Business Case: Netflix - Digital Media Analysis

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.

1. Defining Problem Statement and Analysing basic metrics.

Let us download the file

```
!wget -O netflix_data.csv "https://drive.google.com/uc?export=download&id=1dT8oqejnP0xfnE-_67P0BYQUaklJLYJS"
```

Show hidden output

import necessary libraries

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

Problem Statement : The problem is to analyze and gain insights from a Netflix dataset in order to understand user behavior, content preferences, and trends. This analysis will help Netflix make data-driven decisions to improve user engagement, content creation, and overall user satisfaction.

Load the data

```
df = pd.read_csv("netflix_data.csv")
```

Analysing basic metrics: 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 data set consists of 8807 titles and 12 columns. Visability of NaN values and data consists of two types of plays Movie and TV Show. The duration for Movie is given in min and for TV Show it is in Season or Seasons.

```
df.columns
```

Show hidden output

```
df.index
```

Show hidden output

```
df
```

Show hidden output

```
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 Orleans	NaN	NaN	NaN	September 24, 2021	2021
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021

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.

Shape of the data

ChatGPT

```
df.shape, df.ndim
```

```
((8807, 12), 2)
```

Data types of attributes

```
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        8800 non-null   object
 9   duration      8807 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
```

Conversion of categorical attributes to category

```
# Memory efficient and statistical modelling
categorical_attributes = ['type', 'country', 'rating']
df[categorical_attributes] = df[categorical_attributes].astype('category')
```

```
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   category
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7982 non-null   object
 5   country       7976 non-null   category
 6   date_added    8797 non-null   object
 7   release_year  8807 non-null   int64
 8   rating        8800 non-null   category
 9   duration      8807 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: category(3), int64(1), object(8)
memory usage: 676.6+ KB
```

Missing value Detection

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

```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        10
release_year       0
rating             7
duration           0
listed_in          0
description        0
dtype: int64
```

```
df.duplicated().sum()
```

```
0
```

General Statistical Summary

```
df.describe(include = 'all')
```

	show_id	type	title	director	cast	country	date_added	release_yea
count	8807	8807	8807	6173	7982	7976	8797	8807.0000(
unique	8807	2	8804	4528	7692	748	1767	Na
top	s1	Movie	15-Aug	Rajiv Chilaka	David Attenborough	United States	January 1, 2020	Na
freq	1	6131	2	19	19	2818	109	Na
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2014.18019
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	8.81931
min	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1925.0000(
25%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2013.0000(

```
df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	r
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	NaN	September 24, 2021	2021	

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

```
columns = ['type', 'director', 'country', 'release_year', 'rating', 'duration']
for column in columns:
  print(f"column : {column}")
  print("unique attributes" "   " "value counts")
  print(df[column].value_counts())
  print("Number of unique values or attribute count:", df[column].nunique())
  print("-----------------------------------------------")
```

```
column : type
unique attributes   value counts
Movie      6131
TV Show    2676
Name: type, dtype: int64
Number of unique values or attribute count: 2
-----------------------------------------------
column : director
unique attributes   value counts
Rajiv Chilaka                     19
RaÃƒÂºl Campos, Jan Suter         18
Marcus Raboy                      16
Suhas Kadav                       16
Jay Karas                         14
                                  ..
Raymie Muzquiz, Stu Livingston     1
```

```
Joe Menendez                        1
Eric Bross                          1
Will Eisenberg                      1
Mozez Singh                         1
Name: director, Length: 4528, dtype: int64
Number of unique values or attribute count: 4528
-------------------------------------------------
column : country
unique attributes   value counts
United States                                                                  2818
India                                                                           972
United Kingdom                                                                  419
Japan                                                                           245
South Korea                                                                     199
                                                                               ...
Ireland, Canada, Luxembourg, United States, United Kingdom, Philippines, India    1
Ireland, Canada, United Kingdom, United States                                    1
Ireland, Canada, United States, United Kingdom                                    1
Ireland, France, Iceland, United States, Mexico, Belgium, United Kingdom, Hong Kong    1
Zimbabwe                                                                          1
Name: country, Length: 748, dtype: int64
Number of unique values or attribute count: 748
-------------------------------------------------
column : release_year
unique attributes   value counts
2018    1147
2017    1032
2019    1030
2020     953
2016     902
        ...
1959       1
1925       1
1961       1
1947       1
1966       1
Name: release_year, Length: 74, dtype: int64
Number of unique values or attribute count: 74
-------------------------------------------------
column : rating
unique attributes   value counts
TV-MA       3207
```

If we notice the above output, the country attributes are in more number because of presence multiple values in country rows. Let us split them to get unique attributes and value counts.

```
individual_countries = df['country'].str.split(', ', expand = True).stack()
print("unique attributes" "  " "value counts")
individual_countries.value_counts()
```

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

Data pre-processing

uniquely separating the directors, cast, country, listed_in

```
import pandas as pd

# Read the CSV file
df = pd.read_csv("netflix_data.csv")

# Unnesting using explode (list of arrays in each row gets flattened)
df_list = df.copy()

df_list['cast'] = df_list['cast'].str.split(', ')
df_list = df_list.explode('cast')

df_list['director'] = df_list['director'].str.split(', ')
df_list = df_list.explode('director')

df_list['country'] = df_list['country'].str.split(', ')
df_list = df_list.explode('country')

df_list['listed_in'] = df_list['listed_in'].str.split(', ')
df_list = df_list.explode('listed_in')

# Display the modified dataframe
print(df_list.head())
```

```
  show_id     type                 title         director         cast  \
0      s1    Movie  Dick Johnson Is Dead  Kirsten Johnson          NaN
1      s2  TV Show         Blood & Water              NaN   Ama Qamata
1      s2  TV Show         Blood & Water              NaN   Ama Qamata
1      s2  TV Show         Blood & Water              NaN   Ama Qamata
1      s2  TV Show         Blood & Water              NaN  Khosi Ngema

         country          date_added  release_year rating   duration  \
0  United States  September 25, 2021          2020  PG-13     90 min
1   South Africa  September 24, 2021          2021  TV-MA  2 Seasons
1   South Africa  September 24, 2021          2021  TV-MA  2 Seasons
1   South Africa  September 24, 2021          2021  TV-MA  2 Seasons
1   South Africa  September 24, 2021          2021  TV-MA  2 Seasons

                listed_in                                        description
0           Documentaries  As her father nears the end of his life, filmm...
1  International TV Shows  After crossing paths at a party, a Cape Town t...
1              TV Dramas  After crossing paths at a party, a Cape Town t...
1            TV Mysteries  After crossing paths at a party, a Cape Town t...
1  International TV Shows  After crossing paths at a party, a Cape Town t...
```

```
# Convert 'date_added' to datetime
df_list['date_added'] = pd.to_datetime(df_list['date_added'])

# Split 'date_added' into 'added_year' and 'added_month'
df_list['added_year'] = df_list['date_added'].dt.year
df_list['added_month'] = df_list['date_added'].dt.month
df_list['added_day'] = df_list['date_added'].dt.day

# Drop the original 'date_added' column
df_list.drop(columns=['date_added'], inplace=True)

# Display the first few rows of the DataFrame to verify changes
print(df_list.head())
```

```
  show_id     type                 title         director         cast  \
0      s1    Movie  Dick Johnson Is Dead  Kirsten Johnson          NaN
1      s2  TV Show         Blood & Water              NaN   Ama Qamata
1      s2  TV Show         Blood & Water              NaN   Ama Qamata
1      s2  TV Show         Blood & Water              NaN   Ama Qamata
1      s2  TV Show         Blood & Water              NaN  Khosi Ngema
```

```
         country  release_year rating   duration               listed_in  \
0  United States          2020  PG-13     90 min           Documentaries
1   South Africa          2021  TV-MA  2 Seasons  International TV Shows
1   South Africa          2021  TV-MA  2 Seasons              TV Dramas
1   South Africa          2021  TV-MA  2 Seasons            TV Mysteries
1   South Africa          2021  TV-MA  2 Seasons  International TV Shows

                                        description  added_year  added_month  \
0  As her father nears the end of his life, filmm...      2021.0          9.0
1  After crossing paths at a party, a Cape Town t...      2021.0          9.0
1  After crossing paths at a party, a Cape Town t...      2021.0          9.0
1  After crossing paths at a party, a Cape Town t...      2021.0          9.0
1  After crossing paths at a party, a Cape Town t...      2021.0          9.0

   added_day
0       25.0
1       24.0
1       24.0
1       24.0
1       24.0
```

```
# Replace 'min' in duration with the numeric value
df_list['duration'] = df_list['duration'].str.extract('(\d+)')

# For entries with 'Seasons', replace with the numeric value
df_list.loc[df_list['type'] == 'TV Show', 'duration'] = df_list.loc[df_list['type'] == 'TV Show', 'duration'

# Convert the 'duration' column to numeric
df_list['duration'] = pd.to_numeric(df_list['duration'], errors='coerce')
```

```
df_list.head()
```

	show_id	type	title	director	cast	country	release_year	rating	duration
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2020	PG-13	90
1	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	2021	TV-MA	2
1	s2	TV Show	Blood & Water	NaN	Ama Qamata	South Africa	2021	TV-MA	2

```
df_list.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 201991 entries, 0 to 8806
Data columns (total 14 columns):
 #   Column        Non-Null Count   Dtype
---  ------        --------------   -----
 0   show_id       201991 non-null  object
 1   type          201991 non-null  object
 2   title         201991 non-null  object
 3   director      151348 non-null  object
 4   cast          199845 non-null  object
 5   country       190094 non-null  object
 6   release_year  201991 non-null  int64
 7   rating        201921 non-null  object
 8   duration      201991 non-null  int64
```

```
 9   listed_in     201991 non-null  object
 10  description   201991 non-null  object
 11  added_year    201833 non-null  float64
 12  added_month   201833 non-null  float64
 13  added_day     201833 non-null  float64
dtypes: float64(3), int64(2), object(9)
memory usage: 23.1+ MB
```

```
df_list.shape
```

```
(201991, 14)
```

```
df_movies = df_list[df_list['type'] == 'Movie']
df_movies.shape
```

```
(145843, 14)
```

```
df_tv_shows = df_list[df_list['type'] == 'TV Show']
df_tv_shows.shape
```

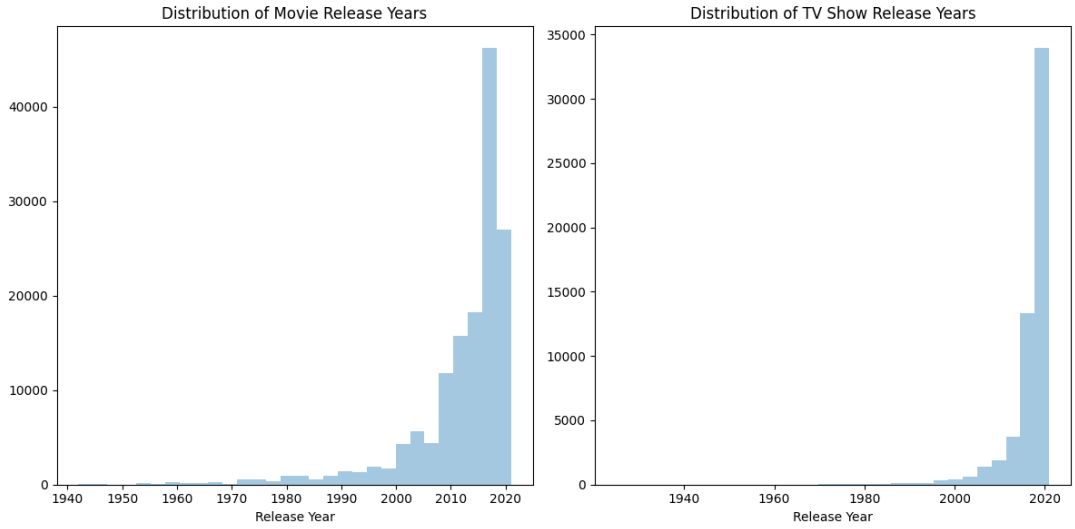
```
(56148, 14)
```

4. Visual Analysis - Univariate, Bivariate after pre-processing of the data

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

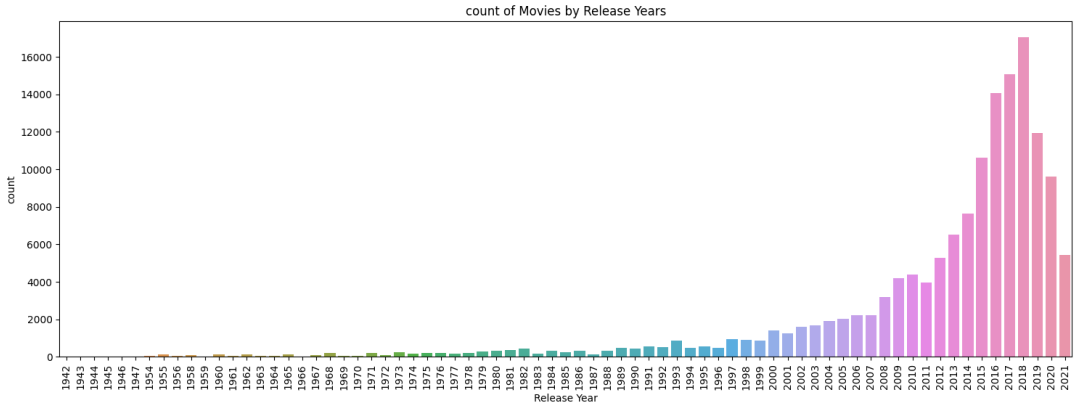
```
# Filter the DataFrame for movies and TV shows
df_movies = df_list[df_list['type'] == 'Movie']
df_tv_shows = df_list[df_list['type'] == 'TV Show']

# Create a figure with two subplots
plt.figure(figsize=(12, 6))

# Subplot 1: Distplot for movie release years
plt.subplot(1, 2, 1)
sns.distplot(df_movies['release_year'], bins=30, kde=False)
plt.title('Distribution of Movie Release Years')
plt.xlabel('Release Year')

# Subplot 2: Distplot for TV show release years
plt.subplot(1, 2, 2)
sns.distplot(df_tv_shows['release_year'], bins=30, kde=False)
plt.title('Distribution of TV Show Release Years')
plt.xlabel('Release Year')

plt.tight_layout()
plt.show()
```

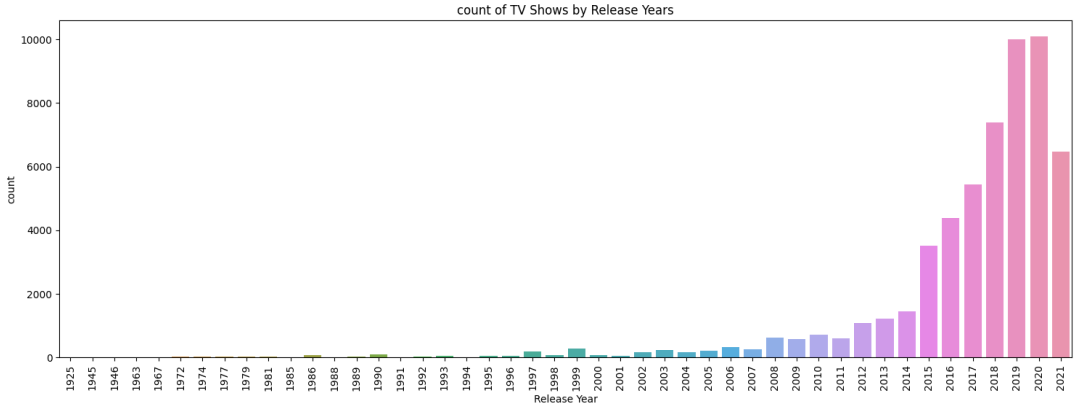
```
<ipython-input-26-ef58b9ad5bd5>:10: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_movies['release_year'], bins=30, kde=False)
<ipython-input-26-ef58b9ad5bd5>:16: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_tv_shows['release_year'], bins=30, kde=False)
```



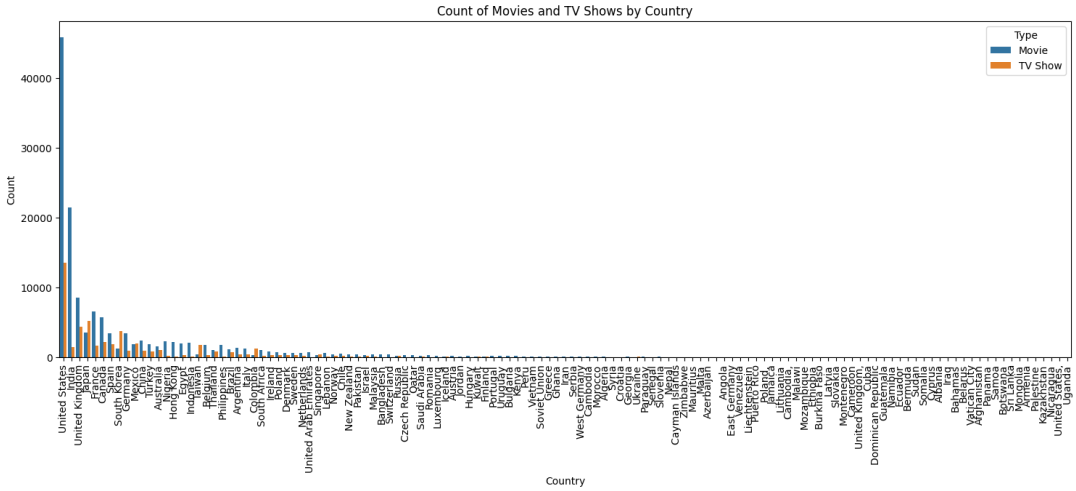
Count plot

```
plt.figure(figsize=(18, 6))

# countplot for movie release years
sns.countplot(data = df_movies, x = 'release_year')
plt.title('count of Movies by Release Years')
plt.xlabel('Release Year')
plt.xticks(rotation=90)
plt.show()

plt.figure(figsize=(18, 6))
#countplot for TV show release years

sns.countplot(data = df_tv_shows, x = 'release_year')
plt.title('count of TV Shows by Release Years')
plt.xlabel('Release Year')
plt.xticks(rotation=90)
plt.show()
```

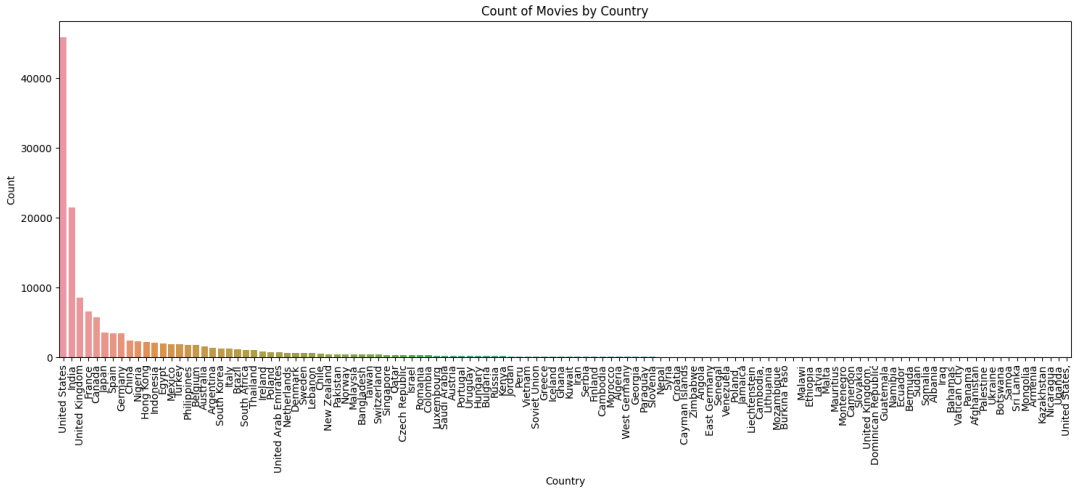


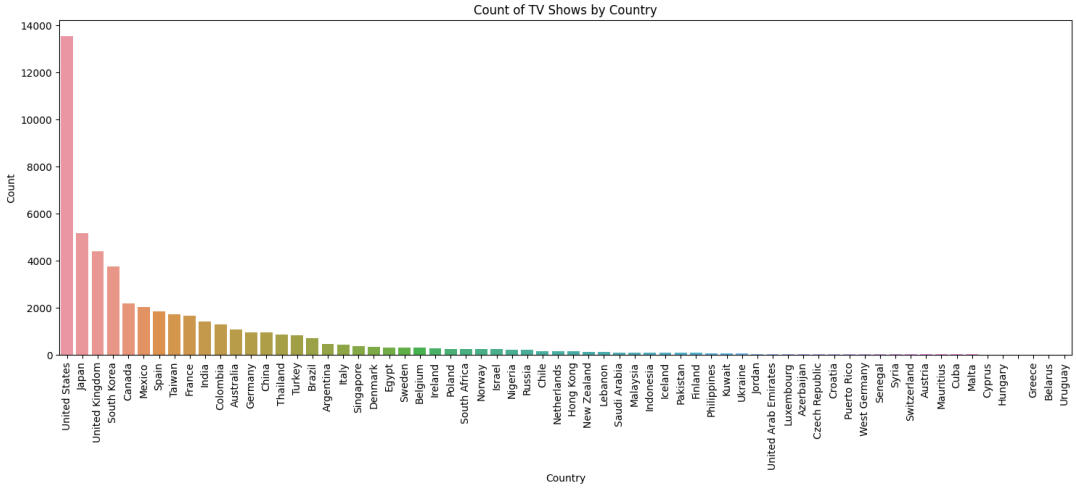



```
plt.figure(figsize=(18, 6))

# countplot for movie and TV show counts by country
sns.countplot(data=df_list, x='country', hue='type', order=df_list['country'].value_counts().index)
plt.title('Count of Movies and TV Shows by Country')
plt.xlabel('Country')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.legend(title='Type')
plt.show()
```



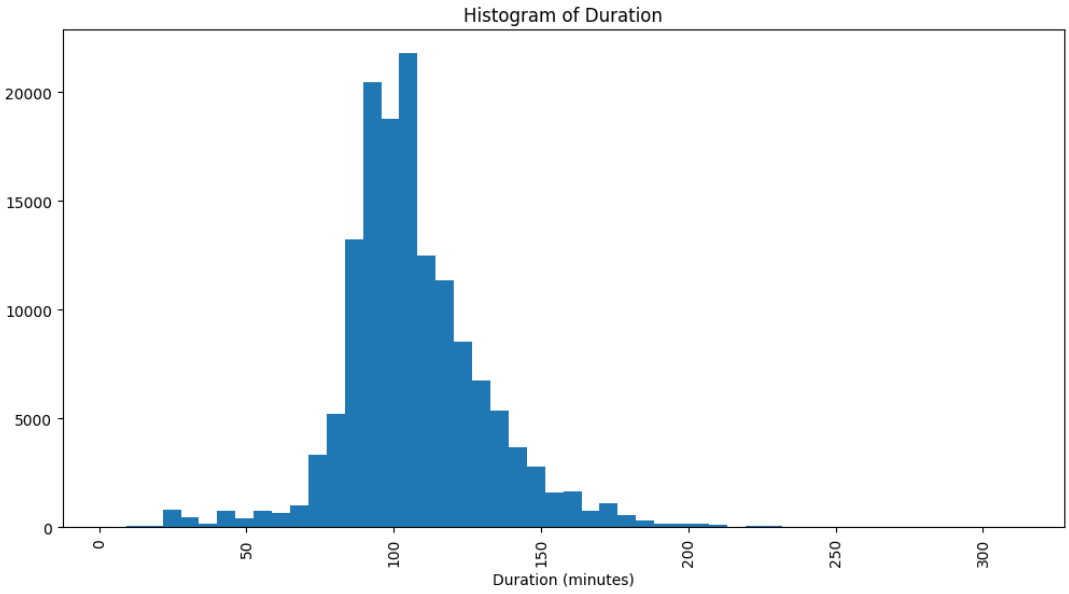

```
plt.figure(figsize=(18, 6))

# countplot for movie directors
sns.countplot(data=df_movies, x='country', order=df_movies['country'].value_counts().index)
plt.title('Count of Movies by Country')
plt.xlabel('Country')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()

plt.figure(figsize=(18, 6))
# countplot for TV show directors
sns.countplot(data=df_tv_shows, x='country', order=df_tv_shows['country'].value_counts().index)
plt.title('Count of TV Shows by Country')
plt.xlabel('Country')
plt.ylabel('Count')
plt.xticks(rotation=90)
plt.show()
```

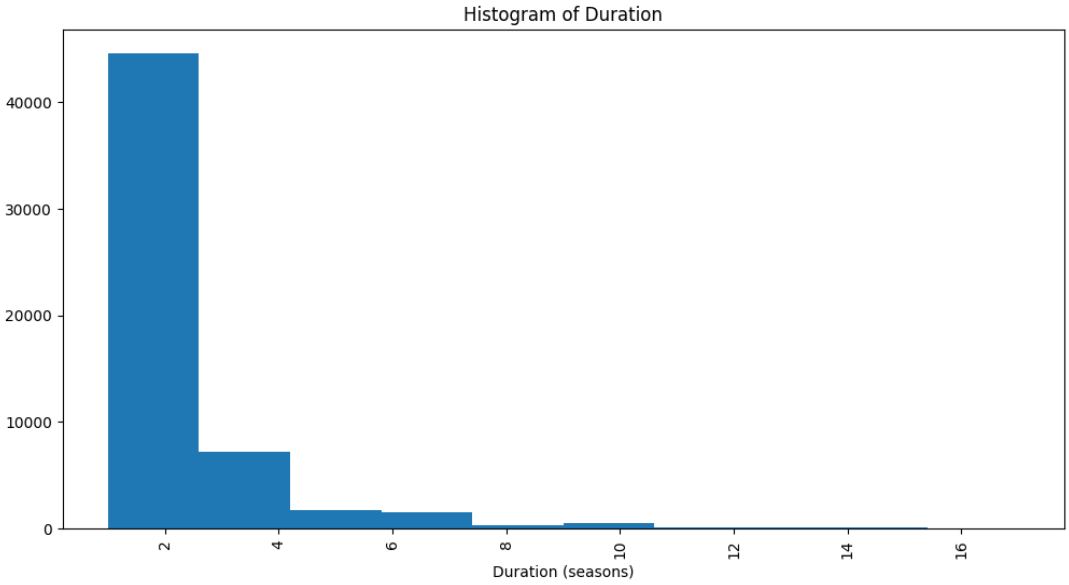




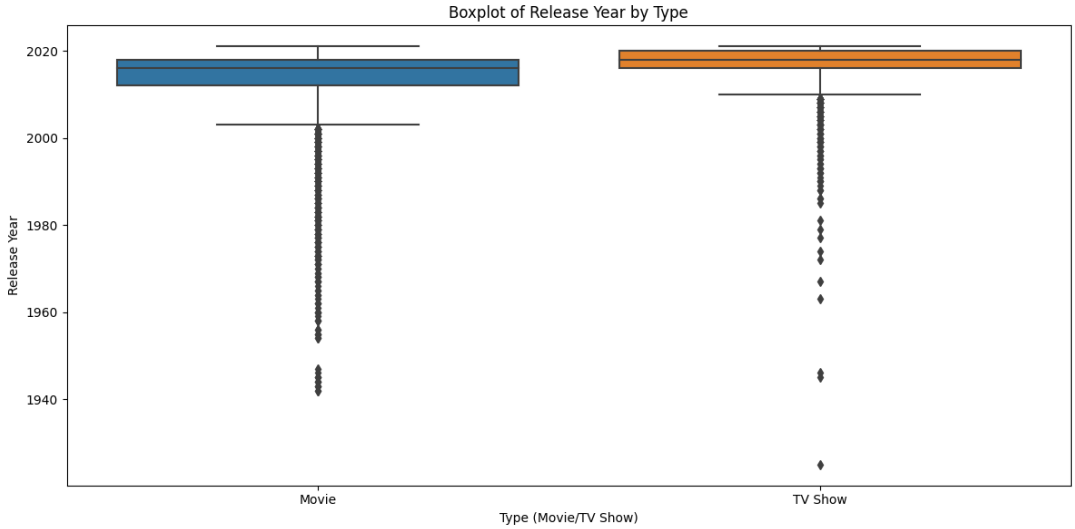
Histogram

```
plt.figure(figsize=(12, 6))

# movies
plt.hist(df_movies['duration'], bins = 50)
plt.title('Histogram of Duration')
plt.xlabel('Duration (minutes)')
plt.xticks(rotation=90)
plt.show()

plt.figure(figsize=(12, 6))

# TV shows
plt.hist(df_tv_shows['duration'], bins = 10)
plt.title('Histogram of Duration')
plt.xlabel('Duration (seasons)')
plt.xticks(rotation=90)
plt.show()
```


Histogram of Duration
20000
15000
10000
5000
0
0
50
100
150
200
250
300
Duration (minutes)

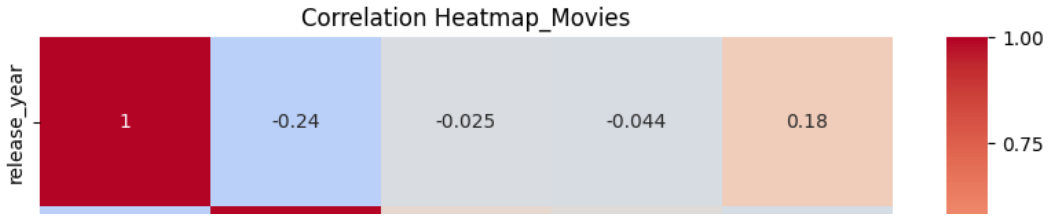

Histogram of Duration
40000
30000
20000
10000
0
2
4
6
8
10
12
14
16
Duration (seasons)

```
# Remove duplicate entries based on release_year and title for movies
df_list_movies_unique = df_list[df_list['type'] == 'Movie'].drop_duplicates(subset=['release_year', 'title']

# Remove duplicate entries based on release_year and title for TV shows
df_list_tv_shows_unique = df_list[df_list['type'] == 'TV Show'].drop_duplicates(subset=['release_year', 'tit

# Combine the unique movies and TV shows DataFrames
df_combined_unique = pd.concat([df_list_movies_unique, df_list_tv_shows_unique])

# Set up the figure and subplots
plt.figure(figsize=(12, 6))

# Boxplot for type vs. release_year

sns.boxplot(data=df_combined_unique, x='type', y='release_year')
plt.title('Boxplot of Release Year by Type')
plt.xlabel('Type (Movie/TV Show)')
plt.ylabel('Release Year')


plt.tight_layout()
plt.show()
```



```
plt.figure(figsize=(10, 8))
correlation_matrix = df_list[df_list['type'] == 'Movie'].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Heatmap_Movies')
plt.show()


plt.figure(figsize=(10, 8))
correlation_matrix = df_list[df_list['type'] == 'TV Show'].corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Heatmap_TV_show')
plt.show()
```

```
<ipython-input-32-3ce285f4760e>:2: FutureWarning: The default value of numeric_only i
  correlation_matrix = df_list[df_list['type'] == 'Movie'].corr()
```



Pairplot

```
df_list_reset = df_list.reset_index(drop=True)
sns.pairplot(df_list_reset, hue = 'type', palette = 'bright')
plt.title('Pair Plot by type')
plt.show()
```

5. Missing Value & Outlier check (Treatment optional)

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

```
show_id            0
type               0
title              0
director        2634
cast             825
country          831
date_added        10
release_year       0
rating             7
duration           0
listed_in          0
description        0
dtype: int64
```

Filling missing values of country based on directors

```
# Group the data by 'director' and find the most frequent country for each group
director_country_mapping = df[df['country'].notnull()].groupby('director')['country'].apply(lambda x: x.mode

# Replace missing country values for directors with a valid mapping
df['country'] = df.apply(lambda row: director_country_mapping[row['director']] if pd.isnull(row['country'])

# Replace missing country values for directors without a valid mapping using the mode of the entire 'country
mode_country = df['country'].mode().iloc[0]
df['country'] = df['country'].fillna(mode_country)
```

```
import numpy as np

# Group the data by 'director' and find the most frequent rating for each group
director_rating_mapping = df[df['rating'].notnull()].groupby('director')['rating'].apply(lambda x: x.mode().

# Replace missing rating values for directors with a valid mapping
df['rating'] = df.apply(lambda row: director_rating_mapping[row['director']] if pd.isnull(row['rating']) and

# Replace missing rating values for directors without a valid mapping using the mode of the entire 'rating'
mode_rating = df['rating'].mode().iloc[0]
df['rating'] = df['rating'].fillna(mode_rating)
```

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

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

```
# Fill missing values in 'date_added' with the mode (most common) date
mode_date = df['date_added'].mode().iloc[0]
df['date_added'] = df['date_added'].fillna(mode_date)
```

```
# Fill missing values in 'director' and 'cast' columns
df['director'] = df['director'].fillna("Unknown")
df['cast'] = df['cast'].fillna("Unknown")
```

```
# Convert 'date_added' to datetime
df['date_added'] = pd.to_datetime(df['date_added'])

# Split 'date_added' into 'added_year' and 'added_month'
df['added_year'] = df['date_added'].dt.year
df['added_month'] = df['date_added'].dt.month
df['added_day'] = df['date_added'].dt.day

# Drop the original 'date_added' column
df.drop(columns=['date_added'], inplace=True)

# Display the first few rows of the DataFrame to verify changes
print(df.head())
```

```
  show_id     type                  title         director  \
0      s1    Movie   Dick Johnson Is Dead  Kirsten Johnson
1      s2  TV Show          Blood & Water          Unknown
2      s3  TV Show              Ganglands  Julien Leclercq
3      s4  TV Show  Jailbirds New Orleans          Unknown
4      s5  TV Show           Kota Factory          Unknown

                                          cast        country  \
0                                      Unknown  United States
1  Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...   South Africa
2  Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...         France
3                                      Unknown  United States
4  Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...          India

   release_year rating   duration  \
0          2020  PG-13     90 min
1          2021  TV-MA  2 Seasons
2          2021  TV-MA   1 Season
3          2021  TV-MA   1 Season
4          2021  TV-MA  2 Seasons

                                     listed_in  \
0                                Documentaries
1   International TV Shows, TV Dramas, TV Mysteries
2  Crime TV Shows, International TV Shows, TV Act...
3                          Docuseries, Reality TV
4  International TV Shows, Romantic TV Shows, TV ...

                                   description  added_year  added_month  \
0  As her father nears the end of his life, filmm...        2021            9
1  After crossing paths at a party, a Cape Town t...        2021            9
2  To protect his family from a powerful drug lor...        2021            9
3  Feuds, flirtations and toilet talk go down amo...        2021            9
4  In a city of coaching centers known to train I...        2021            9

   added_day
0         25
1         24
2         24
3         24
4         24
```

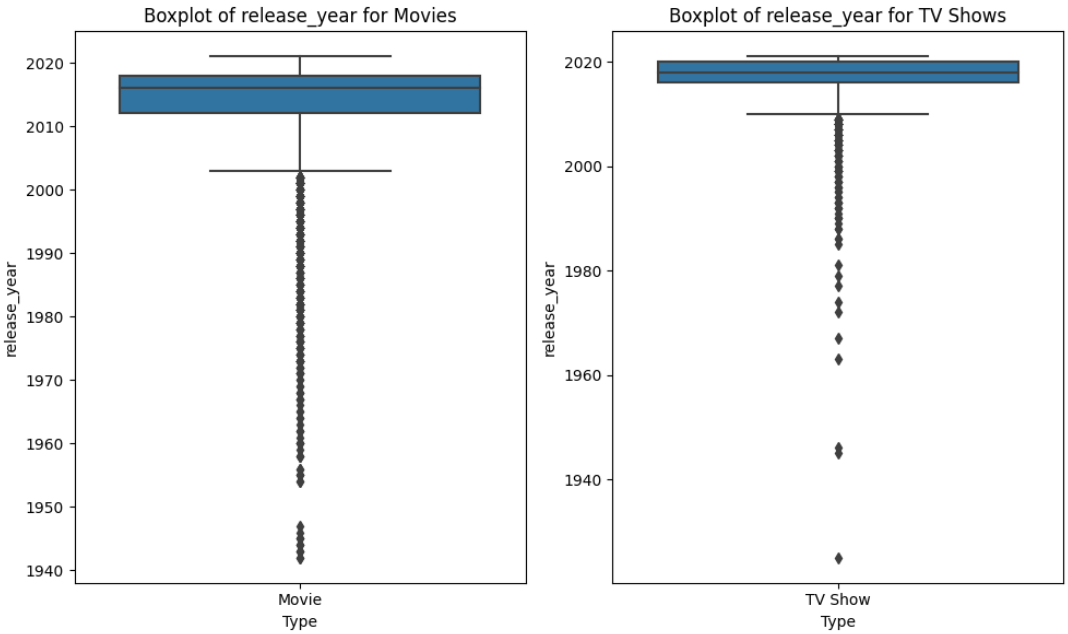
```
# Replace 'min' in duration with the numeric value
df['duration'] = df['duration'].str.extract('(\d+)')

# For entries with 'Seasons', replace with the numeric value
df.loc[df['type'] == 'TV Show', 'duration'] = df.loc[df['type'] == 'TV Show', 'duration'].str.replace(' Seas

# Convert the 'duration' column to numeric
df['duration'] = pd.to_numeric(df['duration'], errors='coerce')
```

Checking for null values

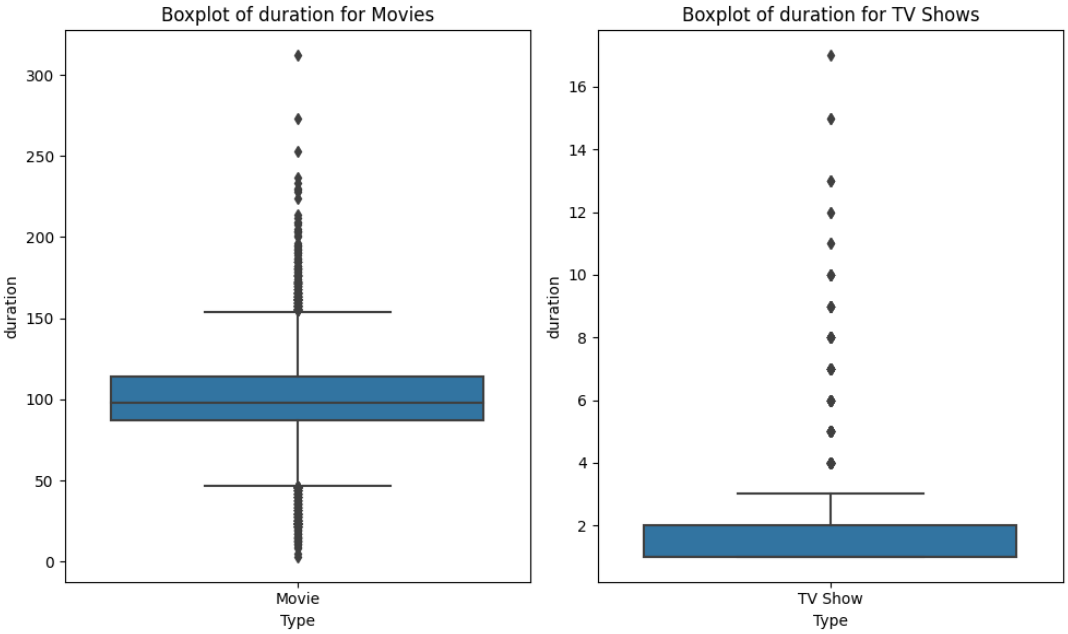
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 14 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      8807 non-null   object
 4   cast          8807 non-null   object
 5   country       8807 non-null   object
 6   release_year  8807 non-null   int64 
 7   rating        8807 non-null   object
 8   duration      8807 non-null   int64 
 9   listed_in     8807 non-null   object
 10  description   8807 non-null   object
 11  added_year    8807 non-null   int64 
 12  added_month   8807 non-null   int64 
 13  added_day     8807 non-null   int64 
dtypes: int64(5), object(9)
memory usage: 963.4+ KB
```

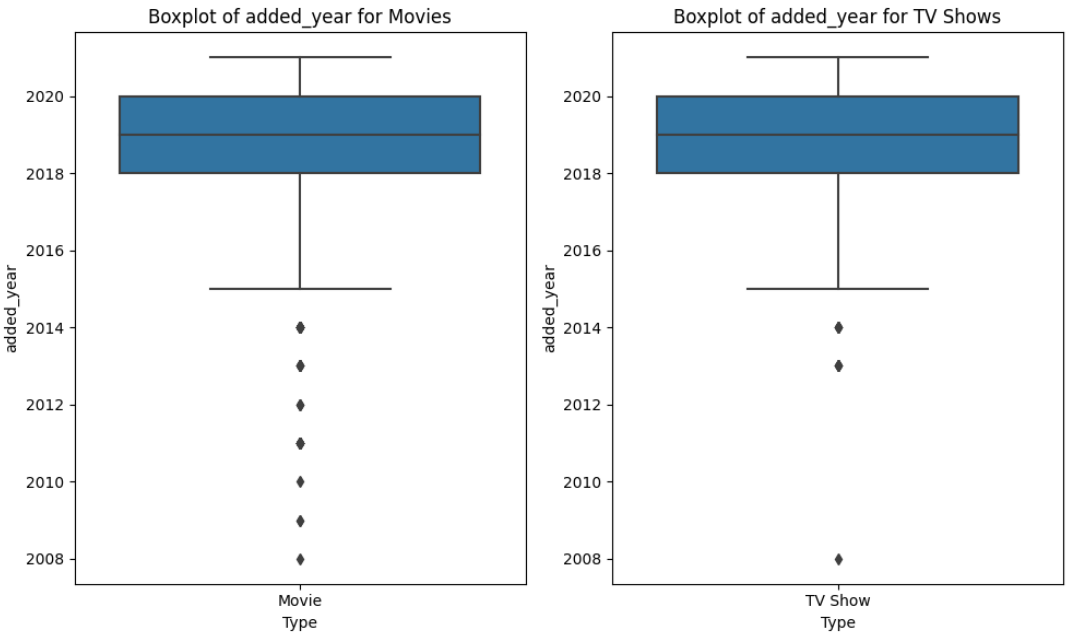
```
# Select the columns with continuous variables
continuous_columns = ['release_year', 'duration', 'added_year', 'added_month', 'added_day']

# Create separate box plots for each continuous variable based on the 'type' column
for column in continuous_columns:
    plt.figure(figsize=(10, 6))

    # Boxplot for Movies
    plt.subplot(1, 2, 1)
    sns.boxplot(data=df[df['type'] == 'Movie'], x='type', y=column)
    plt.title(f'Boxplot of {column} for Movies')
    plt.xlabel('Type')
    plt.ylabel(column)

    # Boxplot for TV Shows
    plt.subplot(1, 2, 2)
    sns.boxplot(data=df[df['type'] == 'TV Show'], x='type', y=column)
    plt.title(f'Boxplot of {column} for TV Shows')
    plt.xlabel('Type')
    plt.ylabel(column)

    plt.tight_layout()
    plt.show()
```


Boxplot of release_year for Movies
release_year
Movie
Type

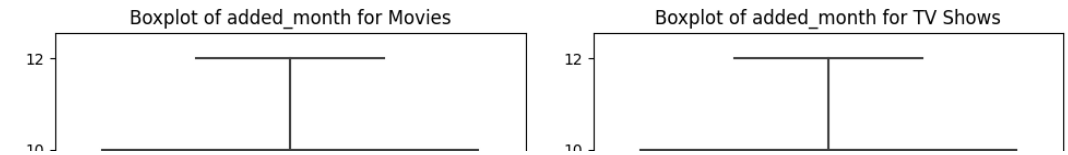
Boxplot of release_year for TV Shows
release_year
TV Show
Type


Boxplot of duration for Movies
duration
Movie
Type

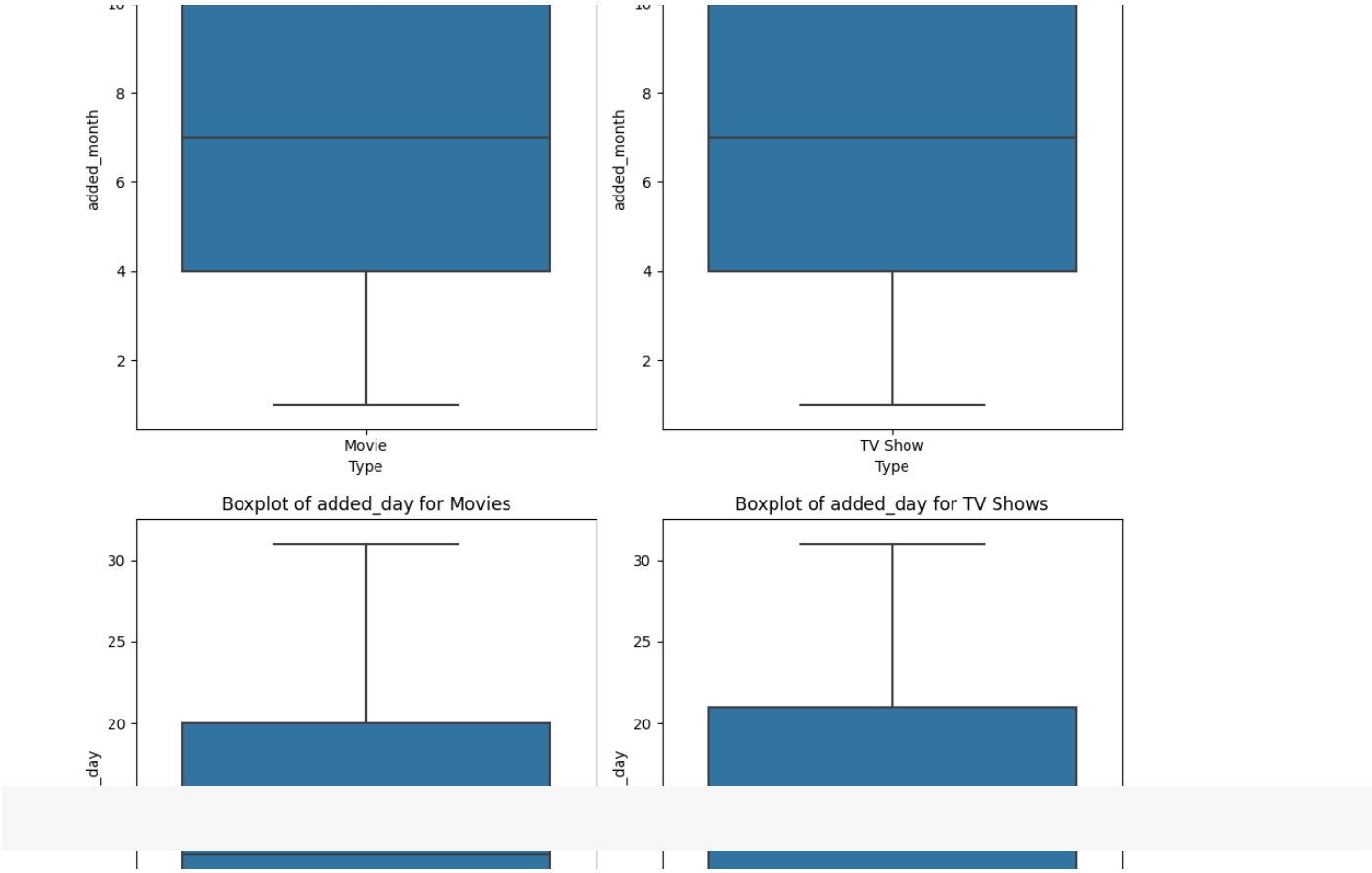
Boxplot of duration for TV Shows
duration
TV Show
Type


Boxplot of added_year for Movies
added_year
Movie
Type

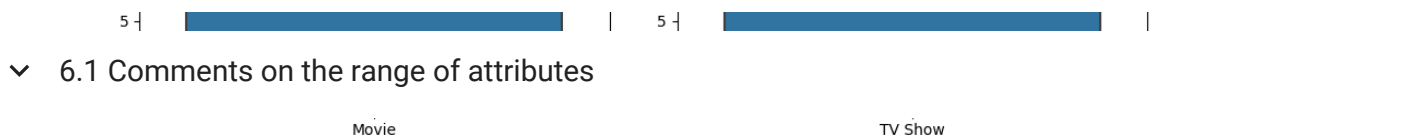
Boxplot of added_year for TV Shows
added_year
TV Show
Type


Boxplot of added_month for Movies

Boxplot of added_month for TV Shows



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



6.1 Comments on the range of attributes

- The dataset contains information about both movies and TV shows.
- Release years range from early 1900s to recent years.
- The duration of movies varies widely, from around 10 minutes to over 300 minutes.
- For TV shows, the duration is mostly in terms of seasons, ranging from 1 to 15 seasons.
- The "date_added" feature shows the dates when content was added to the platform.
- Ratings include various categories such as G, PG, R, TV-Y, TV-MA, etc.
- The dataset includes a variety of genres and descriptions for the content.

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

- The distribution of release years shows that there has been an increase in content over the years.
- The distribution of movie durations is skewed, with most movies having durations around 100 to 150 minutes.
- TV show durations (in terms of seasons) are also skewed, with 2 seasons being high.
- There seems to be a correlation between the year content was added and release year, suggesting that newer content is being added more frequently.
- The generation of movies higher in countries like United States, India, United Kingdom, etc Where as TV Shows are higher in United States, Japan, United Kingdom, etc.

6.3 Comments for each univariate and bivariate plot

- In the distplot of release years, there is a noticeable increase in content in recent years, indicating a growth in production.
- The countplot of the "type" variable shows that there are more movies than TV shows in the dataset.
- The histogram of duration for movies indicates that most movies have durations between 80 and 150 minutes.
- The histogram of duration for TV shows indicate most of them are of 2 number of seasons.
- The box plot for release years indicate most of the movies or shows are released between 2000 to 2020 and the others are in outliers.
- The heatmap indicates there is positive correlation between release year and added day for movies and realease year and added year for TV Shows.
- The pairplot offers insights into relationships between numerical attributes like "release_year," "added_year," and "duration."

7. Business Insights - Should include patterns observed in the data along with what you can infer from it

Some business insights derived from the data analysis, along with patterns observed and inferences drawn:

1. **Release Trends Over the Years:**
    - **Pattern:** The distribution of release years indicates an increasing trend in both movies and TV shows, with more content being produced in recent years.
    - **Inference:** The entertainment industry is experiencing growth, possibly due to increased demand for diverse content across different genres.
2. **Content Ratings Distribution:**
    - **Pattern:** The rating distribution suggests that 'TV-MA' (Mature Audience) is the most common rating for both movies and TV shows.
    - **Inference:** Content targeted at mature audiences is popular, possibly indicating a strong consumer base for such content.
3. **Duration Trends:**
    - **Pattern:** Movies typically have shorter durations, often around 90 minutes and same for TV shows commonly available in "Seasons" format.
    - **Inference:** Consumers have a preference for shorter movie durations, and same for TV shows.
4. **Country Insights:**
    - **Pattern:** The top countries producing content include the United States and India, indicating their dominance in the entertainment industry.
    - **Inference:** These countries likely have well-established entertainment markets and production capabilities.
5. **Director's Influence on Country and Rating:**
    - **Pattern:** Some directors are associated with specific countries and content ratings.
    - **Inference:** Certain directors may specialize in particular genres or themes that are popular in their respective countries, impacting content choices.
6. **Content Growth by Type:**
    - **Pattern:** The number of TV shows has been growing consistently, possibly reflecting the trend of increased episodic content consumption.
    - **Inference:** Consumers are gravitating towards serialized content, creating opportunities for more TV show production.
7. **Release Year vs. Duration Trends:**

- **Pattern:** There is no significant correlation between release year and duration for both movies and TV shows.
- **Inference:** Duration of content is not strongly influenced by release year, indicating that preferences for content length have remained consistent over time.

8. **Release Year vs. Rating Insights:**
   - **Pattern:** Ratings are spread across various release years, suggesting that ratings do not have a strong correlation with the time of release.
   - **Inference:** Ratings are not solely determined by the release era; content quality and thematic elements play a significant role.
9. **Content Distribution by Month:**
   - **Pattern:** Content addition is spread across the months, with some variations.
   - **Inference:** Streaming platforms maintain a consistent flow of new content throughout the year to engage users.
10. **Global Content Diversity:**
    - **Pattern:** The content spans a wide range of genres and themes.
    - **Inference:** Streaming platforms cater to diverse audience preferences, providing a mix of genres to attract a broader user base.

In summary, the data analysis reveals a dynamic entertainment landscape with trends favoring increased production, diverse content offerings, and an audience preference for both shorter movie formats and serialized TV shows. Directors and countries play a role in shaping content themes, while ratings and release years demonstrate the complexity of audience preferences. Streaming platforms are keen on maintaining a balanced release schedule to sustain user engagement throughout the year.

8. Recommendations - Actionable items for business. No technical jargon. No complications. Simple action items that everyone can understand

Some simple and actionable recommendations based on the data analysis:

1. **Focus on Diverse Genres:** Keep offering a wide variety of genres to cater to different viewer preferences. This can help attract a broader audience and keep them engaged.
2. **Invest in TV Show Production:** Since TV shows are gaining popularity, consider increasing the production of serialized content. This can capitalize on the trend of episodic content consumption.
3. **Quality Over Release Year:** Ratings are not solely determined by the release year. Prioritize content quality and thematic elements to ensure audience satisfaction.
4. **Tailor Content for Different Audiences:** Customize content based on ratings. Develop more content for mature audiences (TV-MA) as it appears to be popular.
5. **Consistent Release Schedule:** Maintain a balanced release schedule throughout the year. This can ensure a consistent flow of new content and keep users engaged.
6. **Shorter Movie Formats:** Offer a good mix of shorter movie formats (around 90 minutes) alongside longer ones. This can cater to different viewer preferences.