1. Defining Problem Statement and Analyzing Basic Metrics (10 Points)

Problem Statement: Analyze Netflix data to generate insights that can help Netflix decide which types of shows or movies to produce and how they can grow the business in different countries.

Basic Metrics:

Load the dataset and inspect its basic structure and summary.

```
import pandas as pd

# Load the dataset
df = pd.read_csv('netflix.csv')

# Display basic information
print(df.info())

# Display basic metrics
print(df.describe())
```

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

```
# Display the first few rows of the dataframe
df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021
3	s4	TV Show	Jailbirds New	NaN	NaN	NaN	September 24, 2021	2021

Next steps: Generate code with df View recommended plots

```
# Display the last few rows of the dataframe
df.tail()
```

	show_id	type	title	director	cast	country	date_added	release_y
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	2
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	2
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	2
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2
				Mozez	Vicky Kaushal, Sarah-		March 2,	

```
# Check the data types of each column
df.dtypes
```

```
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
```

```
#Describe the data
print(df.describe(include='all'))
```

```
       show_id   type                title       director  \
count     8807   8807                 8807           6173
unique    8807      2                 8807           4528
top         s1  Movie  Dick Johnson Is Dead  Rajiv Chilaka
freq         1   6131                    1             19
mean       NaN    NaN                  NaN            NaN
std        NaN    NaN                  NaN            NaN
min        NaN    NaN                  NaN            NaN
25%        NaN    NaN                  NaN            NaN
50%        NaN    NaN                  NaN            NaN
75%        NaN    NaN                  NaN            NaN
max        NaN    NaN                  NaN            NaN

                      cast        country       date_added  release_year  \
count                 7982           7976             8797   8807.000000
unique                7692            748             1767           NaN
top     David Attenborough  United States  January 1, 2020           NaN
freq                    19           2818              109           NaN
mean                   NaN            NaN              NaN   2014.180198
std                    NaN            NaN              NaN      8.819312
min                    NaN            NaN              NaN   1925.000000
25%                    NaN            NaN              NaN   2013.000000
50%                    NaN            NaN              NaN   2017.000000
75%                    NaN            NaN              NaN   2019.000000
max                    NaN            NaN              NaN   2021.000000

       rating  duration                     listed_in  \
count    8803      8804                          8807
unique     17       220                           514
top     TV-MA  1 Season  Dramas, International Movies
freq     3207      1793                           362
mean      NaN       NaN                           NaN
std       NaN       NaN                           NaN
min       NaN       NaN                           NaN
25%       NaN       NaN                           NaN
50%       NaN       NaN                           NaN
75%       NaN       NaN                           NaN
max       NaN       NaN                           NaN

                                          description
count                                            8807
unique                                           8775
top     Paranormal activity at a lush, abandoned prope...
freq                                                4
mean                                              NaN
std                                               NaN
min                                               NaN
25%                                               NaN
50%                                               NaN
75%                                               NaN
max                                               NaN
```

2. Observations on Data Shape, Data Types, Categorical Conversion, Missing Values, and Statistical Summary (10 Points)

```
# Observations on the shape of data
print("Shape of the data:", df.shape)

# Data types of all attributes
print("Data types:\n", df.dtypes)

# Convert categorical attributes to 'category'
df['type'] = df['type'].astype('category')
df['rating'] = df['rating'].astype('category')
df['country'] = df['country'].astype('category')
df['listed_in'] = df['listed_in'].astype('category')
```

```
Shape of the data: (8807, 12)
Data types:
 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
```

```
# Check for missing values
df.isnull().sum()
```

```
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
```

```
# Handle missing values (example: fill with mean, median, or drop)

# Fill missing values
df['director'].fillna('Unknown', inplace=True)
df['cast'].fillna('Unknown', inplace=True)
df['country'] = df['country'].cat.add_categories(['Unknown']).fillna('Unknown')
df['date_added'].fillna('Unknown', inplace=True)
df['duration'].fillna('Unknown', inplace=True)

# Replace missing ratings with the most frequent rating
most_frequent_rating = df['rating'].mode()[0]
df['rating'].fillna(most_frequent_rating, inplace=True)

# Verify that all missing values have been handled
print(df.isnull().sum())
```

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

```
# Remove duplicates
df = df.drop_duplicates()
```

```
# Replace "Unknown" in date_added with NaT
df['date_added'].replace('Unknown', pd.NaT, inplace=True)

# Convert date_added to datetime
df['date_added'] = pd.to_datetime(df['date_added'], format="%B %d, %Y", errors='coerce')

# Fill remaining missing dates with a placeholder if necessary
# For example, using the median date
median_date = df['date_added'].median()
df['date_added'].fillna(median_date, inplace=True)
```

```
df.dtypes
```

```
show_id                 object
type                  category
title                   object
director                object
cast                    object
country               category
date_added      datetime64[ns]
release_year             int64
rating                category
duration                object
listed_in             category
description             object
dtype: object
```

Explore the data(EDA)

```
# Summary statistics
df.describe()
```

	date_added	release_year
count	8807	8807.000000
mean	2019-05-23 15:05:29.873963776	2014.180198
min	2008-01-01 00:00:00	1925.000000
25%	2018-04-30 12:00:00	2013.000000
50%	2019-07-12 00:00:00	2017.000000
75%	2020-08-18 00:00:00	2019.000000
max	2021-09-25 00:00:00	2021.000000
std	NaN	8.819312

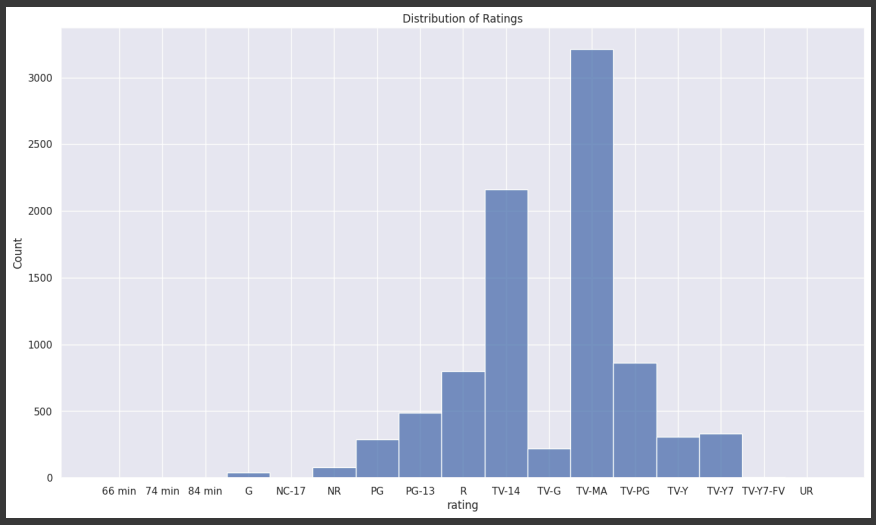
3. Non-Graphical Analysis: Value Counts and Unique Attributes (10 Points)

```
# Value counts for categorical attributes
print("\nType value counts:\n", df['type'].value_counts())
print("\nRating value counts:\n", df['rating'].value_counts())
print("\nCountry value counts (Top 10):\n", df['country'].value_counts().head(10))
print("\nGenre value counts (Top 10):\n", df['listed_in'].value_counts().head(10))

# Unique attributes
print("\nUnique types:", df['type'].unique())
print("\nUnique ratings:", df['rating'].unique())
print("\nUnique countries:", df['country'].unique())
print("\nUnique genres:", df['listed_in'].unique())
```

```
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

Country value counts (Top 10):
 country
United States     2818
India              972
Unknown            831
United Kingdom     419
Japan              245
South Korea        199
Canada             181
Spain              145
France             124
Mexico             110
Name: count, dtype: int64

Genre value counts (Top 10):
 listed_in
Dramas, International Movies                        362
Documentaries                                       359
Stand-Up Comedy                                     334
Comedies, Dramas, International Movies              274
```

```
                              'Argentina, Brazil, France, Poland, Germany, D..., ..., 'Vietnam', 'West Germany', 'Zimbabwe', 'Unkno

Unique genres: ['Documentaries', 'International TV Shows, TV Dramas, TV Mysteries', 'Crime TV Shows, International TV Shows, TV
Length: 514
Categories (514, object): ['Action & Adventure', 'Action & Adventure, Anime Features',
                           'Action & Adventure, Anime Features, Children ..., 'Action & Adventure, Anime Features, Classic M...,
                           ..., 'TV Horror, Teen TV Shows',
                           'TV Sci-Fi & Fantasy, TV Thrillers', 'TV Shows', 'Thrillers']
```

```
# Plot the distribution of a numerical column
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme(rc={'figure.figsize':(16,9)})
sns.histplot(df['rating'])
plt.title('Distribution of Ratings')
plt.show()
```




```
# Value counts for categorical attributes
print(df['type'].value_counts())
print(df['rating'].value_counts())
print(df['country'].value_counts().head(10))

# Unique values
print(df.nunique())
```

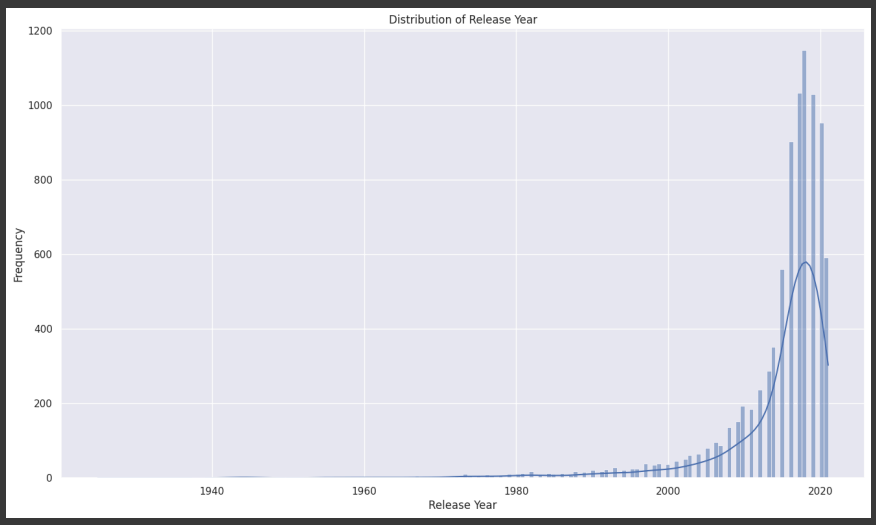
```
type
Movie      6131
TV Show    2676
Name: count, dtype: int64
rating
TV-MA       3211
TV-14       2160
TV-PG        863
R            799
PG-13        490
TV-Y7        334
TV-Y         307
PG           287
TV-G         220
NR            80
G             41
TV-Y7-FV       6
UR             3
NC-17          3
74 min         1
84 min         1
```

```
66 min         1
Name: count, dtype: int64
country
United States     2818
India              972
Unknown            831
United Kingdom     419
Japan              245
South Korea        199
Canada             181
Spain              145
France             124
Mexico             110
Name: count, dtype: int64
show_id         8807
type               2
title           8807
director        4529
cast            7693
country          749
date_added      1699
release_year      74
rating            17
duration         221
listed_in        514
description     8775
dtype: int64
```

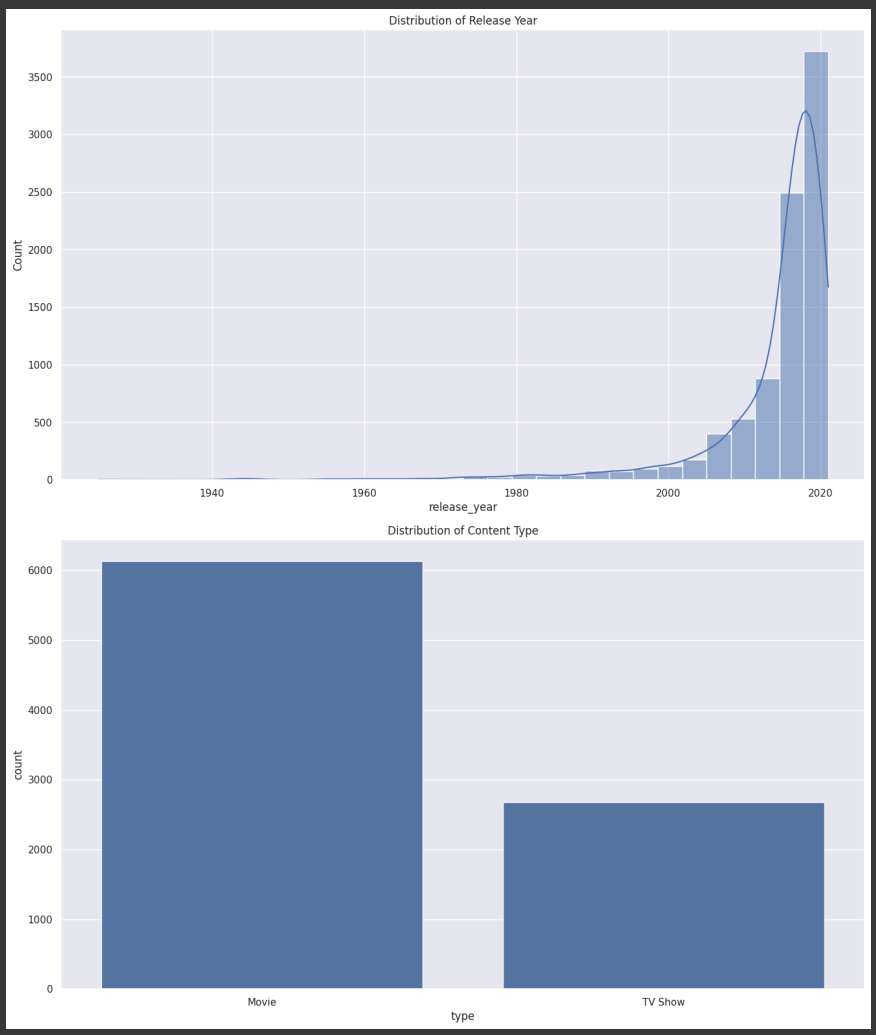
4. Visual Analysis - Univariate, Bivariate after Pre-processing (30 Points)

4.1 For Continuous Variables:

```
# Univariate Analysis for Continuous Variables
sns.histplot(df['release_year'], kde=True)
plt.title('Distribution of Release Year')
plt.xlabel('Release Year')
plt.ylabel('Frequency')
plt.show()
```

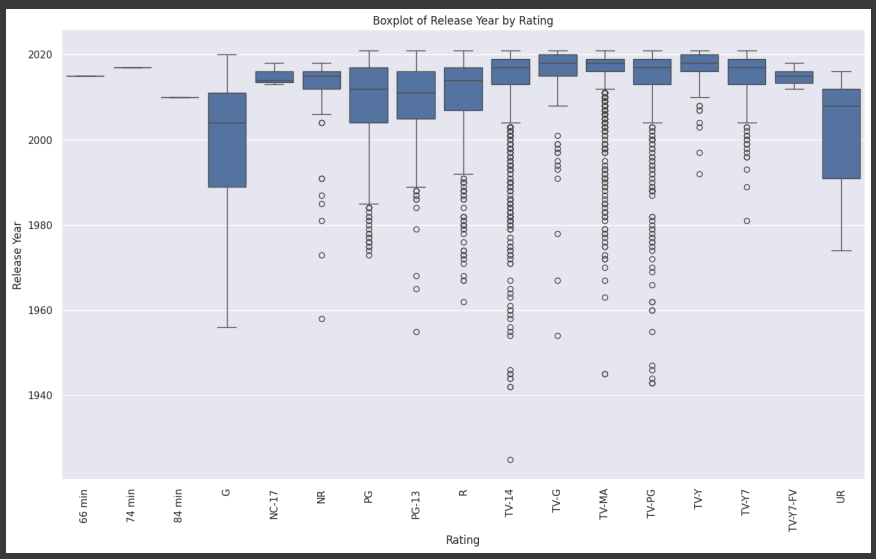


```
import seaborn as sns
import matplotlib.pyplot as plt

# Histogram for release_year
sns.histplot(df['release_year'], bins=30, kde=True)
plt.title('Distribution of Release Year')
plt.show()

# Countplot for type
sns.countplot(x='type', data=df)
plt.title('Distribution of Content Type')
plt.show()
```



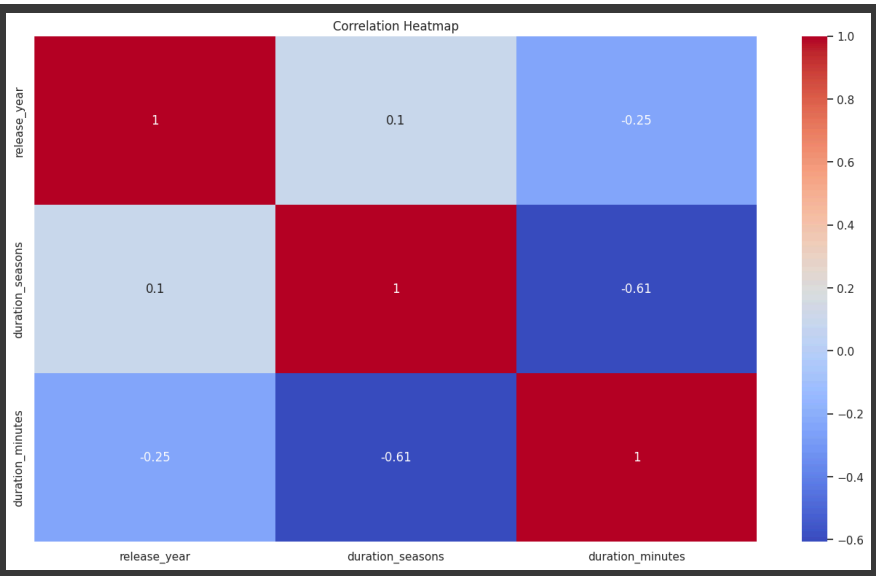
4.2 For Categorical Variables:

```
# Boxplot for Categorical Variables
sns.boxplot(x='rating', y='release_year', data=df)
plt.title('Boxplot of Release Year by Rating')
plt.xlabel('Rating')
plt.ylabel('Release Year')
plt.xticks(rotation=90)
plt.show()
```



4.3 For Correlation:

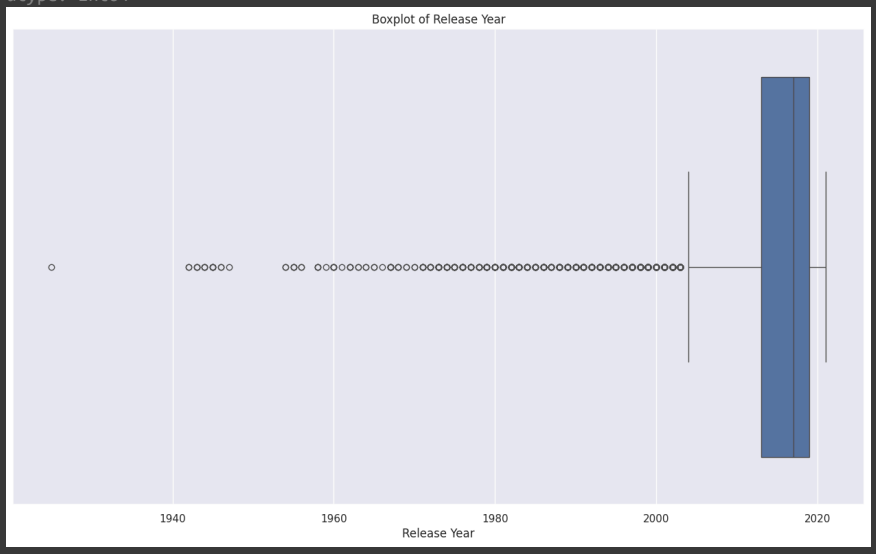
```
# Preprocess duration column
df['duration_seasons'] = df['duration'].apply(lambda x: int(x.split(' ')[0]) if 'Season' in x else 0)
df['duration_minutes'] = df['duration'].apply(lambda x: int(x.split(' ')[0]) if 'min' in x else 0)

# Select numeric columns only
numeric_df = df[['release_year', 'duration_seasons', 'duration_minutes']]

# Compute the correlation matrix
corr = numeric_df.corr()

# Plot the correlation heatmap
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



5. Missing Value & Outlier Check (10 Points)

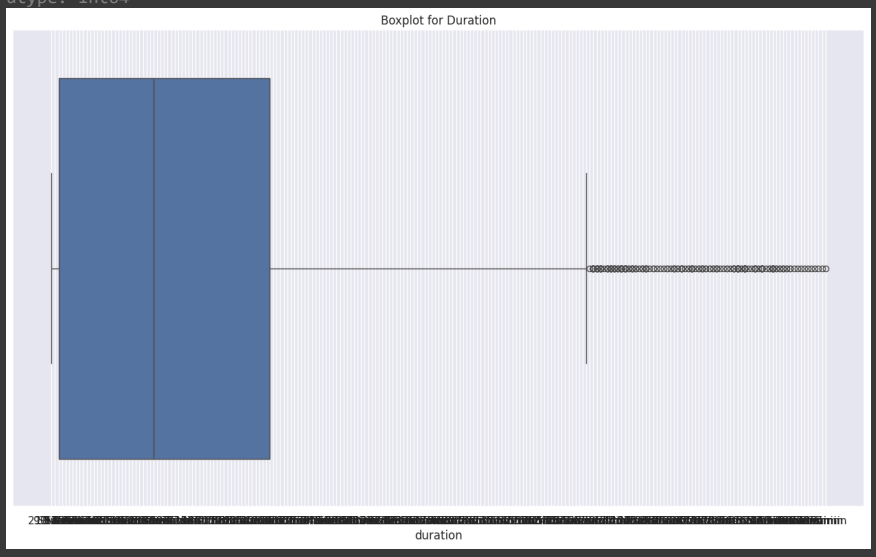
```
# Check for missing values
print("Missing values:\n", df.isnull().sum())

# Outlier detection
sns.boxplot(x=df['release_year'])
plt.title('Boxplot of Release Year')
plt.xlabel('Release Year')
plt.show()
```

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




```
# Missing values check
print(df.isnull().sum())

# Outlier check using boxplot for duration
sns.boxplot(x=df['duration'])
plt.title('Boxplot for Duration')
plt.show()
```

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



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

Comments and Observations:

6.1 Range of Attributes:

Release Year: Range from early 1900s to recent years. Duration: Movies have a wide range of durations, while TV shows indicate seasons, with some shows having multiple seasons while others have durations in minutes.

6.2 Distribution and Relationships:

Release Year Distribution: More recent years have higher counts.Skewed towards more recent years. Rating Distribution: Some ratings are more prevalent than others, such as TV-MA and TV-14. Content Type Distribution: Movies are more prevalent than TV shows. Duration Distribution: Wide range for movies, clustered around fewer seasons for TV shows. Duration Boxplot: Reveals outliers and typical range for movies/TV shows.

6.3 Comments on Plots:

Release Year Histogram: Indicates growth in content over the years. Type Countplot: Shows the dominance of movies. Univariate Plot for Release Year: Shows a peak in recent years indicating more content is produced recently. Boxplot of Release Year by Rating: Reveals that certain ratings are more common in specific periods.

7. Business Insights (10 Points)