# Problem Statement

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

### 1. Analyzing basic metrics

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

```
df=pd.read_csv("https://d2beiqkhq929f0.cloudfront.net/public_assets/assets/000/000/940/original/netflix.csv")
df.head()
```

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

```
df.keys()
```

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

```
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 | United States | January 11, 2020 | 2006 | PG | 88 min | Children & Family Movies, Comedies | Dragged from civilian life, a former superhero... |

### 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

```
df.size
```

```
105684
```

```
df.shape
```

```
(8807, 12)
```

```
df.describe()
```

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

```
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
```

```
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
```

```
df['release_year'].max()
```

```
2021
```

```
df['release_year'].min()
```

```
1925
```

Updating Proper Datatypes

```
df['date_added']=pd.to_datetime(df['date_added'])
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 | 2021-09-25 | 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 | 2021-09-24 | 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 | 2021-09-24 | 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 | Jailbirds New | NaN | NaN | NaN | 2021-09-24 | 2021 | TV-MA | 1 Season | Docuseries, | Feuds, flirtations and |

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

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

```
array(['Movie', 'TV Show'], dtype=object)
```

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

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

```
df['director'].nunique()
```

```
4528
```

```
df['country'].nunique()
```

```
748
```

```
df['country'].value_counts().head()
```

```
United States     2818
India              972
United Kingdom     419
Japan              245
South Korea        199
Name: country, dtype: int64
```

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

```
TV-MA    3207
TV-14    2160
TV-PG     863
R         799
PG-13     490
Name: rating, dtype: int64
```

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

```
NC-17     3
UR        3
74 min    1
84 min    1
66 min    1
Name: rating, dtype: int64
```

```
df['director'].value_counts().head()
```

```
Rajiv Chilaka            19
Raúl Campos, Jan Suter   18
Marcus Raboy             16
Suhas Kadav              16
Jay Karas                14
Name: director, dtype: int64
```

### Checking For Missing Values and Handling them

```
df.fillna({'director':'Unavailable','cast':'Unavailable','rating':'Unavailable',
           'country':'Unavailable'},inplace=True)
df.isna().sum()
```

```
show_id          0
type             0
title            0
director         0
cast             0
country          0
date_added      10
release_year     0
rating           0
duration         3
listed_in        0
description      0
dtype: int64
```

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

```
10
```

```
most_recent_entry_date=df['date_added'].max()
df.fillna({'date_added':most_recent_entry_date}, inplace=True)
df.head()
```

```
<ipython-input-24-7870fe8cda8d>:2: DeprecationWarning: In a future version, `df.iloc[:, i] = newvals` will attempt to set the values inp
  df.fillna({'date_added':most_recent_entry_date}, inplace=True)
```

| | 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 | Unavailable | United States | 2021-09-25 | 2020 | PG-13 | 90 min | Documentaries | As her father nears the end of his life, filmm... |
| **1** | s2 | TV Show | Blood & Water | Unavailable | Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban... | South Africa | 2021-09-24 | 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... | Unavailable | 2021-09-24 | 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 | Unavailable | Unavailable | Unavailable | 2021-09-24 | 2021 | TV-MA | 1 Season | Docuseries, Reality TV | Feuds, flirtations and toilet talk go down amo... |

```
df[df.duration.isnull()]
```

| | show_id | type | title | director | cast | country | date_added | release_year | rating | duration | listed_in | description |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **5541** | s5542 | Movie | Louis C.K. 2017 | Louis C.K. | Louis C.K. | United States | 2017-04-04 | 2017 | 74 min | NaN | Movies | Louis C.K. muses on religion, eternal love, gi... |
| **5794** | s5795 | Movie | Louis C.K.: Hilarious | Louis C.K. | Louis C.K. | United States | 2016-09-16 | 2010 | 84 min | NaN | Movies | Emmy-winning comedy writer Louis C.K. brings h... |
| **5813** | s5814 | Movie | Louis C.K.: Live at the Comedy Store | Louis C.K. | Louis C.K. | United States | 2016-08-15 | 2015 | 66 min | NaN | Movies | The comic puts his trademark hilarious/thought... |

```
df[df.director=='Louis C.K.'].head()
```

| | show_id | type | title | director | cast | country | date_added | release_year | rating | duration | listed_in | description |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **5541** | s5542 | Movie | Louis C.K. 2017 | Louis C.K. | Louis C.K. | United States | 2017-04-04 | 2017 | 74 min | NaN | Movies | Louis C.K. muses on religion, eternal love, gi... |
| **5794** | s5795 | Movie | Louis C.K.: Hilarious | Louis C.K. | Louis C.K. | United States | 2016-09-16 | 2010 | 84 min | NaN | Movies | Emmy-winning comedy writer Louis C.K. brings h... |
| **5813** | s5814 | Movie | Louis C.K.: Live at the Comedy Store | Louis C.K. | Louis C.K. | United States | 2016-08-15 | 2015 | 66 min | NaN | Movies | The comic puts his trademark hilarious/thought... |

```
df.loc[df['director']=='Louis C.K.','duration']=df['rating']
df[df['director']=='Louis C.K.'].head()
```

| | show_id | type | title | director | cast | country | date_added | release_year | rating | duration | listed_in | description |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **5541** | s5542 | Movie | Louis C.K. 2017 | Louis C.K. | Louis C.K. | United States | 2017-04-04 | 2017 | 74 min | 74 min | Movies | Louis C.K. muses on religion, eternal love, gi... |
| **5794** | s5795 | Movie | Louis C.K.: Hilarious | Louis C.K. | Louis C.K. | United States | 2016-09-16 | 2010 | 84 min | 84 min | Movies | Emmy-winning comedy writer Louis C.K. brings h... |
| **5813** | s5814 | Movie | Louis C.K.: Live at the Comedy Store | Louis C.K. | Louis C.K. | United States | 2016-08-15 | 2015 | 66 min | 66 min | Movies | The comic puts his trademark hilarious/thought... |

```
df.loc[df['director']=='Louis C.K.','rating']='Unavailable'
df[df['director']=='Louis C.K.'].head()
```

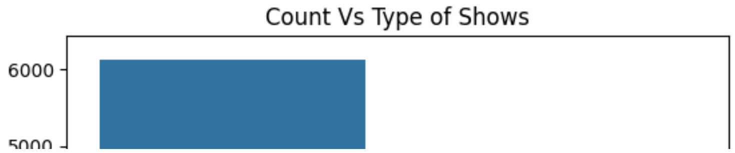
| | show_id | type | title | director | cast | country | date_added | release_year | rating | duration | listed_in | description |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **5541** | s5542 | Movie | Louis C.K. 2017 | Louis C.K. | Louis C.K. | United States | 2017-04-04 | 2017 | Unavailable | 74 min | Movies | Louis C.K. muses on religion, eternal love, gi... |
| **5794** | s5795 | Movie | Louis C.K.: Hilarious | Louis C.K. | Louis C.K. | United States | 2016-09-16 | 2010 | Unavailable | 84 min | Movies | Emmy-winning comedy writer Louis C.K. brings h... |
| **5813** | s5814 | Movie | Louis C.K.: Live at the Comedy Store | Louis C.K. | Louis C.K. | United States | 2016-08-15 | 2015 | Unavailable | 66 min | Movies | The comic puts his trademark hilarious/thought... |

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

### 1. Analysis / Continuous Variables

```
sns.countplot(x='type',data=df)
plt.title('Count Vs Type of Shows')
```

```
Text(0.5, 1.0, 'Count Vs Type of Shows')
```

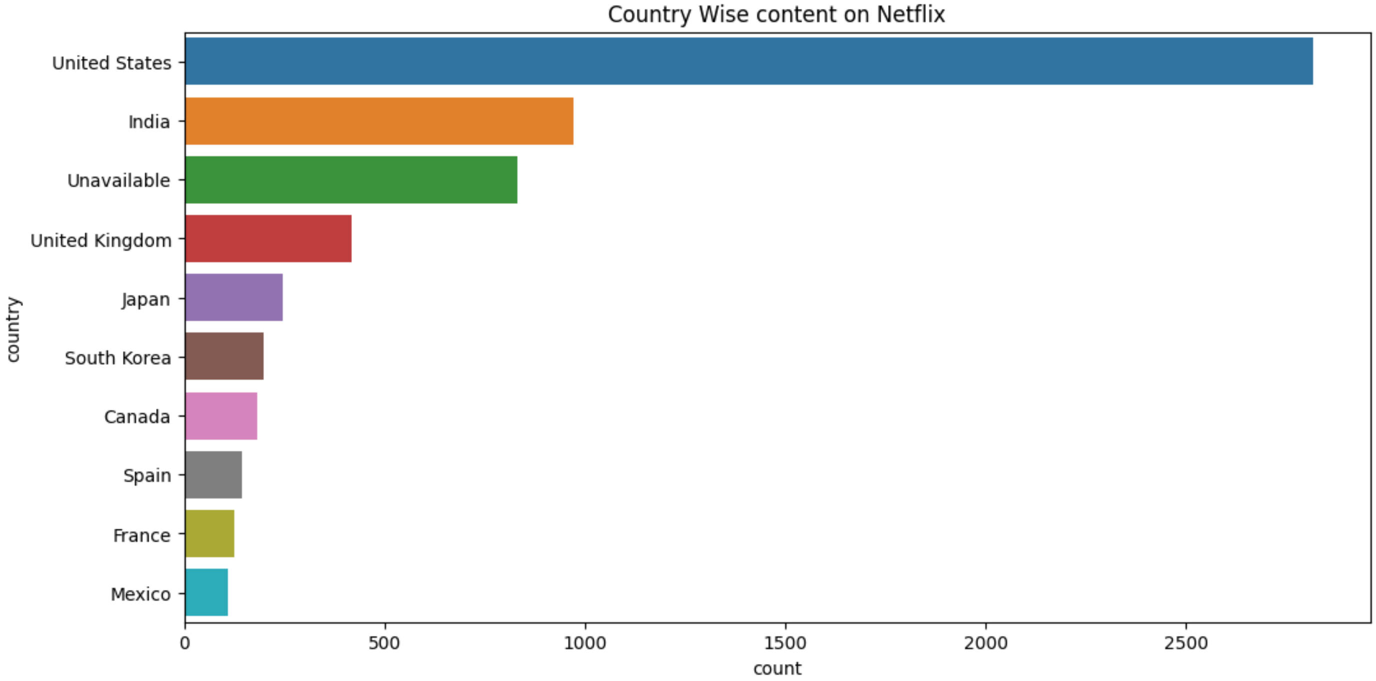


```
df['country'].value_counts().head(10)
```

```
United States     2818
India              972
Unavailable        831
United Kingdom     419
Japan              245
South Korea        199
Canada             181
Spain              145
France             124
Mexico             110
Name: country, dtype: int64
```

```
plt.figure(figsize=(12,6))
sns.countplot(y='country',order=df['country'].value_counts().index[0:10],data=df)
plt.title('Country Wise content on Netflix')
```

```
Text(0.5, 1.0, 'Country Wise content on Netflix')
```

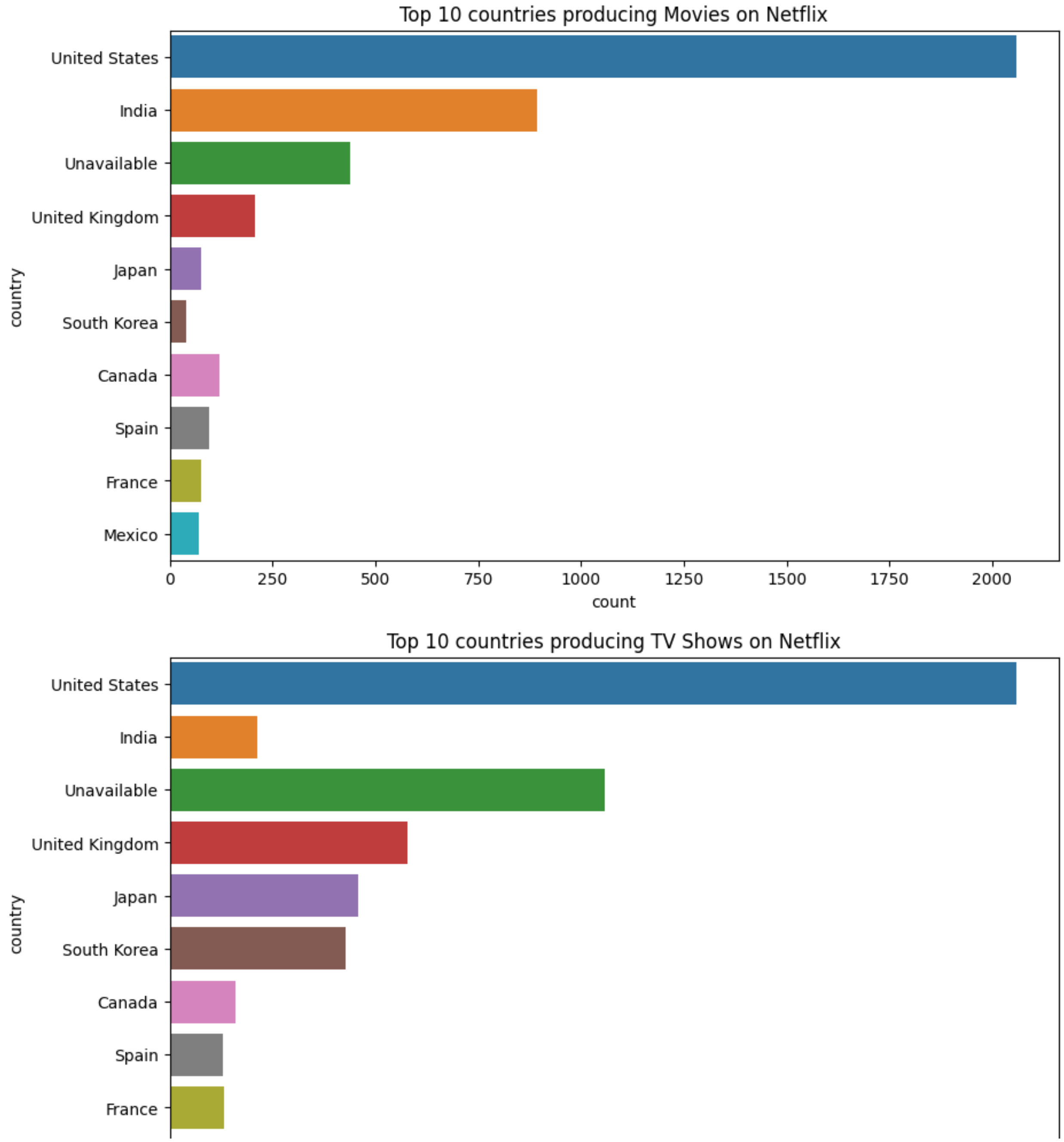


```
movie_country=df[df['type']=='Movie']
tv_show_country=df[df['type']=='TV Show']
release_year=df[df['type']=='release_year']
```

```
plt.figure(figsize=(10,6))
sns.countplot(y='country',order=df['country'].value_counts().index[0:10],data=movie_country)
plt.title('Top 10 countries producing Movies on Netflix')
```

```
plt.figure(figsize=(10,6))
sns.countplot(y='country',order=df['country'].value_counts().index[0:10],data=tv_show_country)
plt.title('Top 10 countries producing TV Shows on Netflix')
```

```
Text(0.5, 1.0, 'Top 10 countries producing TV Shows on Netflix')
```

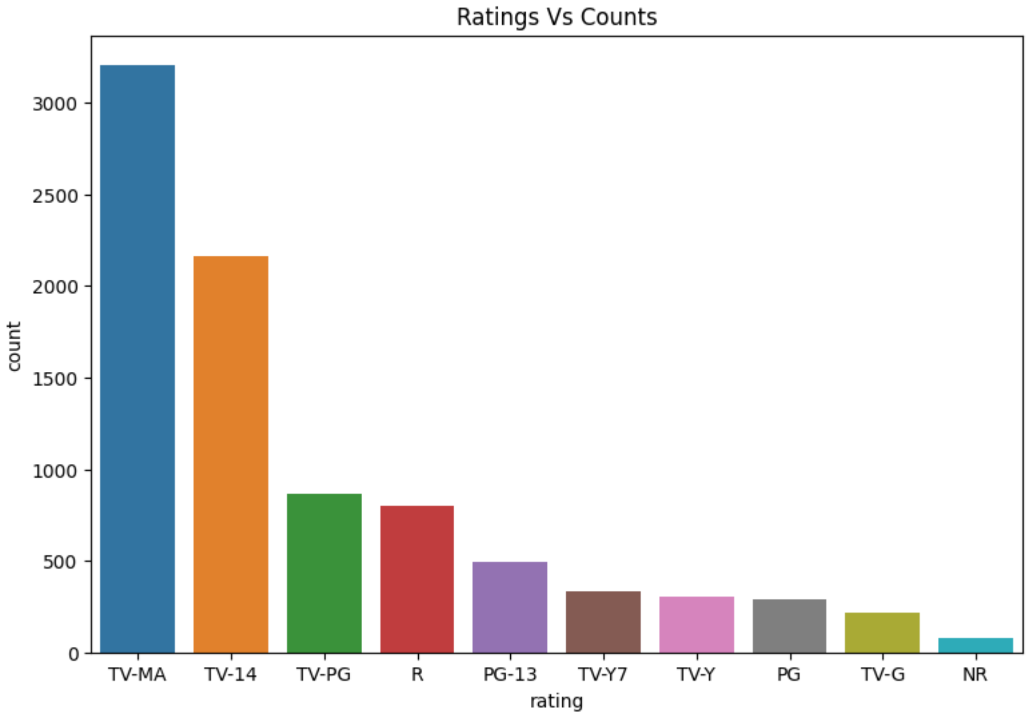


```
df.rating.value_counts()
```

```
TV-MA          3207
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
Unavailable       7
TV-Y7-FV          6
NC-17             3
UR                3
Name: rating, dtype: int64
```

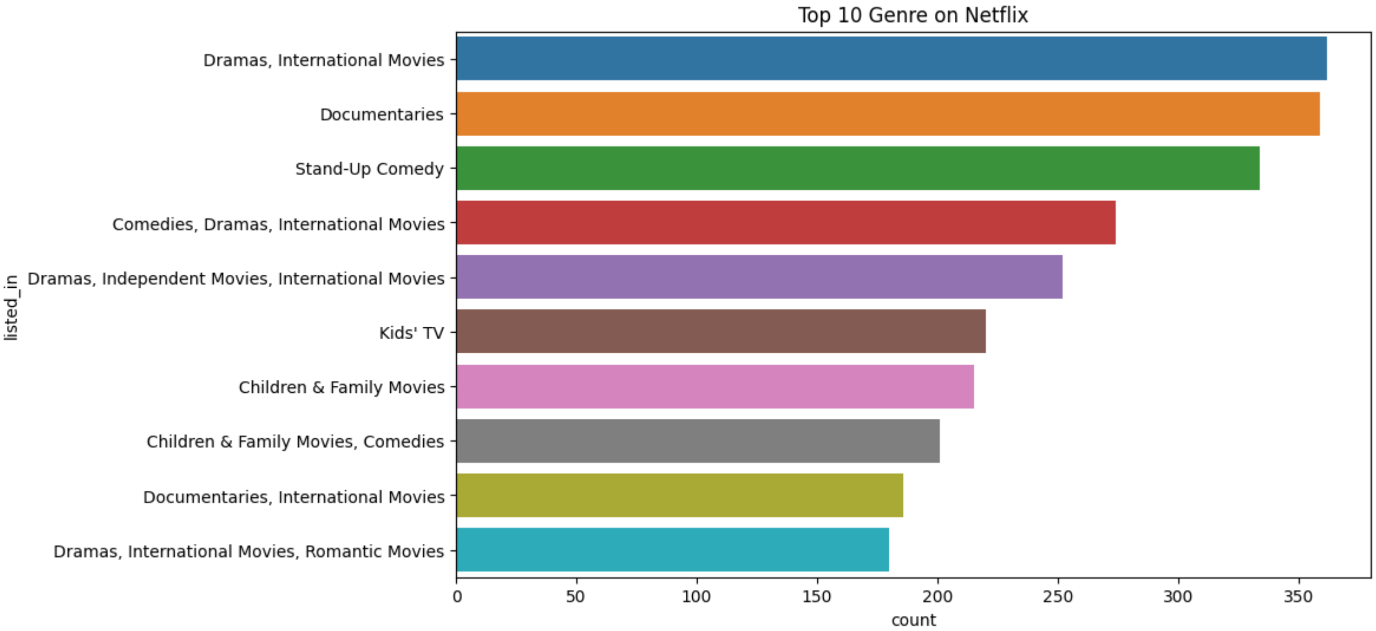
```
plt.figure(figsize=(9,6))
sns.countplot(x='rating',order=df['rating'].value_counts().index[0:10],data=df)
plt.title('Ratings Vs Counts')
```

```
Text(0.5, 1.0, 'Ratings Vs Counts')
```



```
plt.figure(figsize=(10,6))
sns.countplot(y='listed_in',order=df['listed_in'].value_counts().index[0:10],data=df)
plt.title('Top 10 Genre on Netflix')
```

```
Text(0.5, 1.0, 'Top 10 Genre on Netflix')
```



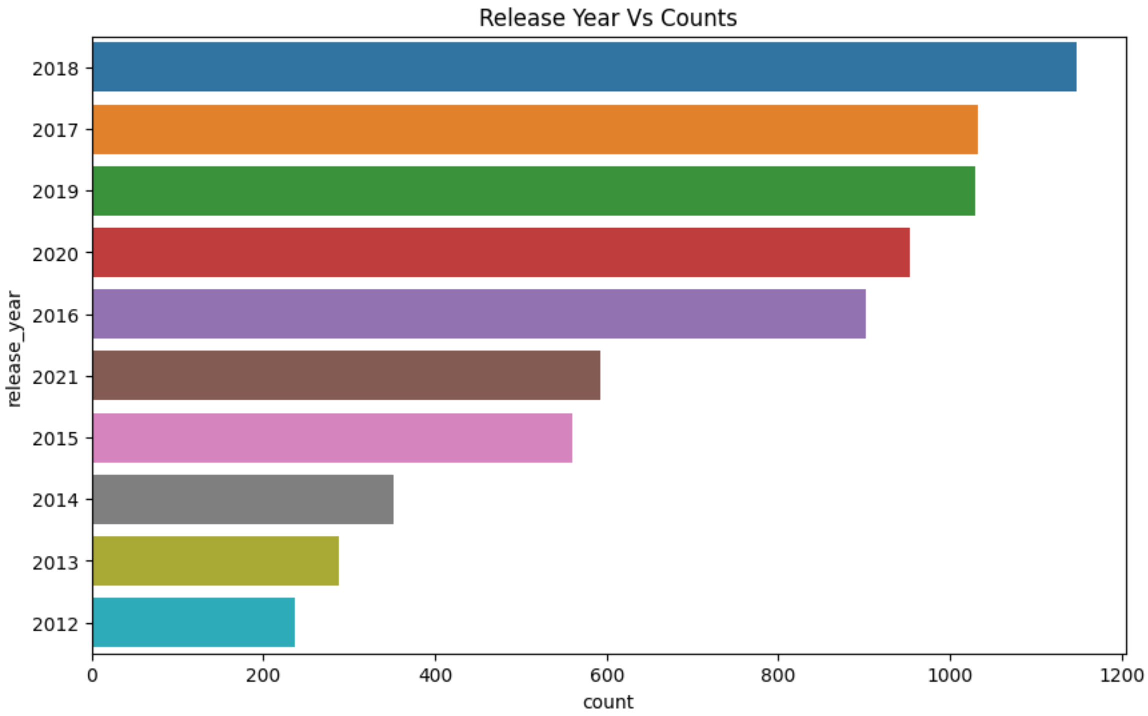
```
df.release_year.value_counts()[:10]
```

```
2018    1147
2017    1032
2019    1030
2020     953
2016     902
2021     592
2015     560
2014     352
2013     288
```

```
    2012     237
    Name: release_year, dtype: int64
```

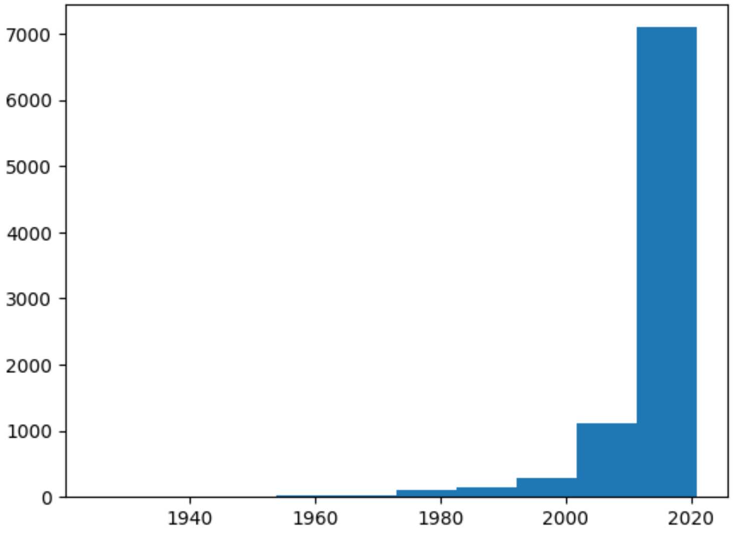
```
plt.figure(figsize=(10,6))
sns.countplot(y='release_year',order=df['release_year'].value_counts().index[0:10],data=df)
plt.title('Release Year Vs Counts')
```

```
    Text(0.5, 1.0, 'Release Year Vs Counts')
```



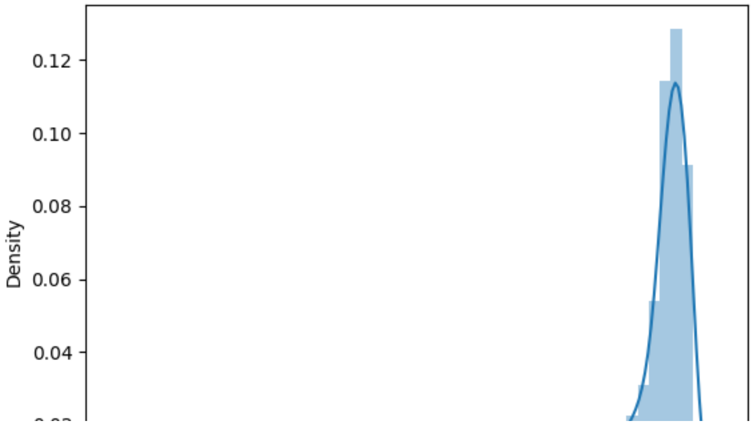
```
plt.hist(df['release_year'])
```

```
    (array([1.000e+00, 8.000e+00, 7.000e+00, 2.100e+01, 2.700e+01, 9.900e+01,
            1.500e+02, 2.940e+02, 1.107e+03, 7.093e+03]),
     array([1925. , 1934.6, 1944.2, 1953.8, 1963.4, 1973. , 1982.6, 1992.2,
            2001.8, 2011.4, 2021. ]),
     <BarContainer object of 10 artists>)
```



```
sns.distplot(df['release_year'])
```

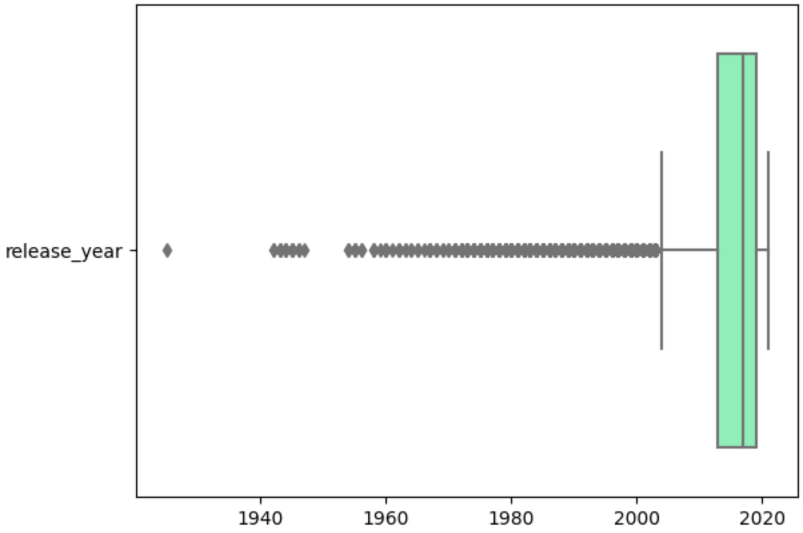
```
<ipython-input-40-5635d90732bd>:1: UserWarning:

`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['release_year'])
<Axes: xlabel='release_year', ylabel='Density'>
```



### 4.2 Categorical Variable

```
sns.boxplot(data=df, palette='rainbow',orient='h')
```

```
<Axes: >
```

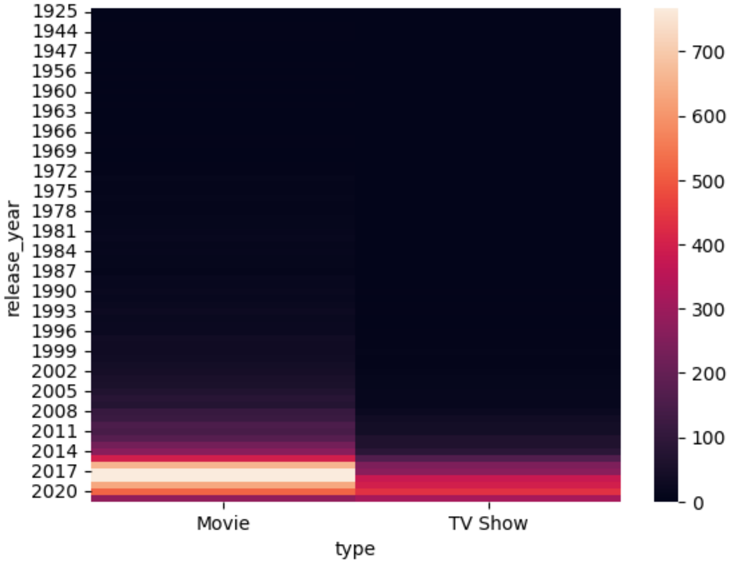


```
sns.boxplot(x='release_year',y='type',data=df,orient='h')
plt.show()
```

### 4.3 For Correlation

```
sns.heatmap(pd.crosstab(df['release_year'],df['type']))
```

```
<Axes: xlabel='type', ylabel='release_year'>
```



### 5. Missing Value and Outlier Check

```
df.isna().sum()       #All missing values have been adjusted in the previous step
```

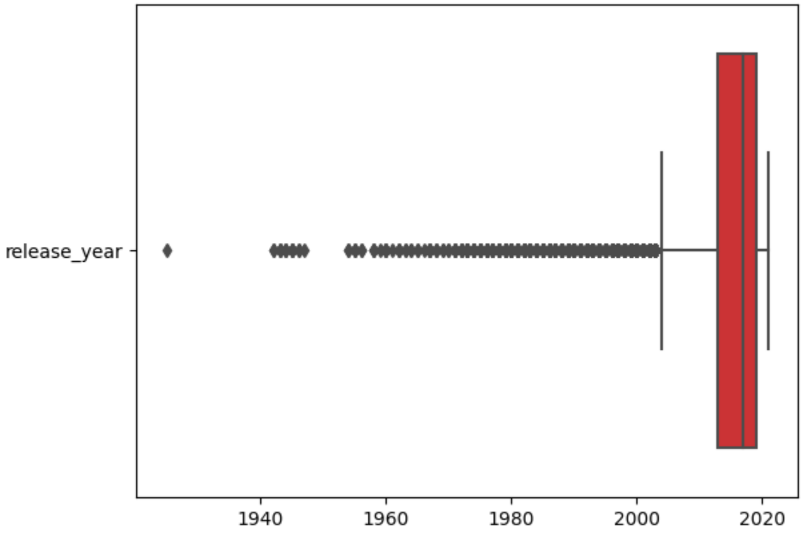
```
show_id         0
type            0
title           0
director        0
cast            0
country         0
date_added      0
release_year    0
rating          0
duration        0
listed_in       0
description     0
dtype: int64
```

```
sns.heatmap(df.isnull())
```

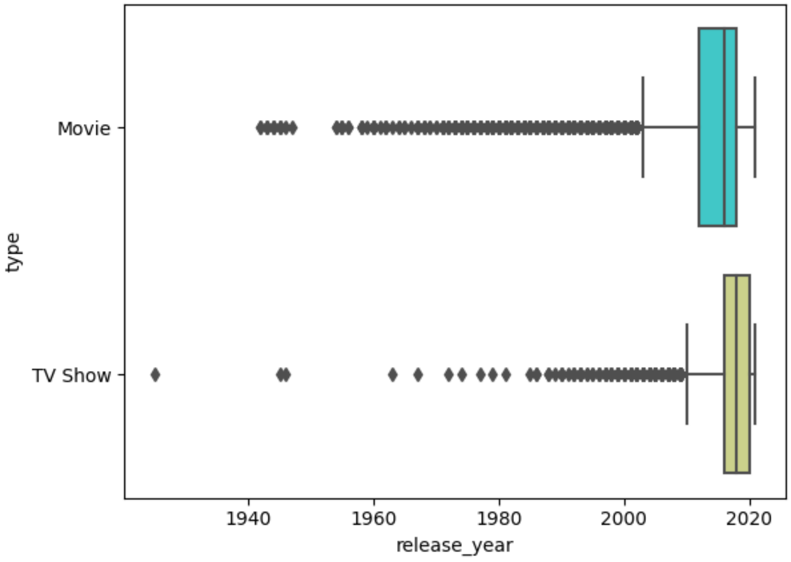
```
<Axes: >
```

```
sns.boxplot(data=df,palette='Set1',orient='h')
```

```
<Axes: >
```



```
sns.boxplot(x='release_year',y='type',data=df,orient='h',palette='rainbow')
plt.show()
```



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

### 6.1 Comments on the range of attributes

Presence of quantitive - Nominal, ordinal,binary makes the data more dynamic

```
a=df['release_year'].max()
a
```

```
     2021
```

```
b=df['release_year'].min()
b
```

```
     1925
```

```
range(a,b)
```

```
     range(2021, 1925)
```

```
df['duration'].max()
```

```
     '99 min'
```

```
df['duration'].min()
```

```
     '1 Season'
```

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

1. Most of the movies released post 2018 and USA is the largest content creator followed by India.
2. Movies listed on Netflix are more than the series.
3. Best Genre is drama and International series.
4. These contents are suitable for most of the viewers as per the listed genre.

**6.3 Comments for each Univariate and Bivariate plot**

1. Type Vs Counts Visual

From the visual we get to know that movie contents are more than TV Shows

2. Country Vs Count Plot

Most of the movies and TV shows are released from USA followed by India

3. rating Vs Count Plot

Most ratings given to TV-MA(3207) followed by TV-14(2106)

4. Distplot between year and density clearly indicates the density of movies released post 2018 are high probably due to Covid and the surge in Online content.
5. In the bivariate plot of type vs release_year contents prior to 2000 has outliers and most contents released during 2018-2021

## 7. Business Insights

1. From the pattern that is observed from the analysis above there are a lot of contents getting produced in USA followed by India and UK which caters to larger segment of Viewers.
2. Outliers prior to 2018 as there were less contents produced and post covid the amount of contents increased online.
3. Dramas, International Movies and Documentaries are most viewed contents on Netflix respectively.
4. Rajiv Chilaka and Campos have most contents on the platform as there is consumption of more internattional content.

## 8. Recommendations

1. Netflix has to target viewers as per the content viewed in different countries.
2. Most of the viewers are inclined towards drama and International Movies.
3. Recommendation system should send the contents as per the user pattern.
4. As there is surge in online contennts post covid,in order to retain and increase the customer base Netflix should have wide variety of contents which should be user specific.
5. If for a country certain age group loves to watch a particular contents, the platform should have enough contents to cater that segment.

❗ 0s completed at 8:27 PM ● ×