Business Case: Netflix – Data Exploration and Visualization

BATCH: DSML Apr23 Beginner Monday 2

Name: Akash Patel **Email:akp22061995@gmail.com**

About Netflix:

Netflix is an American service provider and media services provider and production company headquartered in Los Gatos, California. Netflix was founded in 1997 by Reed Hastings and Marc Randolph in Scotts Valley, California. The company's primary business is its subscription-based streaming service, which offers online streaming of a library of films and television series, including those produced in house.

Business Problem:

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

1. Defining Problem Statement and Analysing basic metrics.

Importing Libraries

import numpy as np

import pandas as pd

import matlpotlib.pyplot as plt

import seaborn as sns

Upload the Netflix_titles.CSV file in google Colab

```
from google.colab import files

uploaded = files.upload()
```

Choose Files netflix_titles.csv

- **netflix_titles.csv**(text/csv) - 3399671 bytes, last modified: 10/2/2023 - 100% done

```
Saving netflix_titles.csv to netflix_titles.csv
```

Loading the Dataset

Using Pandas Library, we will load the CSV file, named it with Netflix_df for the dataset.

Netflix_df = pd.read_csv("netfilx_titles.csv")

Analysing basic Metrics

Number of TV Shows and Movies on the Netflix

```
netflix_df["type"].value_counts()
```

```
Movie        6131
TV Show      2676
Name: type, dtype: int64
```

Number of Unique Countries

```
netflix_df["country"].nunique()
```

```
748
```

Countries With Most TV Shows/Movies

```
netflix_df["country"].value_counts()
```

```
United States                                   2818
India                                            972
United Kingdom                                   419
Japan                                            245
South Korea                                      199
                                                ...
Romania, Bulgaria, Hungary                         1
Uruguay, Guatemala                                 1
France, Senegal, Belgium                           1
Mexico, United States, Spain, Colombia             1
United Arab Emirates, Jordan                       1
Name: country, Length: 748, dtype: int64
```

Oldest release year of movie/TV show on the Netflix:

```
netflix_df["release_year"].unique().min()
```

```
1925
```

Latest release year of movie/TV show on the Netflix:

```
netflix_df["release_year"].unique().max()
```

```
2021
```

Extract First Five rows of the Dataset

```
netflix_df.head()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	September 24, 2021	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...

Extract last Five rows of the Dataset

```
netflix_df.tail()
```

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	2007	R	158 min	Cult Movies, Dramas, Thrillers	A political cartoonist, a crime reporter and a...
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	2018	TV-Y7	2 Seasons	Kids' TV, Korean TV Shows, TV Comedies	While living alone in a spooky town, a young g...
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	2009	R	88 min	Comedies, Horror Movies	Looking to survive in a world taken over by zo...
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2006	PG	88 min	Children & Family Movies, Comedies	Dragged from civilian life, a former superhero...
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	2015	TV-14	111 min	Dramas, International Movies, Music & Musicals	A scrappy but poor boy worms his way into a ty...

Total Rows and Columns in Dataset

```
8807 rows × 12 columns
```

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

Attributes of DataFrame:

Attributes are the data which can be used to fetch the data or any information related to particular dataframe.

Shape of data:

```
netflix_df.shape
```

```
(8807, 12)
```

Data types of all the attributes:

```
netflix_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8807 entries, 0 to 8806
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   show_id       8807 non-null   object
 1   type          8807 non-null   object
 2   title         8807 non-null   object
 3   director      6173 non-null   object
 4   cast          7982 non-null   object
 5   country       7976 non-null   object
 6   date_added    8797 non-null   object
 7   release_year  8807 non-null   int64
 8   rating        8803 non-null   object
 9   duration      8804 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: int64(1), object(11)
memory usage: 825.8+ KB
```

Conversion of Categorical Attirbutes to ‘Catgeory’ data type:

```
netflix_df["type"]=netflix_df["type"].astype("category")
netflix_df["country"]=netflix_df["country"].astype("category")
netflix_df["rating"]=netflix_df["rating"].astype("category")
```

Conversion of data type of ‘date_added’ column from ‘object’ to ‘date_time’

```
netflix_df["date_added"]=pd.to_datetime(netflix_df["date_added"])
```

Data types of all the columns after making changes

```
netflix_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   datetime64[ns]
 7   release_year  8807 non-null   int64
 8   rating        8803 non-null   category
 9   duration      8804 non-null   object
 10  listed_in     8807 non-null   object
 11  description   8807 non-null   object
dtypes: category(3), datetime64[ns](1), int64(1), object(7)
memory usage: 676.6+ KB
```

Missing values for each column

```
netflix_df.isna().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
```

Getting only the number from 'duration column'

```
netflix_df["duration"]=netflix_df["duration"].str.split(" ",expand=True)[0]
netflix_df["duration"]=netflix_df["duration"].astype("float64")
```

Preprocessing of Data:

1. Imputing null values

We will first fill fix the 'duration ' column.
I found out that the 3 missing duration values were all from movies category so.

We will find those missing null values with average movie duration.

Average movie Duration:

```
avg_movie_duration = round(netflix_df[netflix_df["type"] == "Movie"]["duration"].mean(),0)
avg_movie_duration
```

```
100.0
```

Replacing Null values with Average movie Duration:

```
netflix_df["duration"].fillna(avg_movie_duration, inplace = True)
```

```
netflix_df["duration"].isnull().sum()
```

```
0
```

From above, we can see that now there is no null values in duration column, as all the null values is filled with the Average movie duration.

Since the data type of 'rating' column is 'category', we will use mode function to fill the null values.

Mode Value:

Mode function extracts the most common value from the respected column, that can be used to fill the null values in 'rating' column.

```
rating_mode = netflix_df["rating"].mode()[0]
rating_mode
```

```
'TV-MA'
```

'TV-MA' is the mode value for rating column.

Replacing Null values with mode value of rating column:

```
netflix_df["rating"].fillna(rating_mode, inplace = True)
```

```
netflix_df["duration"].isnull().sum()
```

```
0
```

Since the data type of 'country' is categegorical, we will use backfill

```
netflix_df["country"] = netflix_df["country"].fillna(method = "bfill")
```

```
netflix_df["country"].isnull().sum()
```

```
0
```

For 'date_added' column, we will use forwardfill

```
netflix_df["date_added"] = netflix_df["country"].fillna(method = "ffill")
```

```
netflix_df["date_added"].isnull().sum()
```

```
0
```

2. **Unnesting of 'cast'column to new 'Cast' column:**

```
cast = netflix_df["cast"].str.split(", ", expand = True)
merged_df = pd.concat([netflix_df, cast], axis = 1)
melted_df = merged_df.melt(id_vars=merged_df.columns[0:12].tolist(), value_name = "Cast")
melted_df.drop(melted_df[melted_df["Cast"].isna()].index, inplace = True)
melted_df.drop("cast", axis = 1, inplace =True)
melted_df.head()
```

show_id	type	title	director	country	date_added	release_year	rating	duration	listed_in	description	variable	Cast
s2	TV Show	Blood & Water	NaN	South Africa	2021-09-24	2021	TV-MA	2.0	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	0	Ama Qamata
s3	TV Show	Ganglands	Julien Leclercq	India	2021-09-24	2021	TV-MA	1.0	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...	0	Sami Bouajila
s5	TV Show	Kota Factory	NaN	India	2021-09-24	2021	TV-MA	2.0	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...	0	Mayur More
s6	TV Show	Midnight Mass	Mike Flanagan	United States, Ghana, Burkina Faso, United Kin...	2021-09-24	2021	TV-MA	1.0	TV Dramas, TV Horror, TV Mysteries	The arrival of a charismatic young priest brin...	0	Kate Siegel
s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	United States, Ghana, Burkina Faso, United Kin...	2021-09-24	2021	PG	91.0	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...	0	Vanessa Hudgens

64126 rows × 13 columns

3. Unnesting of 'country'column to new 'Country' column:

```
country = melted_df["country"].str.split(", ", expand = True)
merged_df1 = pd.concat([melted_df,country],axis = 1)
merged_df1.drop(["country","variable"], axis = 1, inplace = True)
merged_df1 = merged_df1.melt(id_vars = merged_df1.columns[:11], value_name = "Country").drop("variable",axis = 1)
merged_df1.drop(merged_df1[merged_df1["Country"].isna()].index, inplace = True)
merged_df1
```

	show_id	type	title	director	date_added	release_year	rating	duration	listed_in	description	Cast	Country
0	s2	TV Show	Blood & Water	NaN	2021-09-24	2021	TV-MA	2.0	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...	Ama Qamata	South Africa
1	s3	TV Show	Ganglands	Julien Leclercq	2021-09-24	2021	TV-MA	1.0	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lor...	Sami Bouajila	India
2	s5	TV Show	Kota Factory	NaN	2021-09-24	2021	TV-MA	2.0	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...	Mayur More	India
3	s6	TV Show	Midnight Mass	Mike Flanagan	2021-09-24	2021	TV-MA	1.0	TV Dramas, TV Horror, TV Mysteries	The arrival of a charismatic young priest brin...	Kate Siegel	United States
4	s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	2021-09-24	2021	PG	91.0	Children & Family Movies	Equestria's divided. But a bright-eyed hero be...	Vanessa Hudgens	United States

82199 rows × 12 columns

4. **Unnesting of 'listed_in'column to new 'Genre' column:**

```
listed_in = merged_df1["listed_in"].str.split(", ", expand = True)
merged_df2 = pd.concat([merged_df1,listed_in],axis = 1)
merged_df2.drop("listed_in", axis=1, inplace=True)
merged_df2 = merged_df2.melt(id_vars = merged_df2.columns[:11], value_name = "Genre").drop("variable",axis = 1)
merged_df2.drop(merged_df2[merged_df2["Genre"].isna()].index, inplace = True)
merged_df2
```

show_id	type	title	director	date_added	release_year	rating	duration	description	Cast	Country	Genre
s2	TV Show	Blood & Water	NaN	2021-09-24	2021	TV-MA	2.0	After crossing paths at a party, a Cape Town t...	Ama Qamata	South Africa	International TV Shows
s3	TV Show	Ganglands	Julien Leclercq	2021-09-24	2021	TV-MA	1.0	To protect his family from a powerful drug lor...	Sami Bouajila	India	Crime TV Shows
s5	TV Show	Kota Factory	NaN	2021-09-24	2021	TV-MA	2.0	In a city of coaching centers known to train I...	Mayur More	India	International TV Shows
s6	TV Show	Midnight Mass	Mike Flanagan	2021-09-24	2021	TV-MA	1.0	The arrival of a charismatic young priest brin...	Kate Siegel	United States	TV Dramas
s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	2021-09-24	2021	PG	91.0	Equestria's divided. But a bright-eyed hero be...	Vanessa Hudgens	United States	Children & Family Movies

187974 rows × 12 columns

Saving "merged_df2" to Netflix_titles.csv and Rename it to cleaned_df

```
merged_df2.to_csv('/netflix_titles.csv')
cleaned_df = pd.read_csv('/netflix_titles.csv')
cleaned_df
```

Unnamed: 0	show_id	type	title	director	date_added	release_year	rating	duration	description	Cast	Country	Genre
0	s2	TV Show	Blood & Water	NaN	2021-09-24	2021	TV-MA	2.0	After crossing paths at a party, a Cape Town t...	Ama Qamata	South Africa	International TV Shows
1	s3	TV Show	Ganglands	Julien Leclercq	2021-09-24	2021	TV-MA	1.0	To protect his family from a powerful drug lor...	Sami Bouajila	India	Crime TV Shows
2	s5	TV Show	Kota Factory	NaN	2021-09-24	2021	TV-MA	2.0	In a city of coaching centers known to train I...	Mayur More	India	International TV Shows
3	s6	TV Show	Midnight Mass	Mike Flanagan	2021-09-24	2021	TV-MA	1.0	The arrival of a charismatic young priest brin...	Kate Siegel	United States	TV Dramas
4	s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	2021-09-24	2021	PG	91.0	Equestria's divided. But a bright-eyed hero be...	Vanessa Hudgens	United States	Children & Family Movies

187974 rows × 13 columns

Removing “Unnamed: 0” column from cleaned_df

```
cleaned_df.drop("Unnamed: 0", axis=1, inplace=True)
cleaned_df.head()
```

show_id	type	title	director	date_added	release_year	rating	duration	description	Cast	Country	Genre
s2	TV Show	Blood & Water	NaN	2021-09-24	2021	TV-MA	2.0	After crossing paths at a party, a Cape Town t...	Ama Qamata	South Africa	International TV Shows
s3	TV Show	Ganglands	Julien Leclercq	2021-09-24	2021	TV-MA	1.0	To protect his family from a powerful drug lor...	Sami Bouajila	India	Crime TV Shows
s5	TV Show	Kota Factory	NaN	2021-09-24	2021	TV-MA	2.0	In a city of coaching centers known to train I...	Mayur More	India	International TV Shows
s6	TV Show	Midnight Mass	Mike Flanagan	2021-09-24	2021	TV-MA	1.0	The arrival of a charismatic young priest brin...	Kate Siegel	United States	TV Dramas
s7	Movie	My Little Pony: A New Generation	Robert Cullen, José Luis Ucha	2021-09-24	2021	PG	91.0	Equestria's divided. But a bright-eyed hero be...	Vanessa Hudgens	United States	Children & Family Movies

Getting the most popular director for each Country:

```
direct = cleaned_df.groupby("Country")[["title", "director"]].value_counts().reset_index()
pop_dir = direct.groupby(["Country"]).apply(lambda x:x["director"].value_counts().head(1)).reset_index()
pop_dir
```

	Country	level_1	director
0	Afghanistan	Pieter-Jan De Pue	1
1	Albania	Antonio Morabito	1
2	Algeria	Youssef Chahine	1
3	Angola	Chris Roland, Maradona Dias Dos Santos	1
4	Argentina	Raúl Campos, Jan Suter	5
...	...	...	...
108	Vatican City	Wim Wenders	1
109	Venezuela	Sebastián Schindel	1
110	Vietnam	Victor Vu	1
111	West Germany	Mel Stuart	1
112	Zimbabwe	Tomas Brickhill	1

113 rows × 3 columns

Merge with Cleaned_df & then Remove NaN Values with popular director of that country:

```
qw = cleaned_df.merge(pop_dir, how = 'left', on = "Country")
qw["director_x"].fillna(qw["director_y"], inplace = True)
qw.drop("director_y", axis =1, inplace = True)
```

Unnesting of “director” column with new “Director”

```
director_x = qw["director_x"].str.split(", ", expand = True)
final = pd.concat([qw,director_x],axis = 1)
final.drop("director_x", axis = 1, inplace = True)
final = final.melt(id_vars = final.columns[:11], value_name = "Director").drop("variable",axis = 1)
final.drop(final[final["Director"].isna()].index, inplace = True)
final
```

Cleaned Dataframe: final

	show_id	type	title	date_added	release_year	rating	duration	description	Cast	Country	Genre	Director
0	s2	TV Show	Blood & Water	2021-09-24	2021	TV-MA	2.0	After crossing paths at a party, a Cape Town t...	Ama Qamata	South Africa	International TV Shows	Adze Ugah
1	s3	TV Show	Ganglands	2021-09-24	2021	TV-MA	1.0	To protect his family from a powerful drug lor...	Sami Bouajila	India	Crime TV Shows	Rajiv Chilaka
2	s5	TV Show	Kota Factory	2021-09-24	2021	TV-MA	2.0	In a city of coaching centers known to train I...	Mayur More	India	International TV Shows	Rajiv Chilaka
3	s6	TV Show	Midnight Mass	2021-09-24	2021	TV-MA	1.0	The arrival of a charismatic young priest brin...	Kate Siegel	United States	TV Dramas	Marcus Raboy
4	s7	Movie	My Little Pony: A New Generation	2021-09-24	2021	PG	91.0	Equestria's divided. But a bright-eyed hero be...	Vanessa Hudgens	United States	Children & Family Movies	Marcus Raboy
...	...	...	...	...	...	...	...	...	...	...	...	...
2502786	s5888	Movie	Walt Disney Animation Studios Short Films Coll...	2015-10-25	2015	TV-Y	90.0	This collection of 12 short films from Disney ...	Dave Foley	United States	Children & Family Movies	Mark Henn
2503984	s5888	Movie	Walt Disney Animation Studios Short Films Coll...	2015-10-25	2015	TV-Y	90.0	This collection of 12 short films from Disney ...	Derek Richardson	United States	Children & Family Movies	Mark Henn
2504819	s5888	Movie	Walt Disney Animation Studios Short Films Coll...	2015-10-25	2015	TV-Y	90.0	This collection of 12 short films from Disney ...	Betty White	United States	Children & Family Movies	Mark Henn
2505424	s5888	Movie	Walt Disney Animation Studios Short Films Coll...	2015-10-25	2015	TV-Y	90.0	This collection of 12 short films from Disney ...	Zachary Levi	United States	Children & Family Movies	Mark Henn
2505895	s5888	Movie	Walt Disney Animation Studios Short Films Coll...	2015-10-25	2015	TV-Y	90.0	This collection of 12 short films from Disney ...	Mandy Moore	United States	Children & Family Movies	Mark Henn

340059 rows × 12 columns

Statistical Analysis

Top 5 Directors

```
final.groupby("Director").apply(lambda x: x["title"].nunique()).sort_values(ascending = False).head(5)
```

```
Director
Marcus Raboy       3609
Rajiv Chilaka      1090
Martin Campbell     774
Suhas Kadav         495
Justin G. Dyck      453
dtype: int64
```

Top 5 Countries

```
final.groupby("Country").apply(lambda x: x["title"].nunique()).sort_values(ascending = False).head(5)
```

```
Country
United States     3609
India             1083
United Kingdom     772
Canada             453
France             393
dtype: int64
```

First Movie/Show added on Netflix:

```
final.loc[final["date_added"] == min(final["date_added"]),["title", "date_added"]].drop_duplicates()
```

	title	date_added
5415	To and From New York	2008-01-01

Latest Movie/Show added on Netflix:

```
] final.loc[final["date_added"] == max(final["date_added"]),["title", "date_added"]].head(1)
```

	title	date_added
0	Blood & Water	2021-09-24

Top 10 Popular Actor and Actresses:

```
final.groupby("Cast").apply(lambda x: x["title"].nunique()).sort_values(ascending = False).head(10)
```

```
Cast
Anupam Kher         43
Shah Rukh Khan      35
Julie Tejwani       33
Naseeruddin Shah    32
Takahiro Sakurai    32
Rupa Bhimani        31
Akshay Kumar        30
Om Puri             30
Yuki Kaji           29
Amitabh Bachchan    28
dtype: int64
```

Aggregate quantitative details about the Movies

```
final.loc[final["type"]=="Movie", ["duration","release_year","title"]].drop_duplicates().describe()
```

	duration	release_year
count	5656.000000	5656.000000
mean	101.355552	2012.911775
std	27.797722	9.599338
min	8.000000	1942.000000
25%	88.000000	2011.000000
50%	100.000000	2016.000000
75%	116.000000	2018.000000
max	312.000000	2021.000000

Aggregate quantitative details about the TV Shows

```
final.loc[final["type"]=="TV Show", ["duration","release_year","title"]].drop_duplicates().describe()
```

	duration	release_year
count	2323.000000	2323.000000
mean	1.837279	2016.504520
std	1.662850	5.257565
min	1.000000	1963.000000
25%	1.000000	2015.000000
50%	1.000000	2018.000000
75%	2.000000	2020.000000
max	17.000000	2021.000000

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

Value counts of Movies/TV Shows

```
final.groupby("type")["title"].apply(lambda x: x.nunique())
```

```
type
Movie      5656
TV Show    2323
Name: title, dtype: int64
```

Value_counts of release years

```
final.groupby("release_year")["title"].apply(lambda x: x.nunique())
```

```
release_year
1942       1
1944       1
1945       1
1946       1
1947       1
        ...
2017     911
2018    1026
2019     917
2020     827
2021     494
Name: title, Length: 72, dtype: int64
```

Unique years

```
final["release_year"].unique()
```

```
array([2021, 1993, 2020, 2018, 1996, 1998, 1997, 2010, 2013, 2017, 1975,
       1978, 1983, 1987, 2012, 2001, 2014, 2002, 2003, 2004, 2011, 2008,
       2009, 2007, 2005, 2006, 1994, 2019, 2016, 2015, 1982, 1989, 1990,
       1991, 1999, 1986, 1992, 1984, 1980, 1961, 2000, 1995, 1985, 1976,
       1959, 1988, 1981, 1972, 1964, 1954, 1979, 1958, 1956, 1963, 1970,
       1973, 1974, 1960, 1966, 1971, 1962, 1969, 1977, 1967, 1968, 1965,
       1945, 1946, 1955, 1942, 1947, 1944])
```

Value counts of rating category

```
final.groupby("rating")["title"].apply(lambda x: x.nunique())
```

```
rating

G              40
NC-17           3
NR             63
PG            279
PG-13         477
R             790
TV-14        1954
TV-G          183
TV-MA        2884
TV-PG         719
TV-Y          267
TV-Y7         310
TV-Y7-FV        4
UR              3
Name: title, dtype: int64
```

Value counts of Country

```
final.groupby("Country")["title"].apply(lambda x: x.nunique())
```

```
Country
Afghanistan       1
Albania           1
Algeria           6
Angola            1
Argentina        88
                 ..
Vatican City      1
Venezuela         2
Vietnam           7
West Germany      4
Zimbabwe          1
Name: title, Length: 113, dtype: int64
```

Unique Countries

```
final["Country"].unique()
```

```
array(['South Africa', 'India', 'United States', 'United Kingdom',
       'Germany', 'Mexico', 'Turkey', 'Australia', 'Finland', 'China',
       'Nigeria', 'Japan', 'Spain', 'Belgium', 'France', 'South Korea',
       'Argentina', 'Russia', 'Canada', 'Hong Kong', 'Italy', 'Ireland',
       'New Zealand', 'Jordan', 'Colombia', 'Switzerland', 'Israel',
       'Taiwan', 'Bulgaria', 'Poland', 'Saudi Arabia', 'Thailand',
       'Indonesia', 'Kuwait', 'Egypt', 'Malaysia', 'Vietnam', 'Sweden',
       'Lebanon', 'Brazil', 'Romania', 'Philippines', 'Iceland',
       'Denmark', 'United Arab Emirates', 'Netherlands', 'Norway',
       'Syria', 'Mauritius', 'Austria', 'Czech Republic', 'Cameroon',
       'United Kingdom,', 'Kenya', 'Chile', 'Luxembourg', 'Bangladesh',
       'Portugal', 'Hungary', 'Senegal', 'Singapore', 'Serbia', 'Namibia',
       'Uruguay', 'Peru', 'Mozambique', 'Ghana', 'Zimbabwe', 'Cyprus',
       'Pakistan', 'Paraguay', 'Croatia', 'Cambodia', 'Soviet Union',
       'Georgia', 'Iran', 'Venezuela', 'Poland,', 'Slovenia', 'Guatemala',
       'Jamaica', 'Somalia', 'Nepal', 'Algeria', 'Malta', 'Angola',
       'Iraq', 'Malawi', 'West Germany', 'Qatar', 'Morocco', 'Slovakia',
       'Bermuda', 'Sri Lanka', 'Nicaragua', 'Greece', 'Vatican City',
       'Lithuania', 'East Germany', 'Burkina Faso', 'Cayman Islands',
       'Albania', 'Ecuador', 'Dominican Republic', 'Sudan', 'Cambodia,',
       'Latvia', 'Liechtenstein', 'Panama', 'Montenegro', 'Bahamas',
       'Afghanistan', 'Ethiopia', nan], dtype=object)
```

Value counts of Genre

```
final.groupby("Genre")["title"].apply(lambda x: x.nunique())
```

```
Genre
Action & Adventure                 853
Anime Features                      68
Anime Series                       173
British TV Shows                   287
Children & Family Movies           608
Classic & Cult TV                   28
Classic Movies                     189
Comedies                          1662
Crime TV Shows                     394
Cult Movies                         78
Documentaries                      445
Docuseries                         188
Dramas                            2416
Faith & Spirituality                60
Horror Movies                      354
Independent Movies                 753
International Movies              2574
International TV Shows            1240
Kids' TV                           408
Korean TV Shows                    147
LGBTQ Movies                        85
Movies                              53
Music & Musicals                   340
Reality TV                         163
Romantic Movies                    609
Romantic TV Shows                  357
Sci-Fi & Fantasy                   240
Science & Nature TV                 57
Spanish-Language TV Shows          162
Sports Movies                      165
Stand-Up Comedy                    342
Stand-Up Comedy & Talk Shows        49
TV Action & Adventure              166
TV Comedies                        555
TV Dramas                          756
TV Horror                           72
TV Mysteries                        93
TV Sci-Fi & Fantasy                 82
TV Shows                            11
TV Thrillers                        54
Teen TV Shows                       66
Thrillers                          577
Name: title, dtype: int64
```

Unique Genres

```
final["Genre"].unique()
```

```
array(['International TV Shows', 'Crime TV Shows', 'TV Dramas',
       'Children & Family Movies', 'Dramas', 'British TV Shows',
       'Comedies', 'TV Comedies', 'Thrillers', 'Docuseries',
       'Horror Movies', "Kids' TV", 'Action & Adventure', 'Reality TV',
       'Documentaries', 'Anime Series', 'International Movies',
       'Sci-Fi & Fantasy', 'Classic Movies', 'TV Shows',
       'Stand-Up Comedy', 'TV Action & Adventure', 'Movies',
       'Stand-Up Comedy & Talk Shows', 'Classic & Cult TV',
       'Anime Features', 'Romantic TV Shows', 'Cult Movies',
       'Independent Movies', 'TV Horror', 'Spanish-Language TV Shows',
       'Music & Musicals', 'Romantic Movies', 'LGBTQ Movies',
       'TV Sci-Fi & Fantasy', 'Sports Movies', 'Korean TV Shows',
       'Faith & Spirituality', 'TV Mysteries', 'Teen TV Shows',
       'Science & Nature TV', 'TV Thrillers'], dtype=object)
```

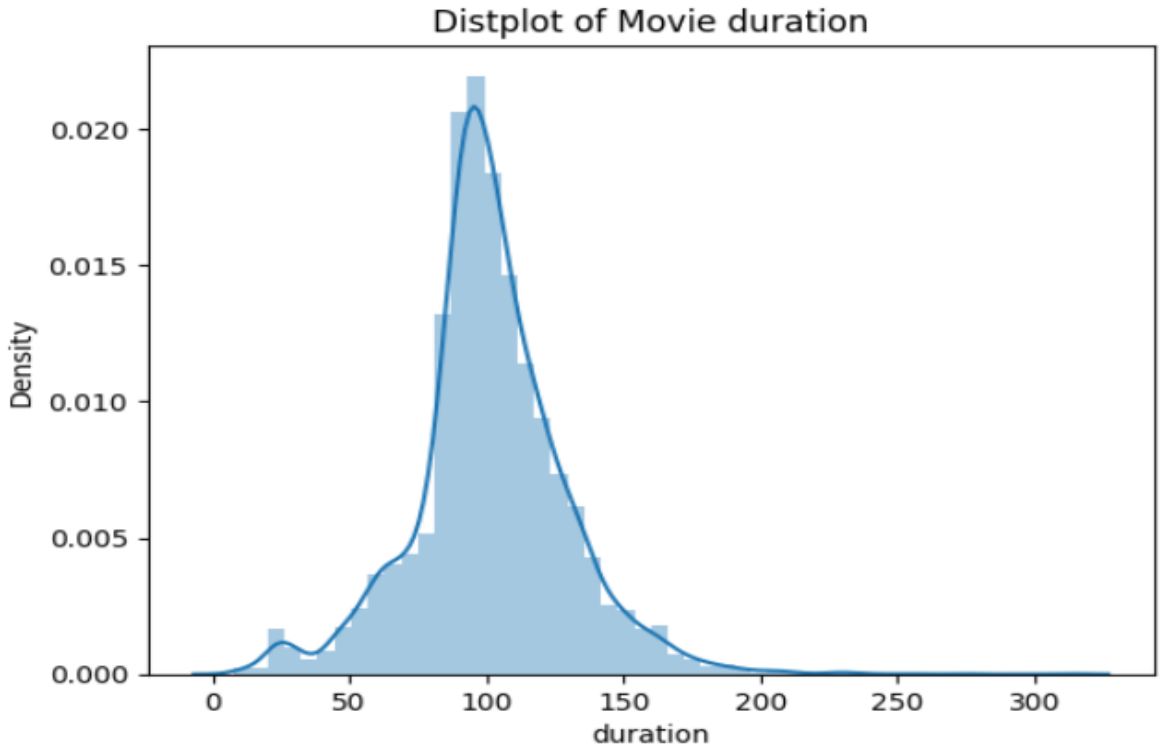
4. Visual Analysis : Univariate, Bivariate after pre-processing of the Data

Note: Pre-processing involves unnesting of the data in columns like Actor, Director, Countrcy

4.1 For Continuous Variable(s): Distplot, countplot and Histogram for univariate Analysis:

Distplot of Movie duration:

```
data_ = final.loc[final["type"]=="Movie",["title", "duration"]].drop_duplicates()
plt.figure(figsize=(16, 8))
sns.distplot(data_["duration"])
plt.title("Distplot of Movie duration")
```

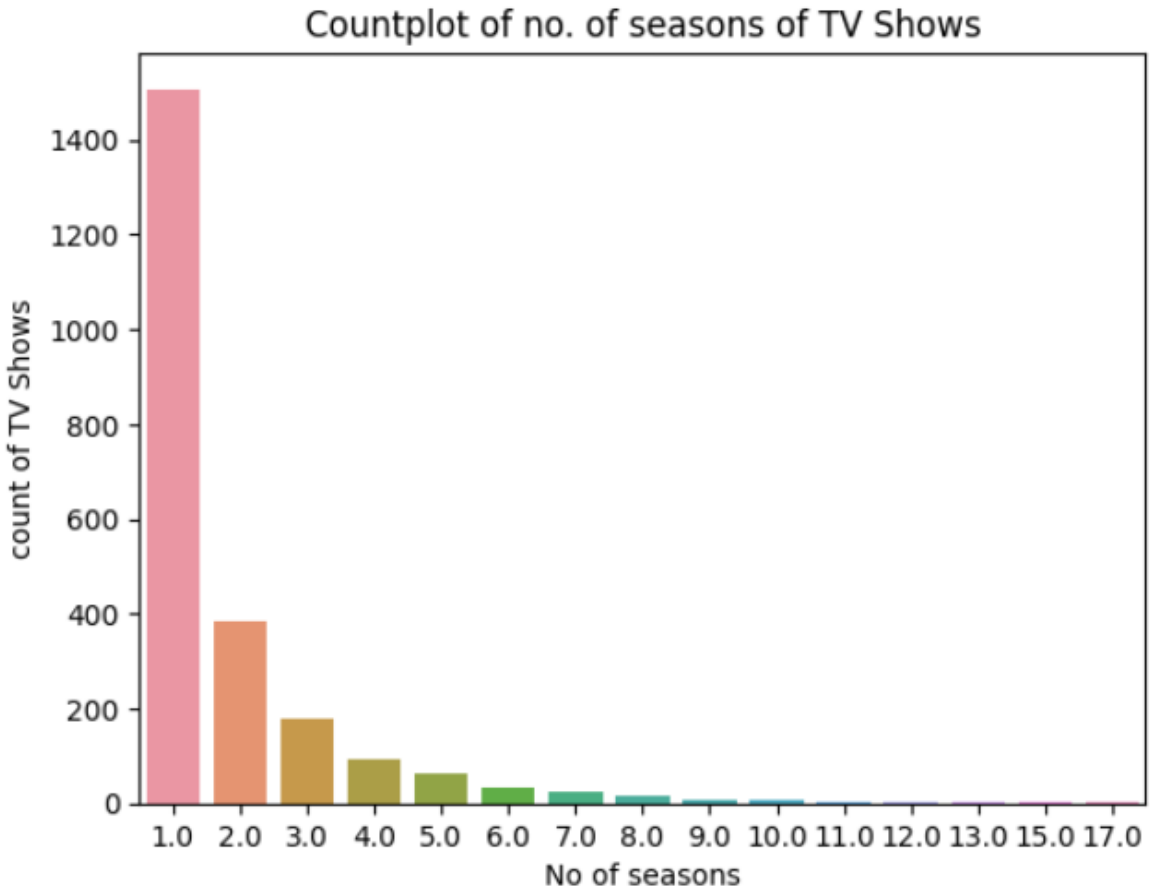


 # Majority of the movies have a duration of about 100 minutes

 # There are less number of movies with a duration between 150-300 minutes.

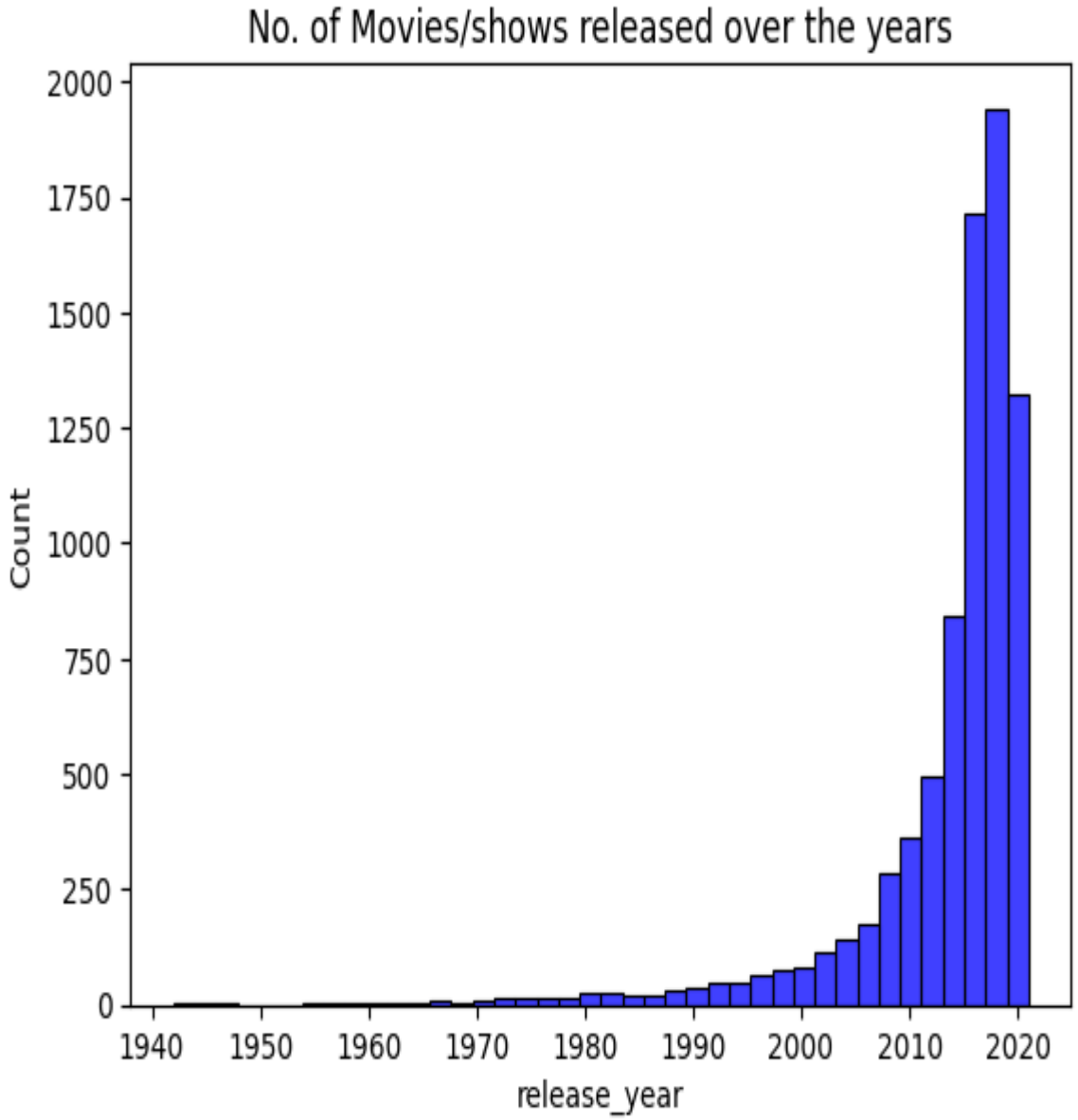
<u>Countplot of no.of Seasons of TV Shows:</u>

```
data_ = final.loc[final["type"]=="TV Show",["title", "duration"]].drop_duplicates()["duration"].value_counts().reset_index()
sns.barplot(data = data_, x= "index", y = "duration")
plt.xlabel("No of seasons")
plt.ylabel("count of TV Shows")
plt.title("Countplot of no. of seasons of TV Shows")
```



Histogram of no.of movies/shows released over the years:

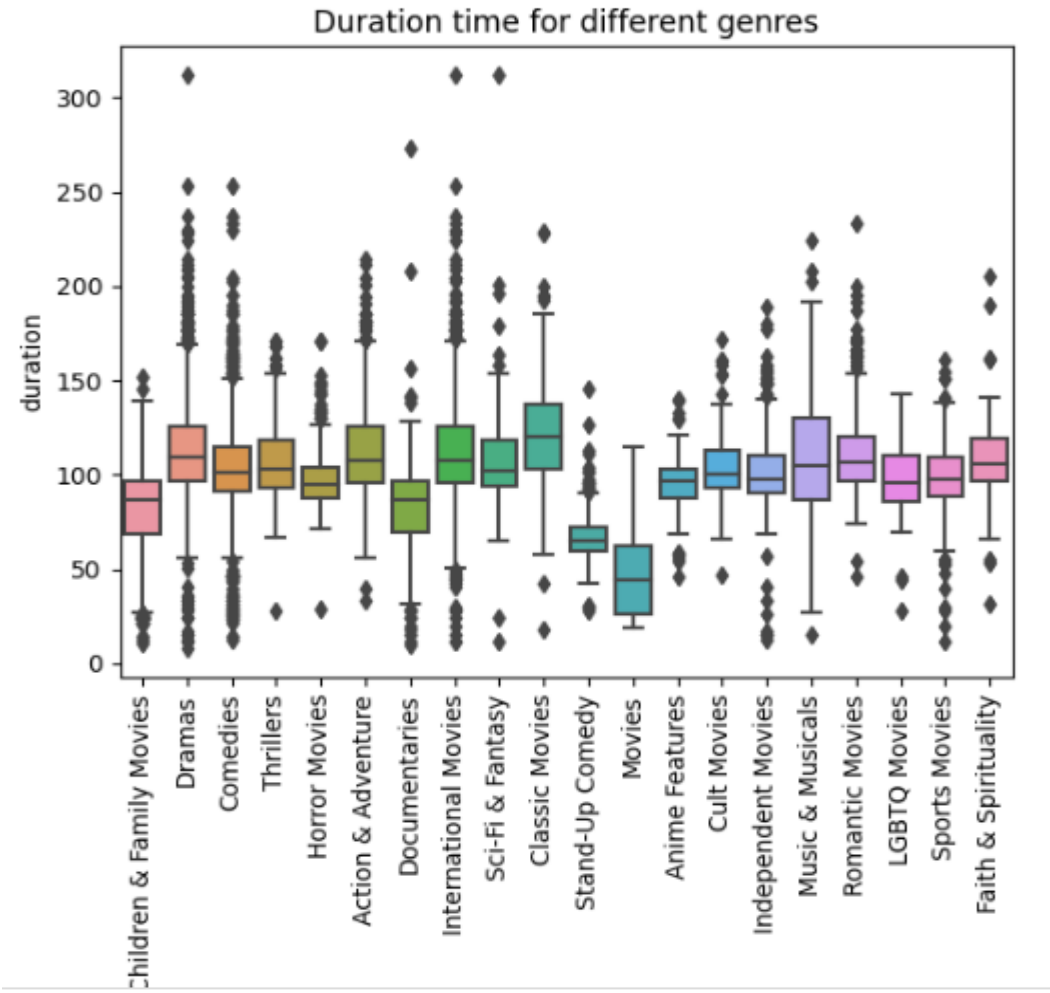
```
data_ = final.loc[:,["title", "release_year"]].drop_duplicates()
plt.figure(figsize=(16, 8))
sns.histplot(data = data_, x= "release_year",bins=40,color = "blue")
plt.title("No. of Movies/shows released over the years")
```



4.2 Categorical Data:

Duration time for the different genres of movies:

```
data_ = final.loc[final["type"]=="Movie", ["title", "Genre", "duration"]].drop_duplicates()
plt.xticks(rotation=90)
sns.boxplot(data = data_, x = "Genre", y = "duration")
plt.title("Duration time for different genres")
```



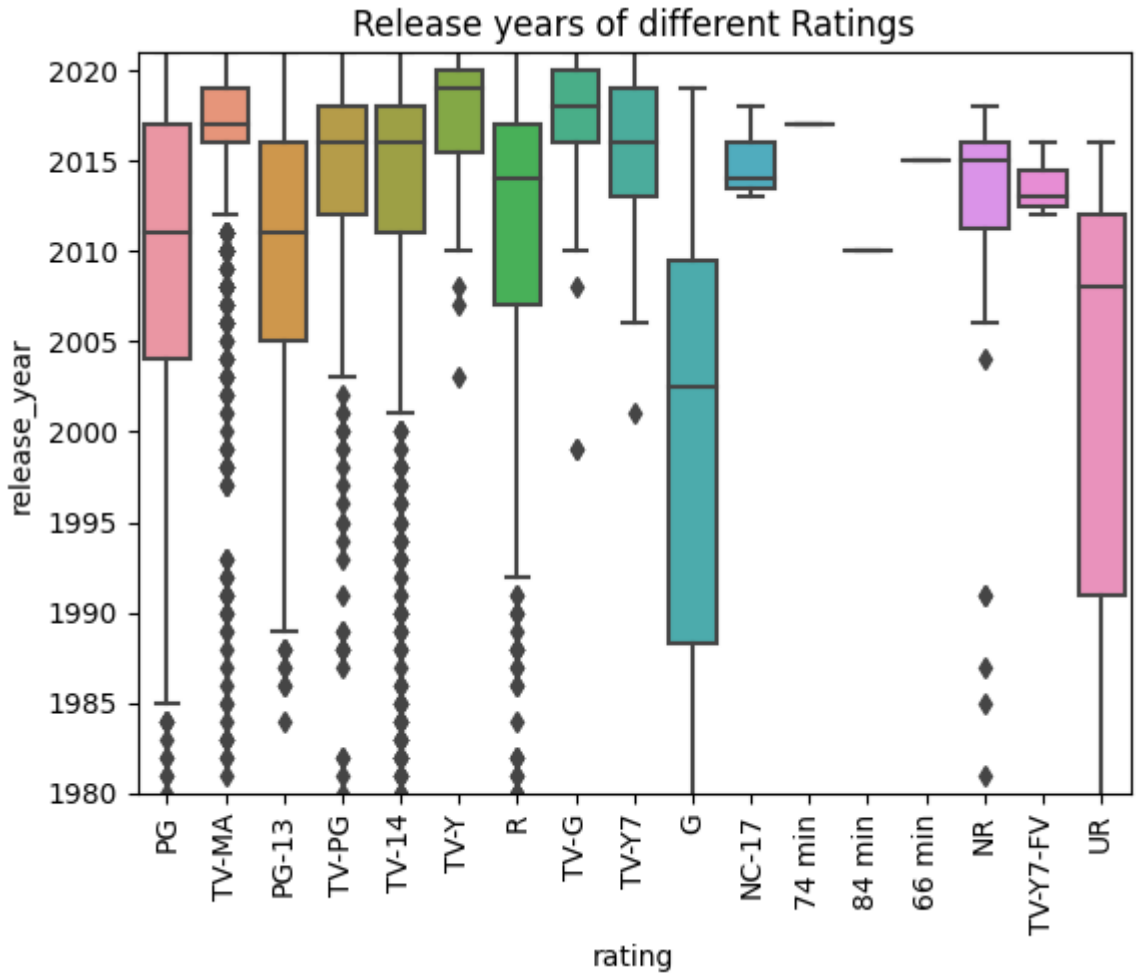
We observe medium duration of ‘Classical movies’ is the Highest.

The genre of ‘movies’ has the least median duration.

The genre ‘International Movies’ and ‘Drama’ have the highest Number of Outliers.

Release year of different ratings:

```
data_ = final.loc[final["type"]=="Movie", ["title", "rating", "release_year"]].drop_duplicates()
plt.xticks(rotation=90)
plt.ylim([1980,2021])
sns.boxplot(data = data_, x = "rating", y = "release_year")
plt.title("Release years of different Ratings")
```



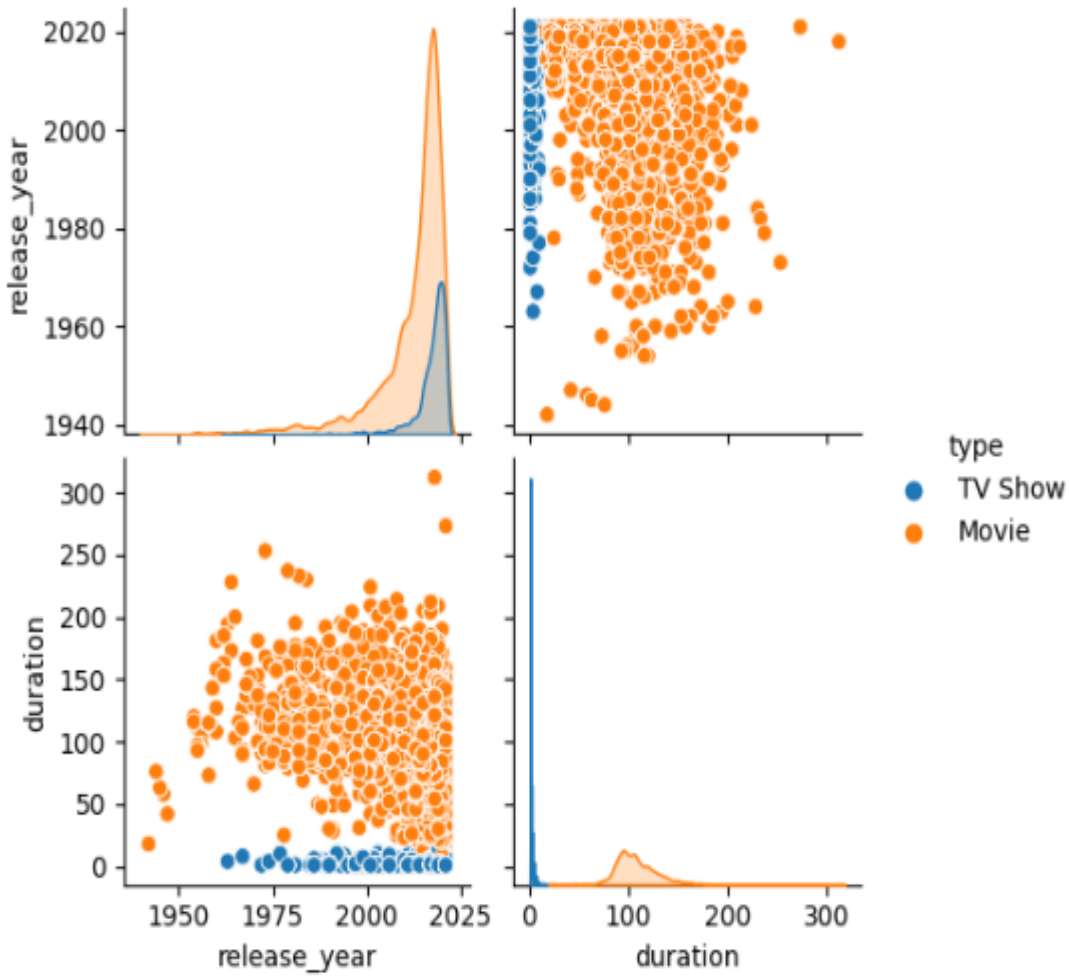
We observe that rating category 'G' and 'UR' are mostly for old movies/shows.

The rating category 'TV-Y' and 'TV-G' are mostly for newer movies/shows.

4.3 Heat maps and Pair plots:

Pair Plot for Numeric Data:

```
final2 = final.copy()
plt.figure(figsize = (18,12))
sns.pairplot(final2, hue = "type")
```



We see that TV shows duration mostly appear at 1, and movies mainly appear around 100.

Most of the movies/shows have been added recently.

The release years have been sparse before the year 2000, but after that it seems the number per year is uniform.

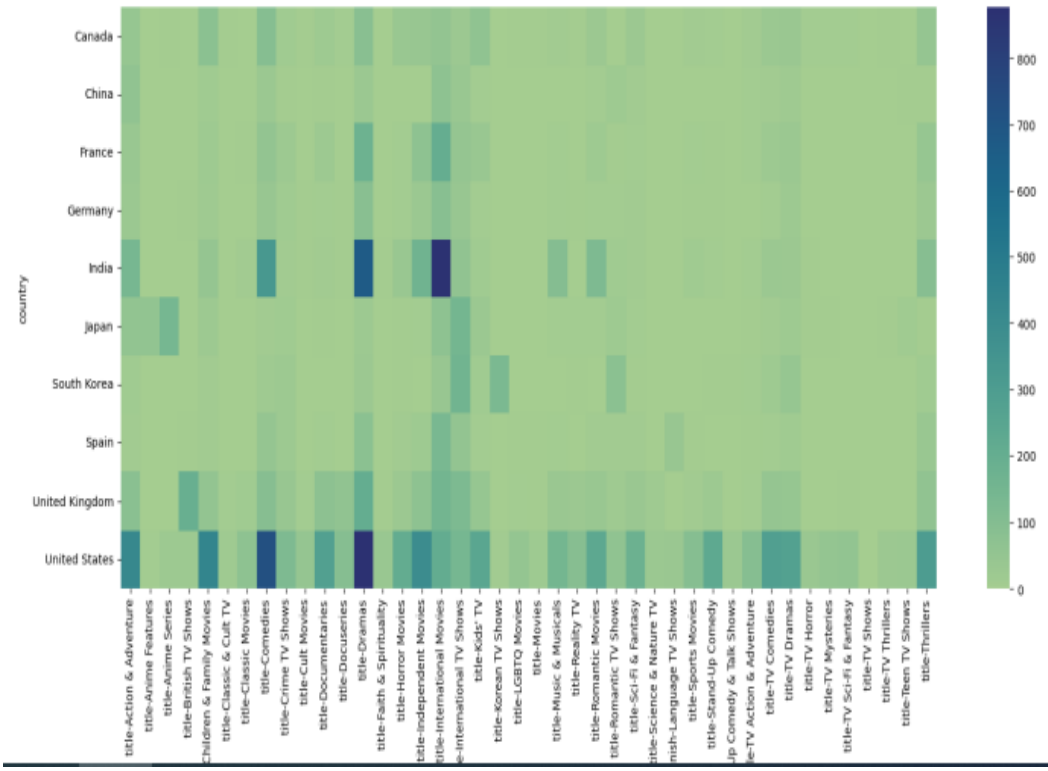
Heatmap to show which genre is the most popular among the top 10 countries:

```
top_country = final.groupby("country").apply(lambda x: x["title"].nunique()).sort_values(ascending = False).head(
data_ = final.loc[final["country"].isin(top_country),["title", "country", "genre"]].drop_duplicates()
data_ = pd.pivot_table(data = data_, index = "country", columns = "genre", aggfunc = "count").fillna(0)
plt.figure(figsize = (18,8))
sns.heatmap(data_,cmap = "crest")

# In India, the genre 'International movies' and 'Dramas' seems to be most popular.
# In US, the genre 'Dramas' and 'Comedy' seems to be the most popular.
```

Out[268]:

```
<Axes: xlabel='None-genre', ylabel='country'>
```



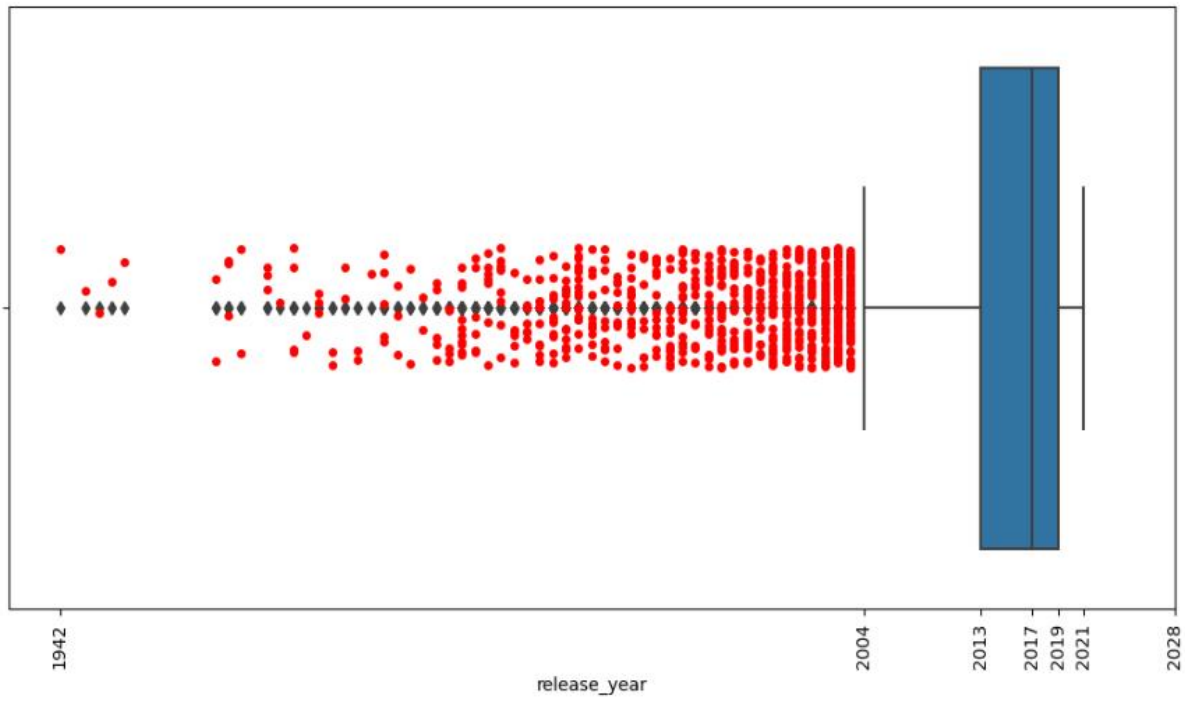
5. Missing Values and Outlier Check

5.1 Missing values have already been addressed in the Preprocessing of the Data set

5.2 Outlier Check

Checking the outlier in Release_year column

```
df = final.loc[:, ["title", "release_year"]].drop_duplicates()
outl = df["release_year"].describe()
Q1 = outl.loc["25%"]
Q3 = outl.loc["75%"]
iqr = Q3 - Q1
low = Q1 - 1.5*iqr
upp = Q3 + 1.5*iqr
outliers = df[(df["release_year"]<low) | (df["release_year"]>upp)]
plt.figure(figsize = (18,8))
plt.xticks(rotation=90)
sns.boxplot(x = df["release_year"])
sns.stripplot(x = outliers["release_year"], color = "red")
plt.xticks([df["release_year"].min(), low, Q1,df["release_year"].median(), Q3, upp, df["release_year"].max() ])
plt.show()
```



Since most of the movies/shows have been added recently, there are no outliers above the upper whisker

All the shows/movies in the outliers are from the year 1942 to 2004.

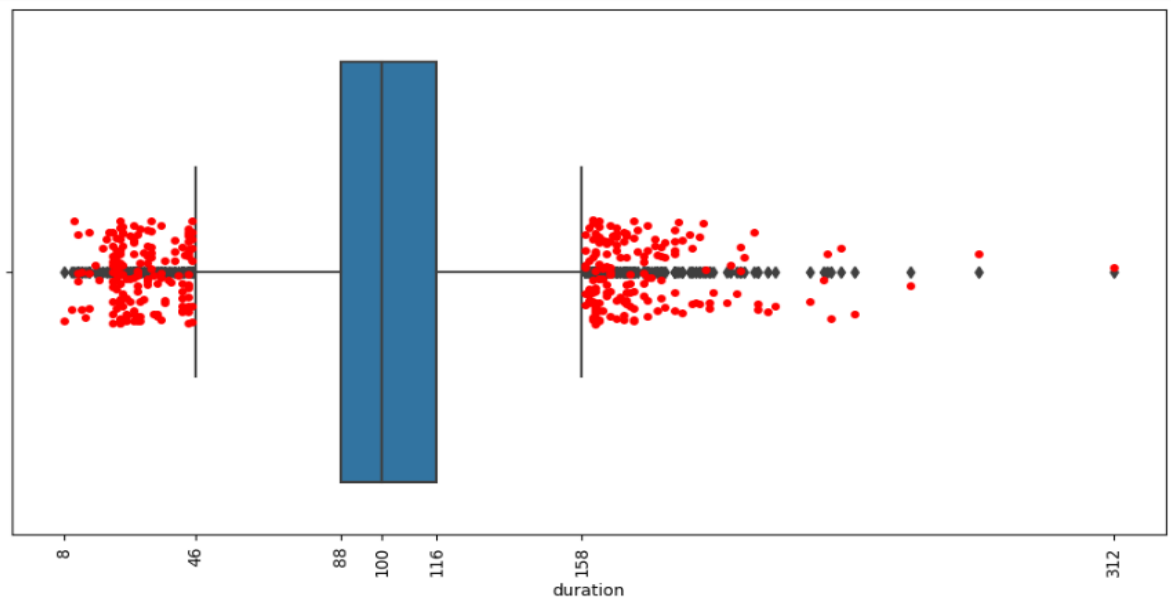
```
outliers
```

	title	release_year
5	Sankofa	1993
16	Avvai Shanmughi	1996
18	Jeans	1998
20	Minsara Kanavu	1997
35	Jaws	1975
...	...	...
7940	Wyatt Earp	1994
7942	XXx	2002
7944	Y Tu Mamá También	2001
7946	Yaadein	2001
7968	Young Tiger	1973

700 rows × 2 columns

Checking the outlier in Movies duration column:

```
df = final.loc[final["type"] =="Movie", ["title", "duration"]].drop_duplicates()
outl = df["duration"].describe()
Q1 = outl.loc["25%"]
Q3 = outl.loc["75%"]
iqr = Q3 - Q1
low = Q1 - 1.5*iqr
upp = Q3 + 1.5*iqr
outliers = df[(df["duration"]<low) | (df["duration"]>upp)]
plt.figure(figsize = (12,6))
plt.xticks(rotation=90)
sns.boxplot(x = df["duration"])
sns.stripplot(x = outliers["duration"], color = "red")
plt.xticks([df["duration"].min(), low, Q1,df["duration"].median(), Q3, upp, df["duration"].max()])
plt.show()
```



We see there are many outliers below the time duration of 46 mins.

The outliers beyond upper whisker range from 158 - 312 mins.

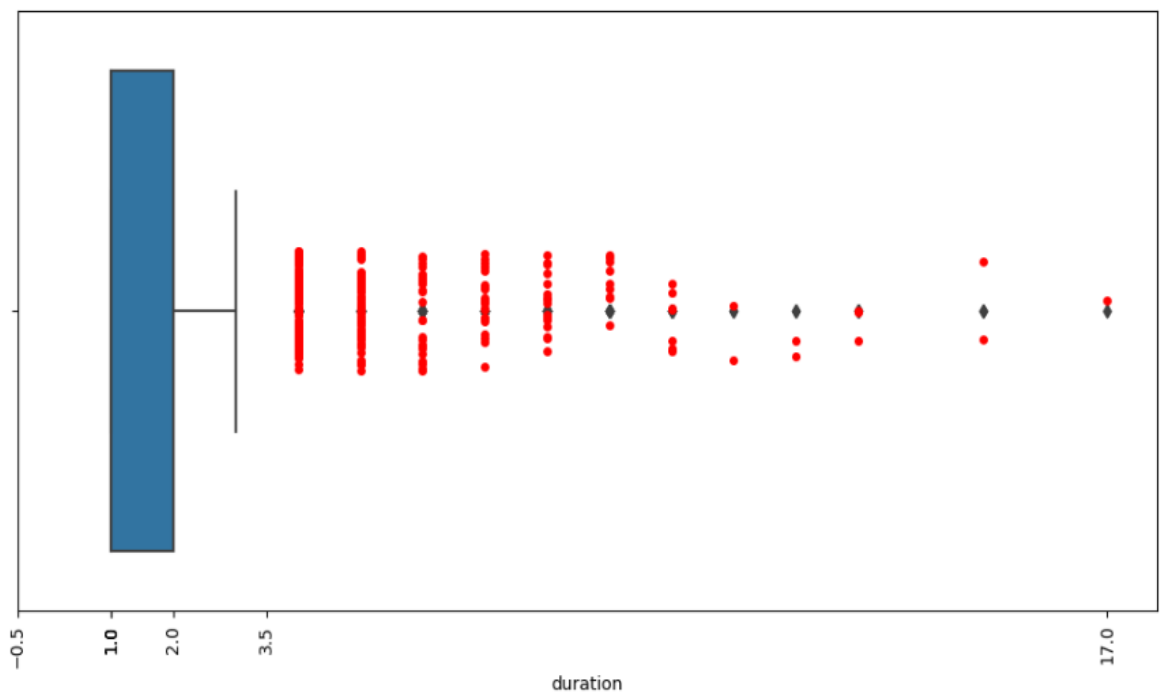
```
outliers
```

	title	duration
16	Avvai Shanmughi	161.0
18	Jeans	166.0
62	A StoryBots Space Adventure	13.0
64	King of Boys	182.0
148	Once Upon a Time in America	229.0
...	...	...
7824	Trimurti	173.0
7833	Tukaram	162.0
7849	Under an Arctic Sky	40.0
7940	Wyatt Earp	191.0
7946	Yaadein	171.0

325 rows × 2 columns

Checking the outlier in TV Show duration column:

```
df = final.loc[final["type"] =="TV Show", ["title", "duration"]].drop_duplicates()
outl = df["duration"].describe()
Q1 = outl.loc["25%"]
Q3 = outl.loc["75%"]
iqr = Q3 - Q1
low = Q1 - 1.5*iqr
upp = Q3 + 1.5*iqr
outliers = df[(df["duration"]<low) | (df["duration"]>upp)]
plt.figure(figsize = (12,6))
plt.xticks(rotation=90)
sns.boxplot(x = df["duration"])
sns.stripplot(x = outliers["duration"], color = "red")
plt.xticks([df["duration"].min(), low, Q1,df["duration"].median(), Q3, upp, df["duration"].max()])
plt.show()
```



Most of the TV Shows are having only one Season.

That is why there is no lower whisker, the median itself is 1.

Outliers start appearing after season 4 or more.

```
outliers
```

	title	duration
6	The Great British Baking Show	9.0
11	Dear White People	4.0
15	Resurrection: Ertugrul	5.0
48	Nailed It	6.0
58	Numberblocks	6.0
...	...	...
7747	The Twilight Zone (Original Series)	4.0
7762	The West Wing	7.0
7846	Ugly Duckling	4.0
7896	Weeds	8.0
7911	When Calls the Heart	5.0

255 rows × 2 columns

6.1 Insights on the range of Attributes:

Release Year: From the above box plot to find the outliers in the release_year column, we see that the range of movie/TV show release year is from 1942 to 2021. The older movies/shows are less compared to recently released ones.

Movie Duration: From the above box plot to find the outliers in the Movie Duration column, we see that it ranges from as low as 8 minutes to 312 minutes. However, the ideal time duration for a movie is 100 mins(median).

TV Show Seasons: From the above mentioned box plots, we see that the number of seasons of TV show ranges from 1 to

7. Most of them are 1 Season shows. There are less number of TV shows which have 4 or more Seasons.

Rating: The number of movies/shows for each rating range from 3 (NC-17, UR) to 2884 (TV-MA). Which means the succefull shows on Netflix are usually from the rating of TV-MA and TV-14.

Genre: The number of movies/shows for each genre is mapped. It is found that 'International Movies' genre has 2574(highest) count and 'TV Shows' genre has 11(least) count.

6.2 Distribution of variables and relation between them

It is seen that 'release year' and date added' variables are mildly related, which makes sense because older movies/shows added in the beginning, and over the years as and when new ones came, they were added on the platform. There is no relation between 'duration' and 'date added'. However 'duration' and 'release year' have negative correlation which means the duration of movies/shows have have slightly decreased over the years.

7. Business Insights:

1. There are 113 Countries but most of the movies/shows come from these top 5 Countries – US, India, UK, Canada and France
2. Successful Directors: Marcus Raboy, Martin Campbell, Toshiya Sinohra.
3. There are 70% of the content on the neflix is movies and 30% of the content on the Netflix is TV shows.

```
final.groupby("type")["title"].apply(lambda x: x.nunique())*100/final.groupby("type")["title"].apply(lambda x: x.
```

```
Out[337]:

type
Movie      70.875125
TV Show    29.124875
Name: title, dtype: float64
```

4. Successfull Actors: Anupam Kher and Shah rukh khan have been featured in the most number of movies. And the top actors list is dominated my India.

5. Top Genre: The top 3 Genres are 'International Movies', 'Drama' and 'Comedy'.

6. Duration: The median duration for Movies and TV shows are 1h 40mins and 1 season respectively.

7. Genre: Anime and Classical Movie genre are becoming popular recently.

8. Genre duration: We observe median duration of 'classical movies' is the highest and the genre of 'Movies' is the least.

9. Favourite genre in the biggest markets: Popular genre in US is 'Drama' and in India it is 'International Movies'.

10. We can see from the below table that “Director” Rajiv Chilaka and Anupam kher as “Cast” in most number of movies.

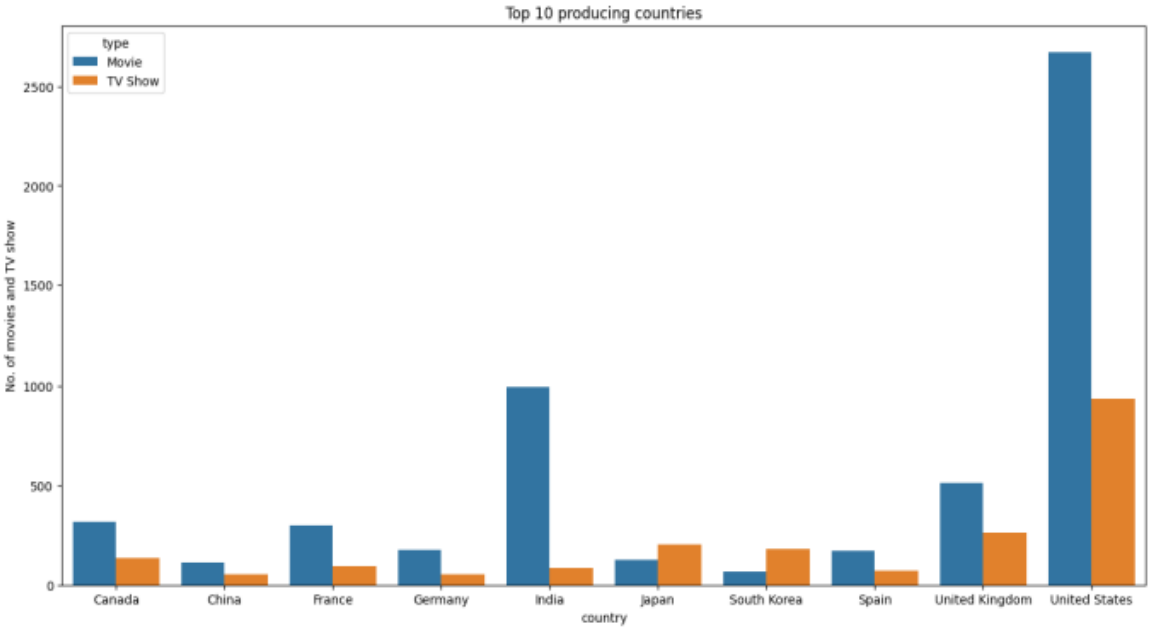
```
data_ = final.loc[:, ["Cast", "title", "Director"]].drop_duplicates()
data_ = data_.groupby(["Director","Cast"]).count().sort_values(by = "title", ascending = False).reset_index()
data_.head(10)
```

	Director	Cast	title
0	Rajiv Chilaka	Anupam Kher	40
1	Rajiv Chilaka	Shah Rukh Khan	34
2	Rajiv Chilaka	Naseeruddin Shah	31
3	Toshiya Shinohara	Takahiro Sakurai	29
4	Rajiv Chilaka	Akshay Kumar	29
5	Rajiv Chilaka	Om Puri	29
6	Toshiya Shinohara	Yuki Kaji	28
7	Rajiv Chilaka	Paresh Rawal	28
8	Rajiv Chilaka	Amitabh Bachchan	28
9	Rajiv Chilaka	Boman Irani	27

11. In Japan and South Korea, TV shows are more popular than movies. Rest of the remaining top countries, movies are more popular than TV shows.

```
#Top 10 countries and their distribution of movies and TV shows
data_ = final.loc[:, ["type", "title", "country"]].drop_duplicates()
data_ = data_.groupby(["country", "type"])["title"].count().reset_index()
top_country = final.groupby("country").apply(lambda x: x["title"].nunique()).sort_values(ascending = False).head(
data_ = data_[data_["country"].isin(top_country)]

plt.figure(figsize=(16, 8))
sns.barplot(data = data_, x = "country", y ="title", hue = "type")
plt.ylabel("No. of movies and TV show")
plt.title("Top 10 producing countries")
plt.show()
```



8. Recommendations:

1. Country: There are 113 countries but not all of them give the most return. We should focus the content more on important countries which - US, India, UK, Canada and France.
2. Successful directors: Since certain director's movie/show are featured more than others, Netflix can make original movies/show by hiring the top directors. For example: Marcus Raboy, Martin Campbell, Toshiya Shinohara.

3. Successful Actors: If Netflix has the budget to pay for star - studded cast, it can hire popular actors/actress to attract more people into the platform. For example: Anupam Kher, Shah Rukh Khan, Takahiro Sakurai etc,.
4. Director - Cast combo: If Netflix has budget constraint, it can hire successful yet lesser know Director- Cast combination. The best combination is mentioned in the table above.
5. Targeting the right genre for specific countries: Netflix can recommend popular genre to the audience of that country. For example: US - Drama, comedy, India - International Movies, UK - 'British TV Shows', Japan - Anime etc,.
6. Duration: Netflix can give more preference to movies whose duration is around 1h 40mins, and shows with 1 or 2 seasons. Since data suggests, this is the ideal duration.
7. Netflix can produce or sponsor more towards specific genres of movies/show. From the data it is visible that specific genre like 'Anime' and 'classical movies' are getting popular recently throughout the world.
8. In countries like Japan and South Korea, Netflix should recommend more TV shows rather than wasting resources on Movies.
9. Rating: If Netflix does produce its original content it should prefer TV-Y, TV-G rating category. Since they are more popular recently.