts = release_years.value_counts().sort_index()

bins = pd.cut(netflix_final['release_year'], bins=range(1965, 2020, 5), right=False)

# Plotting
plt.figure(figsize=(10, 6))
netflix_final.groupby(bins)['release_year'].count().plot(kind='bar', color='skyblue')
plt.title('Distribution of Movies Release Years (5-Year Bins)')
plt.xlabel('Release Year Bins')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout()
plt.show()
```

Output:



Insights:

- Overall, the data underscores Netflix's commitment to enriching its content offerings over time. The platform's focus on expanding its movie library, particularly in the past few years, highlights its strategy to cater to diverse audience tastes and maintain its position as a leading streaming service globally.

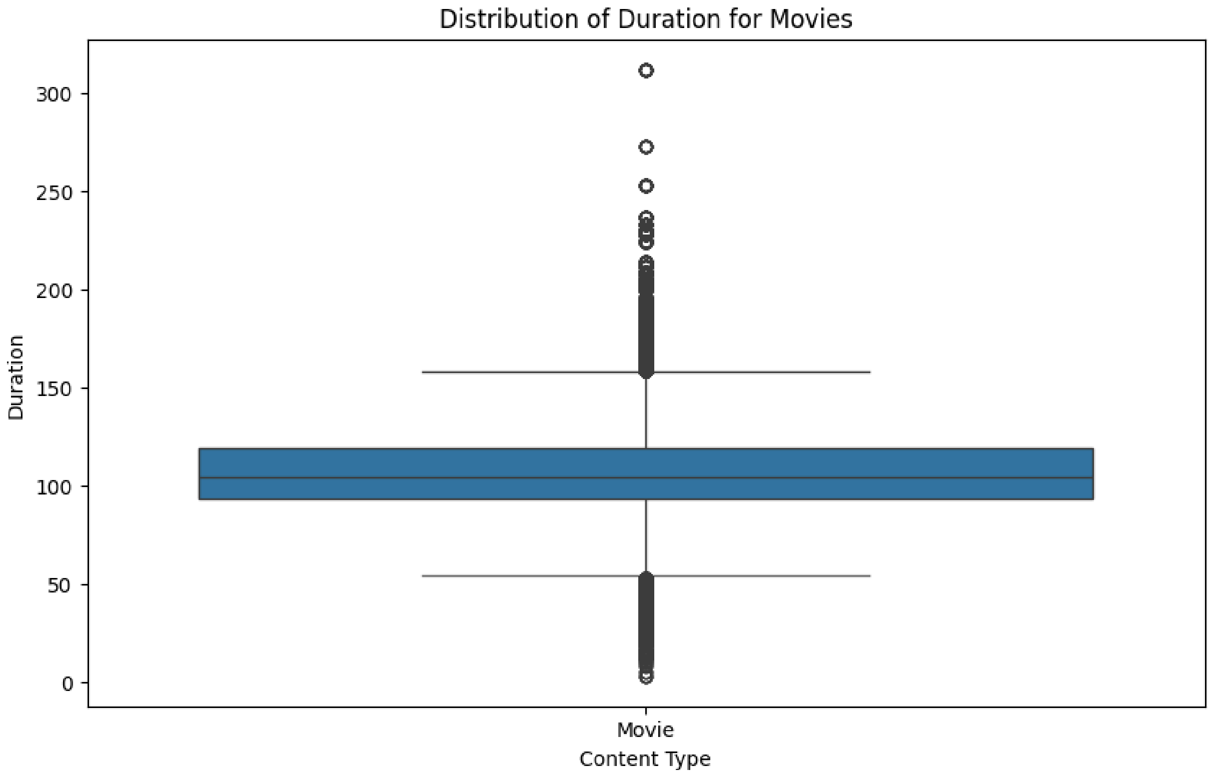
- These insights capture the trends in Netflix's content expansion over the years, emphasizing the platform's strategic shift towards enhancing its movie collection while still maintaining a steady growth in TV show additions.

4.2. For categorical variable(s): Boxplot.

Input:

```
netflix_movies_df = netflix_final[netflix_final['type'] == 'Movie']

# Extracting duration and converting to integer
netflix_movies_df['duration'] =
netflix_movies_df['duration'].str.extract('(\d+)',
expand=False).astype(int)

# Creating a boxplot for movie duration
plt.figure(figsize=(10, 6))
sns.boxplot(data=netflix_movies_df, x='type', y='duration')
plt.xlabel('Content Type')
plt.ylabel('Duration')
plt.title('Distribution of Duration for Movies')
plt.show()
```

Output:



Insights:

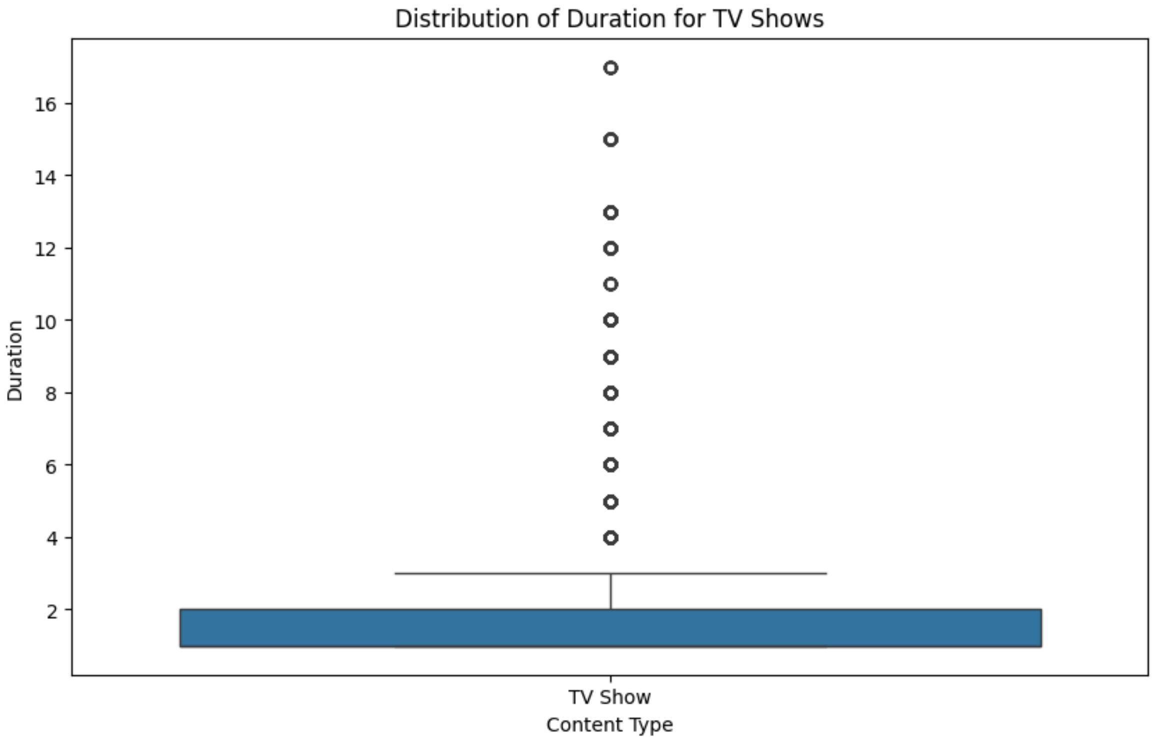
- Analyzing the box plot of movie durations on Netflix reveals that the majority of movies fall within a reasonable duration range, with only a few outliers extending to approximately 2.5 hours or more. This indicates that Netflix's movie offerings are generally designed to adhere to standard viewing times, catering to typical viewer preferences for movie length.

- This insight suggests that most movies available on Netflix are structured to fit into a standard viewing time frame, ensuring they appeal to a wide audience while also accommodating the occasional longer film for those seeking more extended cinematic experiences.

Input:

```
netflix_shows_df = netflix_final[netflix_final['type'] == 'TV Show']

# Extracting duration and converting to integer
netflix_shows_df['duration'] =
netflix_shows_df['duration'].str.extract('(\d+)',
expand=False).astype(int)

# Creating a boxplot for TV show duration
plt.figure(figsize=(10, 6))
sns.boxplot(data=netflix_shows_df, x='type', y='duration')
plt.xlabel('Content Type')
plt.ylabel('Duration')
plt.title('Distribution of Duration for TV Shows')
plt.show()
```

Output:

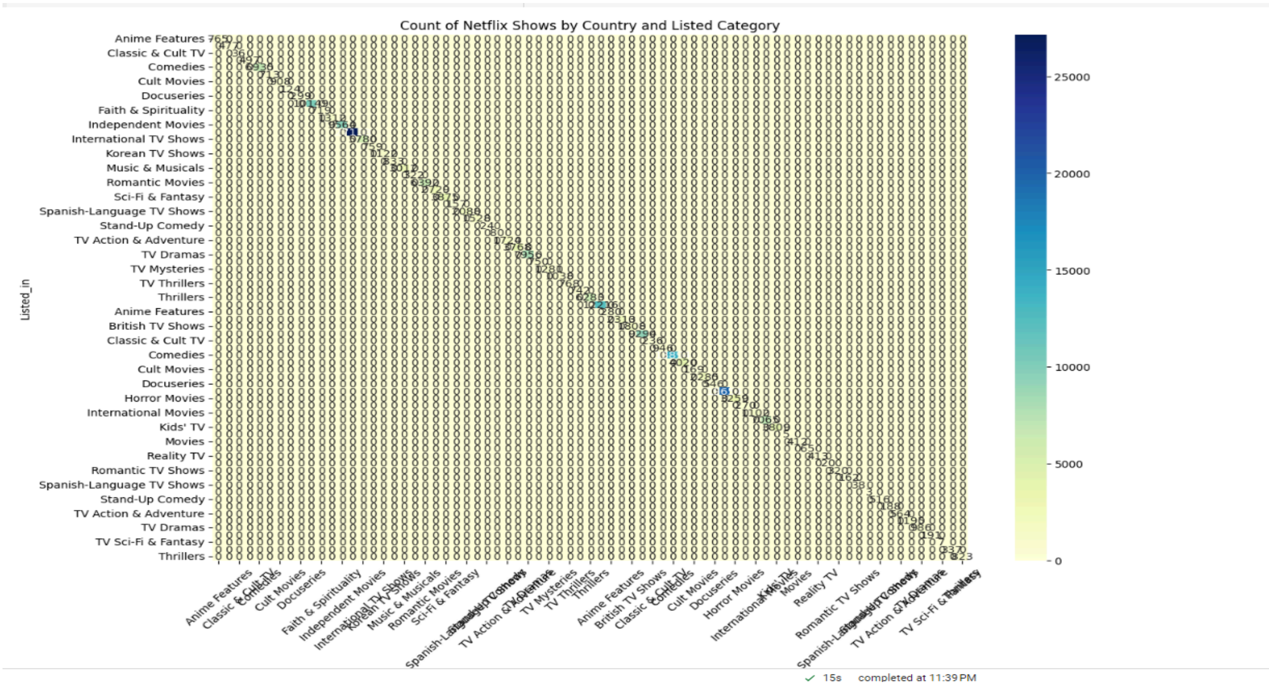


4.3. For correlation: Heatmaps, Pairplots.

Input:

```
heatmap_data = netflix_final.groupby(['listed_in',
'listed_in']).size().unstack(fill_value=0)

plt.figure(figsize=(14, 10))
sns.heatmap(heatmap_data, cmap='YlGnBu', annot=True, fmt='d',
cbar=True)
plt.title('Count of Netflix Shows by Country and Listed Category')
plt.xlabel('Listed in')
plt.ylabel('Listed_in')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

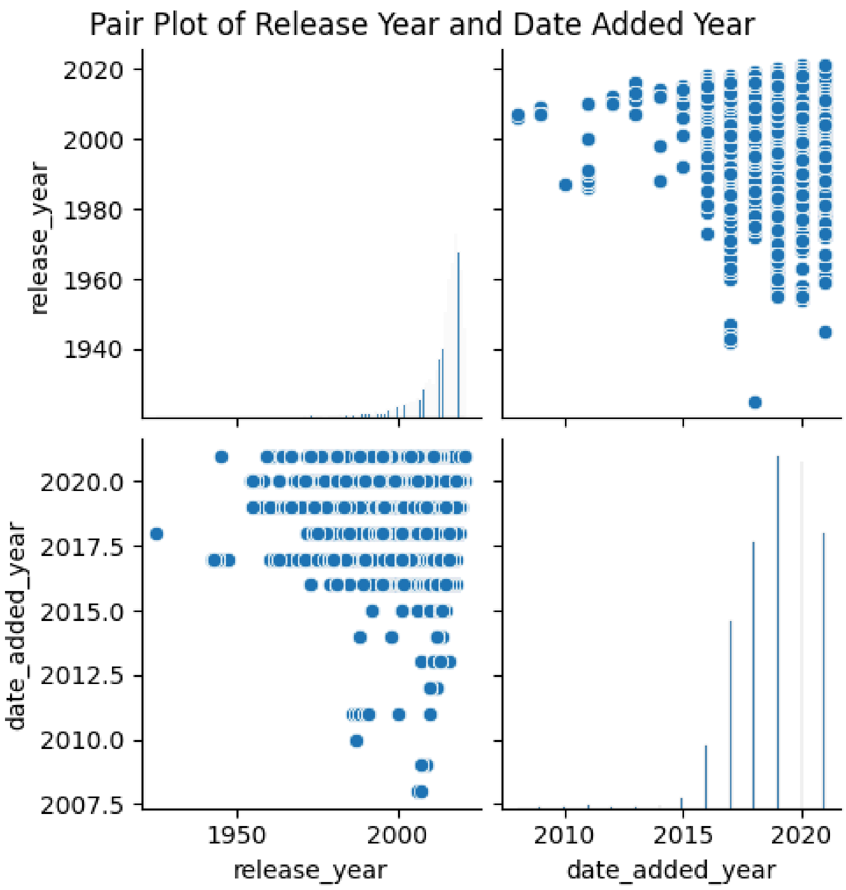
Output:



Input:

```python
import seaborn as sns
import matplotlib.pyplot as plt

# Filter out non-numeric columns for pair plotting
numeric_columns = ['release_year', 'date_added_year']

# Create pair plot
sns.pairplot(netflix_final[numeric_columns])
plt.suptitle('Pair Plot of Release Year and Date Added Year', y=1.02)
plt.show()
```

Output:



Insights :

- Genres play a crucial role in how Netflix categorizes and organizes its vast content library. The correlation analysis of genres reveals interesting relationships between different types of content. The heatmap generated from the genre correlation matrix illustrates strong positive correlations between specific genres. For instance, TV Dramas and International TV Shows show a significant positive correlation, indicating that viewers who enjoy one genre are likely to enjoy the other. Similarly, Romantic TV Shows and International TV Shows also exhibit a strong positive correlation, suggesting that romantic content tends to have an international appeal.

- This insight underscores Netflix's strategy of curating content that appeals to diverse audience tastes and preferences. By leveraging genre correlations, Netflix can effectively recommend related content to viewers, enhancing their viewing experience and satisfaction. This approach not only helps in organizing their extensive catalog but also ensures that users discover content that aligns with their interests, ultimately contributing to Netflix's continued success in the streaming industry.

5. Missing Value & Outlier check (Treatment optional).

What is an Outlier?

An outlier in a dataset refers to an observation that significantly deviates from the majority of other data points. It is an unusual or abnormal value that stands out from the rest of the data. For instance, in a dataset like [10, 15, 22, 330, 30, 45, 60], the value 330 is an outlier because it is much larger than the other values. Detecting outliers is crucial because they can skew statistical analyses and machine learning models, leading to unreliable predictions. Techniques like visual detection using box plots, which utilize the Interquartile Range (IQR), are commonly used to identify outliers in data.

Why Do We Need to Treat Outliers?

Outliers can adversely affect the performance of machine learning models such as linear regression, logistic regression, and support vector machines. These models are sensitive to extreme values, which can distort the mathematical relationships within the data and produce inaccurate results. While some outliers may represent genuine anomalies in the data, others can simply be errors or noise. Therefore, it is essential to properly handle outliers to ensure the robustness and accuracy of analytical and predictive models.

Visual Detection: Box Plots

Box plots are effective visual tools for detecting outliers and understanding the distribution of data. They display the data's quartiles (25th, 50th, and 75th percentiles) as well as the outliers beyond the whiskers, which represent the data boundaries. Outliers, which lie outside the whiskers, are easily identified on a box plot. This method helps in visually assessing the range and distribution of data, making it easier to interpret and analyze outliers in datasets.

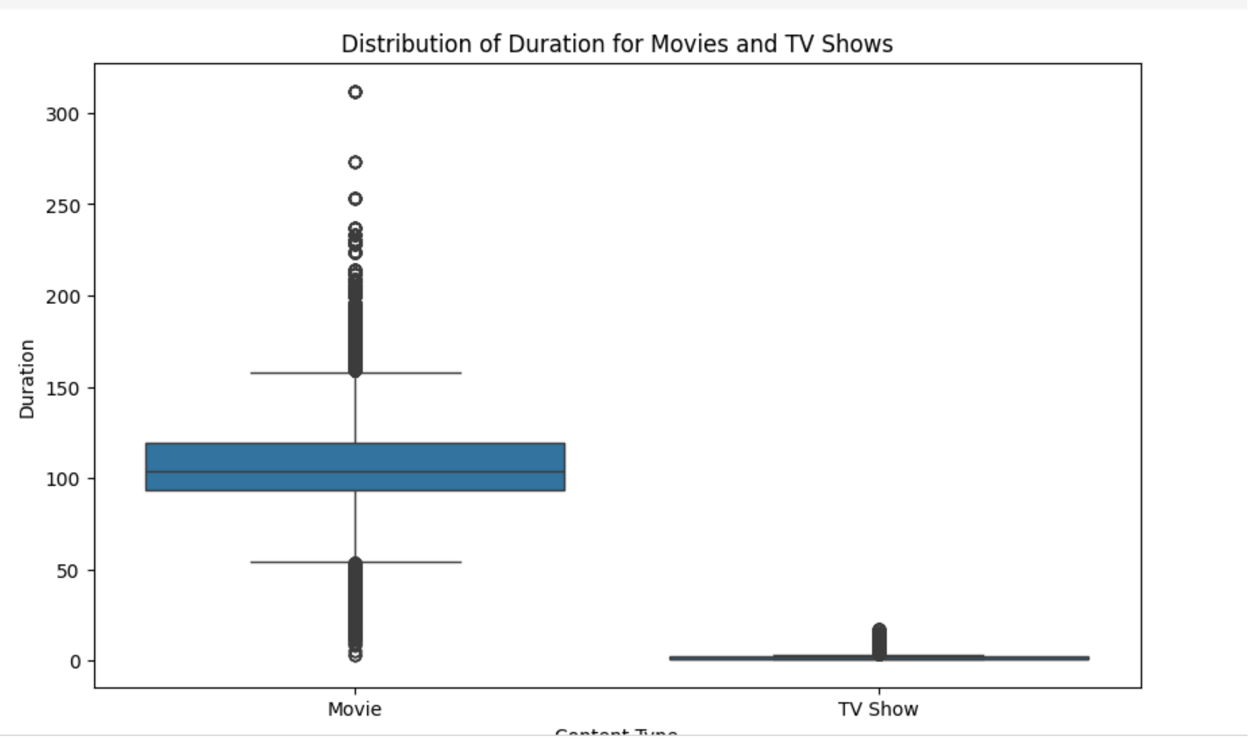
Duration Distribution for Movies and TV Shows

Analyzing the distribution of durations for movies and TV shows allows us to gain insights into the typical lengths of content available on Netflix. By creating box plots for these distributions, we can visualize the spread of durations and identify outliers or standard durations. This analysis helps in understanding viewer preferences and content trends, guiding decisions on content production and platform optimization.

Input:

```
# Extract duration for movies and TV shows
netflix_final['duration'] = netflix_final['duration'].str.extract('(\d+)',
expand=False).astype(float)

# Create a boxplot for movie and TV show duration
plt.figure(figsize=(10, 6))
sns.boxplot(data=netflix_final, x='type', y='duration')
plt.xlabel('Content Type')
plt.ylabel('Duration')
plt.title('Distribution of Duration for Movies and TV Shows')
plt.show()
```

Output:



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

6.1. Comments on the range of attributes.

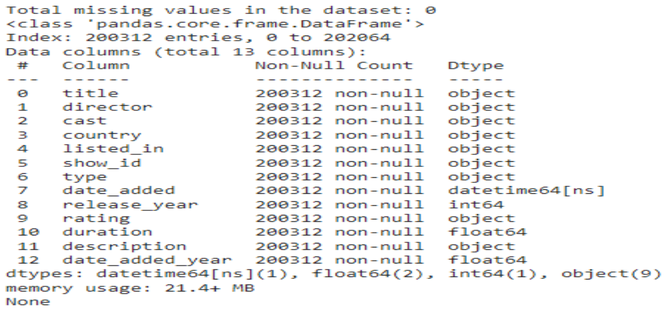
6.2. Comments on the distribution of the variables and relationship between them.

6.3. Comments for each univariate and bivariate plot.

Input:

```python
# Replace missing values in 'director', 'cast', and 'country'
netflix_final['director'].fillna('No Director', inplace=True)
netflix_final['cast'].fillna('No Cast', inplace=True)
netflix_final['country'].fillna('Country Unavailable', inplace=True)

# Drop rows with missing values in 'date_added' and 'rating'
netflix_final.dropna(subset=['date_added', 'rating'], inplace=True)

# Check if there are any remaining missing values
missing_values_count = netflix_final.isnull().sum().sum()
print(f'Total missing values in the dataset: {missing_values_count}')

# Display the updated dataset information
print(netflix_final.info())
```

Output



```
Total missing values in the dataset: 0
<class 'pandas.core.frame.DataFrame'>
Index: 200312 entries, 0 to 202064
Data columns (total 13 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   title            200312 non-null  object
 1   director         200312 non-null  object
 2   cast             200312 non-null  object
 3   country          200312 non-null  object
 4   listed_in        200312 non-null  object
 5   show_id          200312 non-null  object
 6   type             200312 non-null  object
 7   date_added       200312 non-null  datetime64[ns]
 8   release_year     200312 non-null  int64
 9   rating           200312 non-null  object
 10  duration         200312 non-null  float64
 11  description      200312 non-null  object
 12  date_added_year  200312 non-null  float64
dtypes: datetime64[ns](1), float64(2), int64(1), object(9)
memory usage: 21.4+ MB
None
```

7. Business Insights.

Quantity: Our analysis shows that Netflix has added a larger number of movies compared to TV shows, reflecting the dominance of movies in their content library as expected.

Content Addition: July emerged as the month when Netflix adds the most content, closely followed by December, indicating a strategic approach to content release aligned with holidays and seasonal trends.

Genre Correlation: Strong positive correlations were observed between various genres, such as TV dramas and international TV shows, romantic and international TV shows, and independent movies and dramas. These correlations provide insights into viewer preferences and content interconnections.

Movie Lengths: Analysis of movie durations indicated a peak around the 1960s, stabilizing around 100 minutes, reflecting trends in movie lengths over time.

TV Show Episodes: Most TV shows on Netflix consist of one season, suggesting a preference for shorter series among viewers.

Common Themes: Words like love, life, family, and adventure were frequently found in titles and descriptions, capturing recurring themes in Netflix content.

Rating Distribution: The distribution of ratings over the years offers insights into the evolving content landscape and audience reception.

Data-Driven Insights: Our data analysis journey showcased the power of data in unraveling the mysteries of Netflix's content landscape, providing valuable insights for viewers and content creators.

Continued Relevance: As the streaming industry evolves, understanding these patterns and trends becomes increasingly essential for navigating the dynamic landscape of Netflix and its vast library.

Happy Streaming: We hope this analysis has been an enlightening and entertaining journey into the world of Netflix. We encourage you to explore the captivating stories within its ever-changing content offerings. Let data guide your streaming adventures

8. Recommendations.

Diversify Content: Netflix should focus more on TV shows to cater to viewers who prefer serialized content over movies.

Collaborate with Directors: Partnering with both top directors and promising directors with fewer movies and high ratings could lead to compelling new content that resonates with audiences.

Genre Expansion: While international movies are popular, Netflix should prioritize expanding genres like horror and comedy to attract a wider audience.

Focus on Thrillers in TV Shows: Developing thriller genres in TV shows can lead to longer-running series, increasing viewer engagement.

Strategic Release Planning: OTT releases should be strategically planned around holidays, year-end, and weekends to maximize viewer engagement and subscriptions.

Direct-to-OTT Releases: Consider releasing well-received movies directly to OTT platforms to leverage positive word-of-mouth and boost subscriptions.

Utilize Popular Actors: Leveraging actors with substantial followings for TV shows or web series can attract dedicated fan bases and increase viewership.

Increase International Advertisement: Expand advertising efforts in countries with fewer Netflix releases to attract local audiences and enhance market presence.