Business Case: Netflix - Data Exploration and Visualisation

About NETFLIX

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.

Business Problem

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.

Problem Statement:

Given the Netflix dataset, the problem statement could be to analyze the content available on Netflix and gain insights into its library. You may want to focus on understanding the distribution of content, popular genres, top-rated movies or TV shows, and trends over time. This can help Netflix make data-driven decisions for content acquisition and improve user experience.

In [1]:

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

In [2]:

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

In [3]:

```
df
```

Out[3]:

	show_id	type	title	director	cast	country	date_added	release_year	rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	September 24, 2021	2021	TV-MA
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila,	NaN	September 24, 2021	2021	TV-MA

	show_id	type	title	director	cast	country	date_added	release_year	rating
					Tracy Gotoas, Samuel Jouy, Nabi...				
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	September 24, 2021	2021	TV-MA
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA
...	...	...	...	...	...	...	...	...	...
8802	s8803	Movie	Zodiac	David Fincher	Mark Ruffalo, Jake Gyllenhaal, Robert Downey J...	United States	November 20, 2019	2007	R
8803	s8804	TV Show	Zombie Dumb	NaN	NaN	NaN	July 1, 2019	2018	TV-Y7
8804	s8805	Movie	Zombieland	Ruben Fleischer	Jesse Eisenberg, Woody Harrelson, Emma Stone, ...	United States	November 1, 2019	2009	R
8805	s8806	Movie	Zoom	Peter Hewitt	Tim Allen, Courteney Cox, Chevy Chase, Kate Ma...	United States	January 11, 2020	2006	PG
8806	s8807	Movie	Zubaan	Mozez Singh	Vicky Kaushal, Sarah-Jane Dias, Raaghav Chanan...	India	March 2, 2019	2015	TV-14

8807 rows × 12 columns

In [4]:

```
df.shape
```

Out[4]:
```
(8807, 12)
```

In [5]:
```
df.describe( include = 'all')
```

Out[5]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	du
count	8807	8807	8807	6173	7982	7976	8797	8807.000000	8803	
unique	8807	2	8807	4528	7692	748	1767	NaN	17	
top	s8260	Movie	The Evil Dead	Rajiv Chilaka	David Attenborough	United States	January 1, 2020	NaN	TV-MA	1
freq	1	6131	1	19	19	2818	109	NaN	3207	
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2014.180198	NaN	
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	8.819312	NaN	
min	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1925.000000	NaN	
25%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2013.000000	NaN	
50%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2017.000000	NaN	
75%	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2019.000000	NaN	
max	NaN	NaN	NaN	NaN	NaN	NaN	NaN	2021.000000	NaN	

In [6]:
```
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
```

In [7]:
```
df.nunique()
```

Out[7]:
```
show_id         8807
type               2
title           8807
director        4528
cast            7692
country          748
date_added      1767
```

```
release_year      74
rating            17
duration         220
listed_in        514
description     8775
dtype: int64
```

In [8]:

```
df.isnull().sum()/ len(df) * 100
```

Out[8]:

```
show_id          0.000000
type             0.000000
title            0.000000
director        29.908028
cast             9.367549
country          9.435676
date_added       0.113546
release_year     0.000000
rating           0.045418
duration         0.034064
listed_in        0.000000
description      0.000000
dtype: float64
```

In [9]:

```
# columns
df.columns
```

Out[9]:

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

Since some columns have nested values, will unnest them and prepare final dataset¶

In [10]:

```
# Un-nesting Directors  columns:

Dir_col = df['director'].apply(lambda x: str(x).split(", ")).tolist()
df_1 = pd.DataFrame(Dir_col, index = df['title'])
df_1 = df_1.stack()
df_1 = pd.DataFrame(df_1.reset_index())
df_1.rename(columns={0:'Directors'},inplace=True)
df_1 = df_1.drop(['level_1'],axis=1)
df_1.head(10)
```

Out[10]:

	title	Directors
0	Dick Johnson Is Dead	Kirsten Johnson
1	Blood & Water	nan
2	Ganglands	Julien Leclercq
3	Jailbirds New Orleans	nan
4	Kota Factory	nan
5	Midnight Mass	Mike Flanagan
6	My Little Pony: A New Generation	Robert Cullen
7	My Little Pony: A New Generation	José Luis Ucha
8	Sankofa	Haile Gerima
9	The Great British Baking Show	Andy Devonshire

In [11]:

```
# Un-nesting cast  columns:

cast_col = df['cast'].apply(lambda x: str(x).split(", ")).tolist()
df_2 = pd.DataFrame(cast_col, index = df['title'])
df_2 = df_2.stack()
df_2 = pd.DataFrame(df_2.reset_index())
df_2.rename(columns={0:'Actors'},inplace=True)
df_2 = df_2.drop(['level_1'],axis=1)
df_2.head(10)
```

Out[11]:

	title	Actors
0	Dick Johnson Is Dead	nan
1	Blood & Water	Ama Qamata
2	Blood & Water	Khosi Ngema
3	Blood & Water	Gail Mabalane
4	Blood & Water	Thabang Molaba
5	Blood & Water	Dillon Windvogel
6	Blood & Water	Natasha Thahane
7	Blood & Water	Arno Greeff
8	Blood & Water	Xolile Tshabalala
9	Blood & Water	Getmore Sithole

In [12]:

```
# Un-nesting listed_in  columns:

lst_col = df['listed_in'].apply(lambda x: str(x).split(", ")).tolist()
df_3 = pd.DataFrame(lst_col, index = df['title'])
df_3 = df_3.stack()
df_3 = pd.DataFrame(df_3.reset_index())
df_3.rename(columns={0:'Genre'},inplace=True)
df_3 = df_3.drop(['level_1'],axis=1)
df_3.head(10)
```

Out[12]:

	title	Genre
0	Dick Johnson Is Dead	Documentaries
1	Blood & Water	International TV Shows
2	Blood & Water	TV Dramas
3	Blood & Water	TV Mysteries
4	Ganglands	Crime TV Shows
5	Ganglands	International TV Shows
6	Ganglands	TV Action & Adventure
7	Jailbirds New Orleans	Docuseries
8	Jailbirds New Orleans	Reality TV
9	Kota Factory	International TV Shows

In [13]:

```
# Un-nesting country  columns:

country_col = df['country'].apply(lambda x: str(x).split(", ")).tolist()
df_4 = pd.DataFrame(country_col, index = df['title'])
df_4 = df_4.stack()
df_4 = pd.DataFrame(df_4.reset_index())
df_4.rename(columns={0:'Country'},inplace=True)
df_4 = df_4.drop(['level_1'],axis=1)
df_4.head(10)
```

Out[13]:

	title	Country
0	Dick Johnson Is Dead	United States
1	Blood & Water	South Africa
2	Ganglands	nan
3	Jailbirds New Orleans	nan
4	Kota Factory	India
5	Midnight Mass	nan
6	My Little Pony: A New Generation	nan
7	Sankofa	United States
8	Sankofa	Ghana
9	Sankofa	Burkina Faso

We are merging all un-nested dataframes using merge function

In [14]:

```
df_5 = df_2.merge(df_1,on=['title'],how='inner')

df_6 = df_5.merge(df_3,on=['title'],how='inner')

df_7 = df_6.merge(df_4,on=['title'],how='inner')

df_7.head()
```

Out[14]:

	title	Actors	Directors	Genre	Country
0	Dick Johnson Is Dead	nan	Kirsten Johnson	Documentaries	United States
1	Blood & Water	Ama Qamata	nan	International TV Shows	South Africa
2	Blood & Water	Ama Qamata	nan	TV Dramas	South Africa
3	Blood & Water	Ama Qamata	nan	TV Mysteries	South Africa
4	Blood & Water	Khosi Ngema	nan	International TV Shows	South Africa

In [15]:

```
df_7.shape
```

Out[15]:

```
(201991, 5)
```

Merging unnested data with the given dataframe

In [16]:

```
# merging unnested data with the given dataframe

df = df_7.merge(df[['show_id', 'type', 'title', 'date_added',
       'release_year', 'rating', 'duration']],on=['title'],how='left')
df.head()
```

Out[16]:

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year
0	Dick Johnson Is Dead	nan	Kirsten Johnson	Documentaries	United States	s1	Movie	September 25, 2021	2020
1	Blood & Water	Ama Qamata	nan	International TV Shows	South Africa	s2	TV Show	September 24, 2021	2021
2	Blood & Water	Ama Qamata	nan	TV Dramas	South Africa	s2	TV Show	September 24, 2021	2021
3	Blood & Water	Ama Qamata	nan	TV Mysteries	South Africa	s2	TV Show	September 24, 2021	2021
4	Blood & Water	Khosi Ngema	nan	International TV Shows	South Africa	s2	TV Show	September 24, 2021	2021

In [17]:

```
df.shape
```

Out[17]:

```
(201991, 11)
```

In [18]:

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

Out[18]:

```
title             0
Actors            0
Directors         0
Genre             0
Country           0
show_id           0
type              0
date_added      158
release_year      0
rating           67
duration          3
dtype: int64
```

There were some missing values will treat them

In [19]:

```
Total_null = df.isnull().sum().sort_values(ascending = False)
Percentage = ((df.isnull().sum()/df.isnull().count())*100).sort_values(ascending = F
print("Total records = ", df.shape[0])

missing_data = pd.concat([Total_null,Percentage.round(2)],axis=1,keys=['Total Missin
missing_data.head(10)
```

```
Total records =  201991
```

Out[19]:

	Total Missing	In Percentage
date_added	158	0.08
rating	67	0.03

	Total Missing	In Percentage
duration	3	0.00
title	0	0.00
Actors	0	0.00
Directors	0	0.00
Genre	0	0.00
Country	0	0.00
show_id	0	0.00
type	0	0.00

Above table gives missing values summary in absolute value and in Percentage, date added has the maximum missing values

Missing value treatment¶

In [20]:

```
# some columns having nan which is missing value, we have to replace

df['Actors'].replace(['nan'],['Unknown Actor'],inplace=True)
df['Directors'].replace(['nan'],['Unknown Director'],inplace=True)
df['Country'].replace(['nan'],[np.nan],inplace=True)
df.head()
```

Out[20]:

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year
0	Dick Johnson Is Dead	Unknown Actor	Kirsten Johnson	Documentaries	United States	s1	Movie	September 25, 2021	2020
1	Blood & Water	Ama Qamata	Unknown Director	International TV Shows	South Africa	s2	TV Show	September 24, 2021	2021
2	Blood & Water	Ama Qamata	Unknown Director	TV Dramas	South Africa	s2	TV Show	September 24, 2021	2021
3	Blood & Water	Ama Qamata	Unknown Director	TV Mysteries	South Africa	s2	TV Show	September 24, 2021	2021
4	Blood & Water	Khosi Ngema	Unknown Director	International TV Shows	South Africa	s2	TV Show	September 24, 2021	2021

In [21]:

```
Total_null = df.isnull().sum().sort_values(ascending = False)
Percentage = ((df.isnull().sum()/df.isnull().count())*100).sort_values(ascending = F
print("Total records = ", df.shape[0])

missing_data = pd.concat([Total_null,Percentage.round(2)],axis=1,keys=['Total Missin
missing_data.head(10)
```

```
Total records =  201991
```

Out[21]:

	Total Missing	In Percentage
Country	11897	5.89
date_added	158	0.08

	Total Missing	In Percentage
rating	67	0.03
duration	3	0.00
title	0	0.00
Actors	0	0.00
Directors	0	0.00
Genre	0	0.00
show_id	0	0.00
type	0	0.00

After replacing string nan with np.nan, actual null values of country went upto 5.89 %

In [22]:

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

Out[22]:

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_year	rat
126537	Louis C.K. 2017	Louis C.K.	Louis C.K.	Movies	United States	s5542	Movie	April 4, 2017	2017	
131603	Louis C.K.: Hilarious	Louis C.K.	Louis C.K.	Movies	United States	s5795	Movie	September 16, 2016	2010	
131737	Louis C.K.: Live at the Comedy Store	Louis C.K.	Louis C.K.	Movies	United States	s5814	Movie	August 15, 2016	2015	

duration and rating columns got messed up and values got exchanged will add rating column values into duration column missing values

In [23]:

```
df.loc[df['duration'].isnull(),'duration'] = df.loc[df['duration'].isnull(),'duratic
df.loc[df['rating'].str.contains('min', na=False),'rating'] = 'NR'
df['rating'].fillna('NR',inplace=True)
df.isnull().sum()
```

Out[23]:

```
title               0
Actors              0
Directors           0
Genre               0
Country         11897
show_id             0
type                0
date_added        158
release_year        0
rating              0
duration            0
dtype: int64
```

Filling missing values of date added column with mode value with respective release years

In [24]:

```python
for i in df[df['date_added'].isnull()]['release_year'].unique():
    date = df[df['release_year'] == i]['date_added'].mode().values[0]
    df.loc[df['release_year'] == i,'date_added'] = df.loc[df['release_year']==i,'dat
```

In [25]:

```python
df[df.Country.isna()]
```

Out[25]:

	title	Actors	Directors	Genre	Country	show_id	type	date_added	release_
58	Ganglands	Sami Bouajila	Julien Leclercq	Crime TV Shows	NaN	s3	TV Show	September 24, 2021	2
59	Ganglands	Sami Bouajila	Julien Leclercq	International TV Shows	NaN	s3	TV Show	September 24, 2021	2
60	Ganglands	Sami Bouajila	Julien Leclercq	TV Action & Adventure	NaN	s3	TV Show	September 24, 2021	2
61	Ganglands	Tracy Gotoas	Julien Leclercq	Crime TV Shows	NaN	s3	TV Show	September 24, 2021	2
62	Ganglands	Tracy Gotoas	Julien Leclercq	International TV Shows	NaN	s3	TV Show	September 24, 2021	2
...	...	...	...	...	...	...	...	...	
201424	YOM	Mayur Vyas	Unknown Director	Kids' TV	NaN	s8786	TV Show	June 7, 2018	2
201425	YOM	Ketan Kava	Unknown Director	Kids' TV	NaN	s8786	TV Show	June 7, 2018	2
201932	Zombie Dumb	Unknown Actor	Unknown Director	Kids' TV	NaN	s8804	TV Show	July 1, 2019	2
201933	Zombie Dumb	Unknown Actor	Unknown Director	Korean TV Shows	NaN	s8804	TV Show	July 1, 2019	2
201934	Zombie Dumb	Unknown Actor	Unknown Director	TV Comedies	NaN	s8804	TV Show	July 1, 2019	2

11897 rows × 11 columns

Filling missing values of country column with mode value with respective directors

In [26]:

```python
for i in df[df['Country'].isnull()]['Directors'].unique():
    if i in df[~df['Country'].isnull()]['Directors'].unique():
        country = df[df['Directors'] == i]['Country'].mode().values[0]
        df.loc[df['Directors'] == i,'Country'] = df.loc[df['Directors'] == i,'Countr
```

In [27]:

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

Out[27]:

```
title              0
Actors             0
Directors          0
Genre              0
Country         4276
show_id            0
type               0
date_added         0
release_year       0
```

```
rating            0
duration          0
dtype: int64
```

In [28]:

```
for i in df[df['Country'].isnull()]['Actors'].unique():
    if i in df[~df['Country'].isnull()]['Actors'].unique():
        imp = df[df['Actors'] == i]['Country'].mode().values[0]
        df.loc[df['Actors']==i, 'Country'] = df.loc[df['Actors'] == i, 'Country'].fi
```

In [29]:

```
df['Country'].fillna('Unknown Country',inplace=True)
df.isnull().sum()
```

Out[29]:

```
title           0
Actors          0
Directors       0
Genre           0
Country         0
show_id         0
type            0
date_added      0
release_year    0
rating          0
duration        0
dtype: int64
```

Remaining missing values will be replaced using actors column

Now missing values handling is over, will deep dive into data analysis

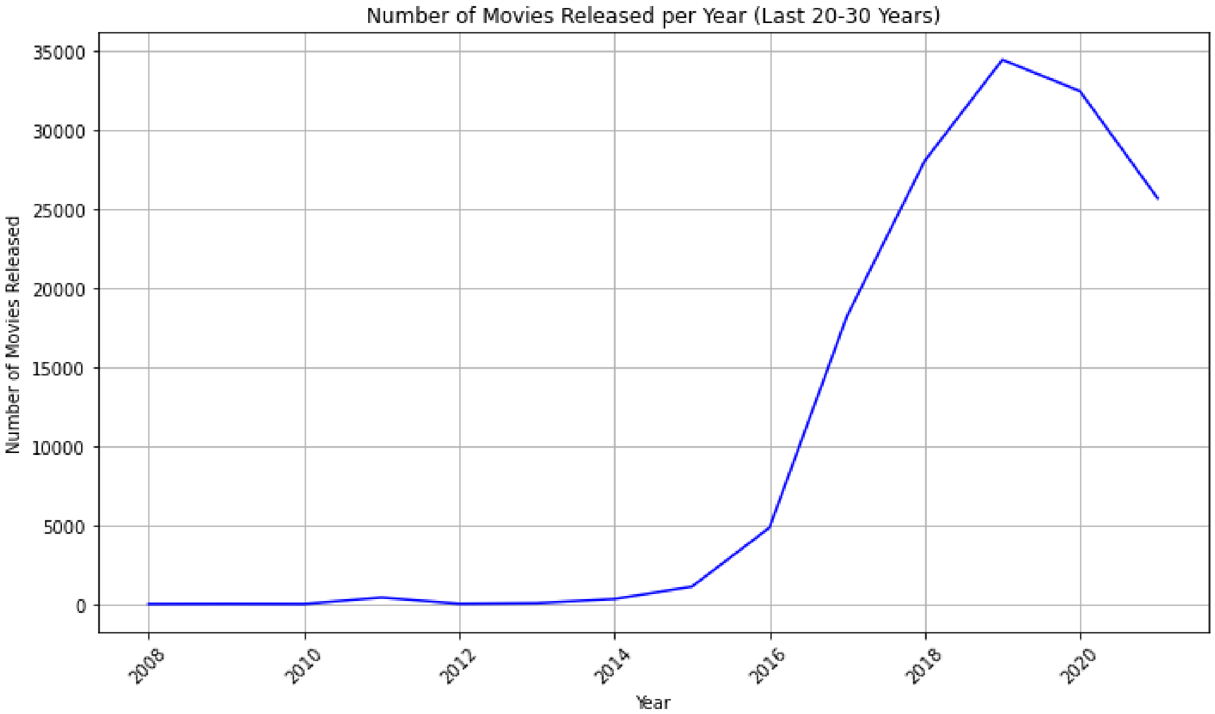
1. How has the number of movies released per year changed over the last 20-30 years?

In [30]:

```
df['date_added'] = pd.to_datetime(df['date_added'])
df['release_year'] = df['date_added'].dt.year
movies_df = df[df['type'] == 'Movie']
movies_per_year = movies_df['release_year'].value_counts().sort_index()
```

In [31]:

```
plt.figure(figsize=(10, 6))
plt.plot(movies_per_year.index, movies_per_year.values, color='b')
plt.xlabel('Year')
plt.ylabel('Number of Movies Released')
plt.title('Number of Movies Released per Year (Last 20-30 Years)')
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



According to line chart, from year 2014 there is rise in movies till 2019 but from 2019 there fall of movies.

2. Comparison of tv shows vs. movies.

In [32]:

```
# Filter the DataFrame for movies and TV shows

movies_df = df[df['type'] == 'Movie']
tv_shows_df = df[df['type'] == 'TV Show']
```

In [33]:

```
# Find the number of movies produced in each country and pick the top 10 countries

top_movies_countries = movies_df['Country'].value_counts().head(10)
```

In [34]:

```
# Find the number of TV shows produced in each country and pick the top 10 countries

top_tv_shows_countries = tv_shows_df['Country'].value_counts().head(10)
```

In [35]:

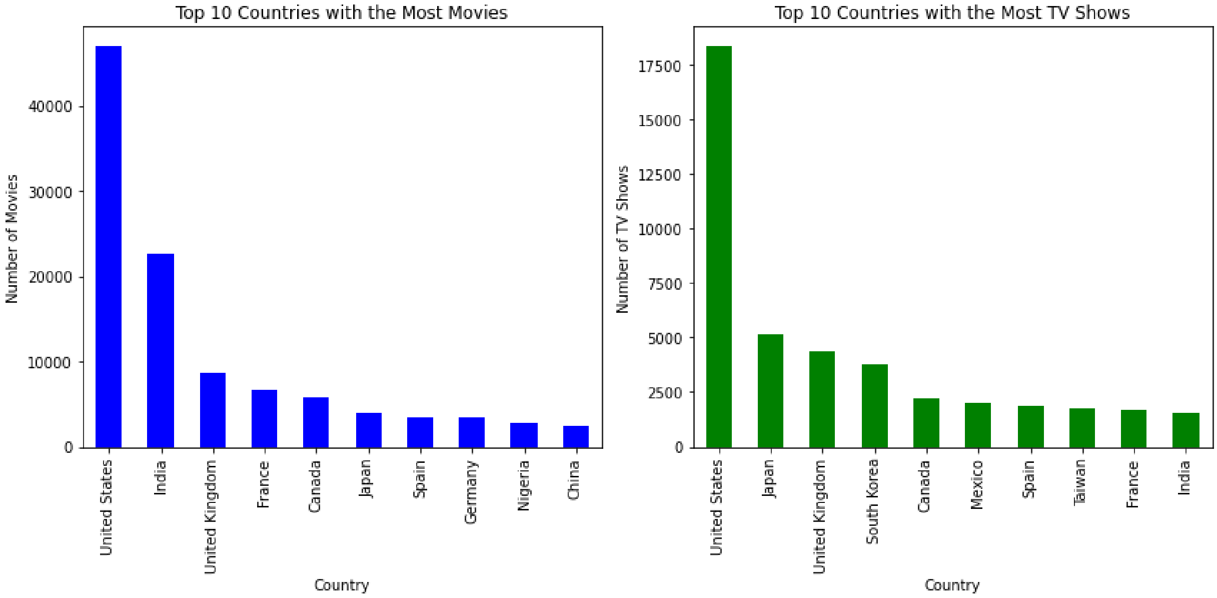
```
# Plot the visualization:

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

plt.subplot(1, 2, 1)
top_movies_countries.plot(kind='bar', color='b')
plt.xlabel('Country')
plt.ylabel('Number of Movies')
plt.title('Top 10 Countries with the Most Movies')

plt.subplot(1, 2, 2)
top_tv_shows_countries.plot(kind='bar', color='g')
plt.xlabel('Country')
plt.ylabel('Number of TV Shows')
plt.title('Top 10 Countries with the Most TV Shows')

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



According to the both the graphs, The United States is the country at the top for producing Movies compared to TV shows. Also, India is second for producing Movies. Japan and The United Kingdom are the top countries to produced the TV shows.

3. What is the best time to launch a TV show?

In [36]:

```
# Find which is the best month to release the Tv-show or the movie. Do the analysis
```

In [37]:

```
# Convert the "date_added" column to a datetime data type:

df['date_added'] = pd.to_datetime(df['date_added'])
```

In [38]:

```
# Separate the DataFrame into TV shows and movies:

tv_shows_df = df[df['type'] == 'TV Show']
movies_df = df[df['type'] == 'Movie']
```

In [39]:

```
# Group the data by month and calculate the count of TV shows and movies added in ea

tv_shows_by_month = tv_shows_df['date_added'].dt.month.value_counts().sort_index()
movies_by_month = movies_df['date_added'].dt.month.value_counts().sort_index()
```

In [44]:

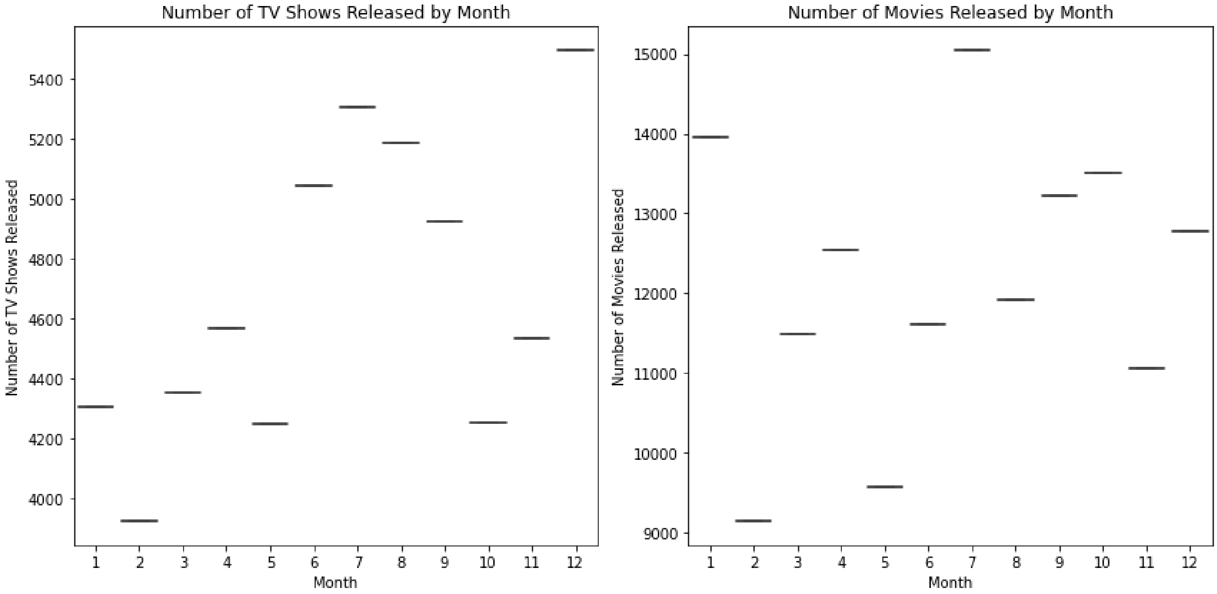
```
# Plot the results:

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

plt.subplot(1, 2, 1)
sns.boxplot(x=tv_shows_by_month.index, y=tv_shows_by_month.values, data = df)
plt.xlabel('Month')
plt.ylabel('Number of TV Shows Released')
plt.title('Number of TV Shows Released by Month')

plt.subplot(1, 2, 2)
sns.boxplot(x=movies_by_month.index, y=movies_by_month.values, data = df)
plt.xlabel('Month')
plt.ylabel('Number of Movies Released')
plt.title('Number of Movies Released by Month')

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



To launch TV shows month of December, July and August is best period. To launch Movie month of July, January and October is best period.

4. Analysis of actors of different types of movies.

In [41]:

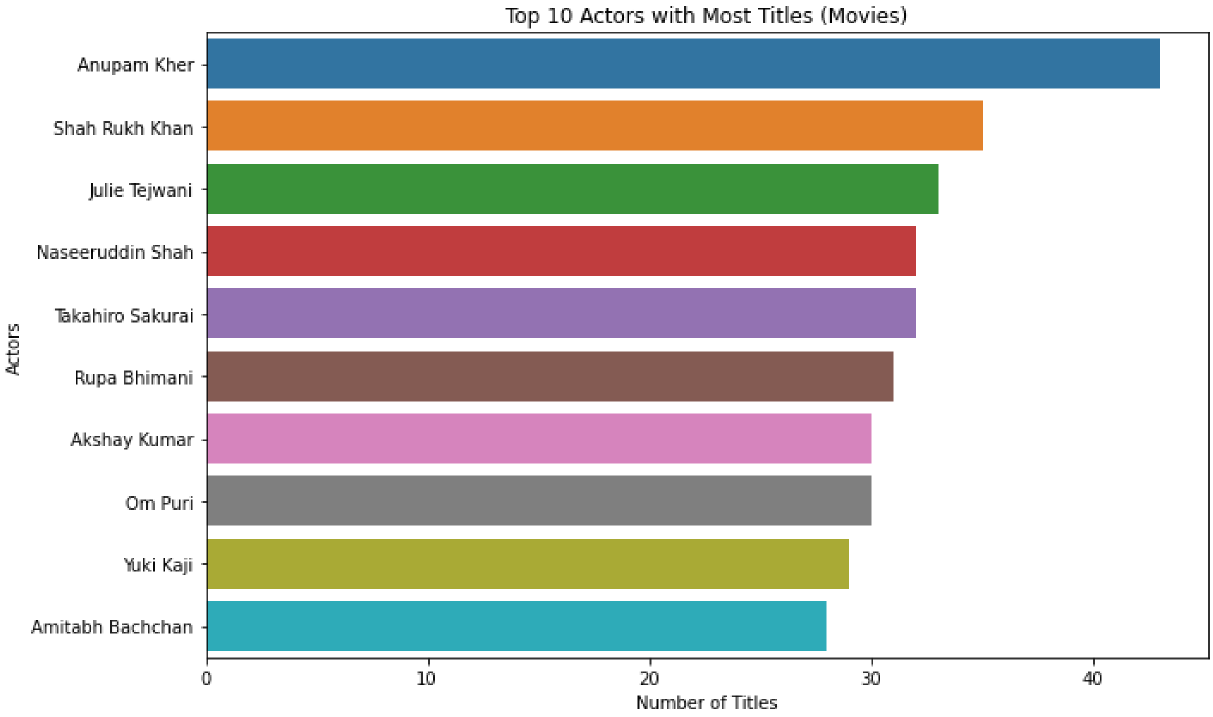
```
# Identify the top 10 actors who have appeared in most movies.

# Filter out the rows where the "actors" is not unknown:

df = df[df['Actors'] != 'Unknown Actor']

# Group the data by actor and count the number of unique titles:

actors_counts = df.groupby('Actors')['title'].nunique().sort_values(ascending=False)
top_10_actors = actors_counts.head(10)
# plot
plt.figure(figsize=(10, 6))
sns.barplot(x = top_10_actors.values, y = top_10_actors.index, data = df)
plt.xlabel('Number of Titles')
plt.ylabel('Actors')
plt.title('Top 10 Actors with Most Titles (Movies)')
plt.tight_layout()
plt.show()
```



Anupam Kher, Shah Rukh Khan, Naseeruddin Shah are most popular actors who appeared in the most movies. There is list of top 10 actors who are appeared most in the Bollywood movies.

5. Which genre movies are more popular or produced more

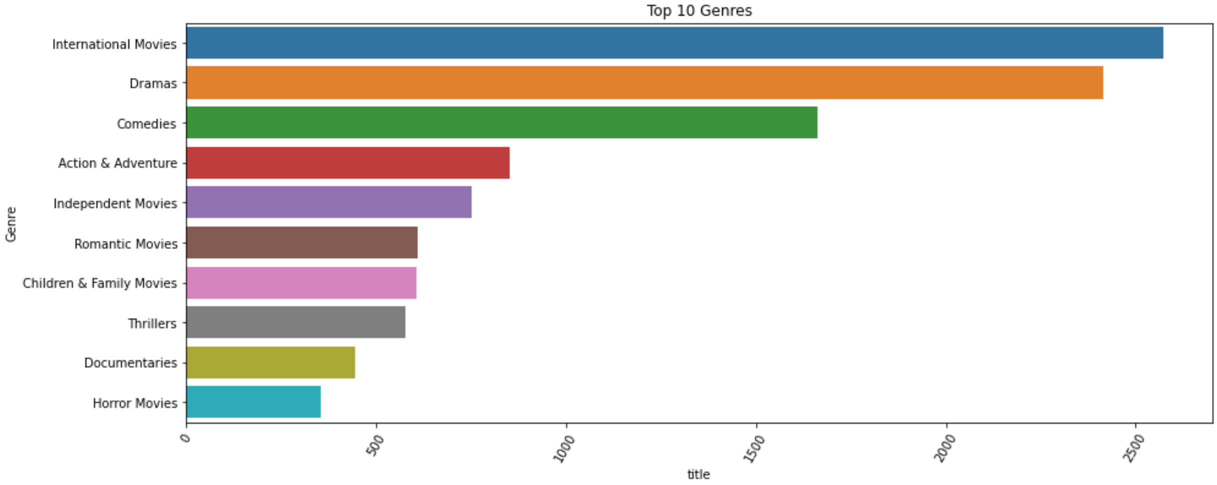
Univariate Analysis

In [42]:

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

df_movies = df[df['type']=='Movie']

df_genre = df_movies.groupby(['Genre']).agg({"title":"nunique"}).reset_index().sort_
plt.figure(figsize = (15,6))
sns.barplot(y = "Genre",x = 'title', data = df_genre)
plt.xticks(rotation = 60)
plt.title('Top 10 Genres')
plt.show()
```



International Movies, Dramas, Comedies are most popular genre as well as it more focused in recent years.

6. Understanding what content is available in different countries

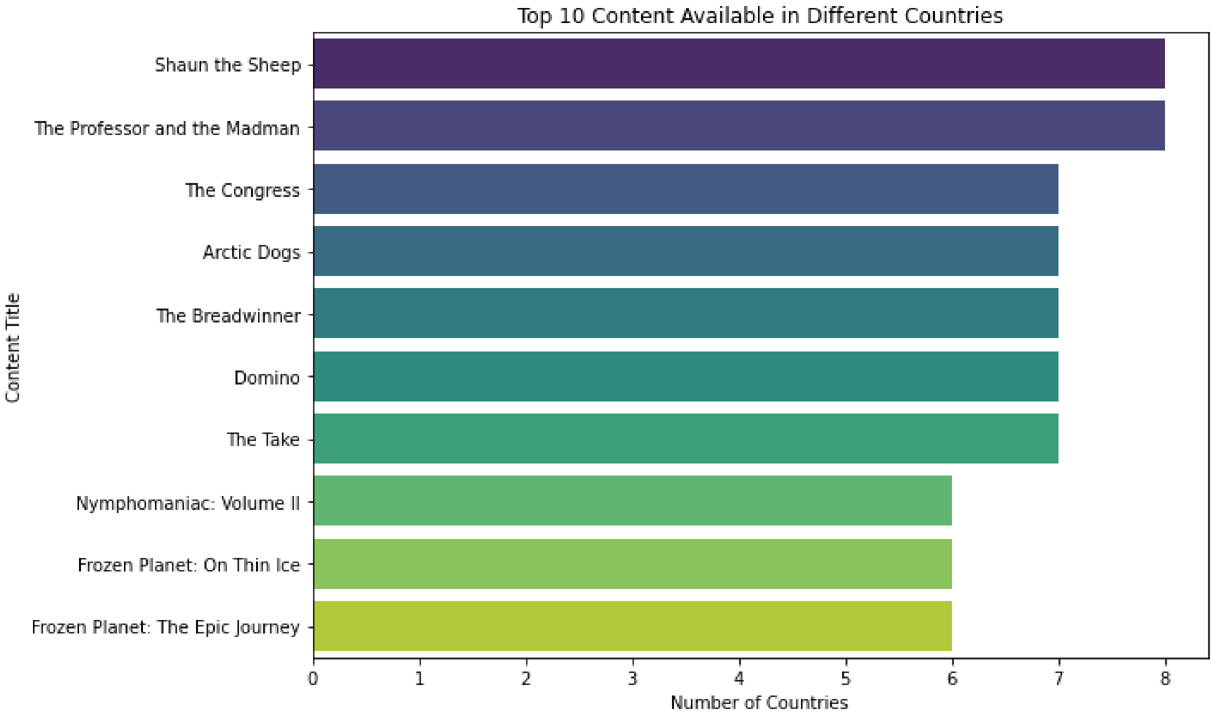
In [43]:

```
import pandas as pd
import matplotlib.pyplot as plt


# Group the data by content title and count the number of unique countries for each
content_countries_count = df.groupby('title')['Country'].nunique().reset_index()

# Sort the DataFrame in descending order based on the number of countries
top_10_content_countries = content_countries_count.sort_values(by='Country', ascendi

# Create the horizontal bar chart
plt.figure(figsize=(10, 6))
sns.barplot(x='Country', y='title', data=top_10_content_countries, palette='viridis'
plt.xlabel('Number of Countries')
plt.ylabel('Content Title')
plt.title('Top 10 Content Available in Different Countries')
plt.tight_layout()
plt.show()
```



From bar chart we can see the bar chart showing top 10 contents in different countries. The professor and the Madman is on the top for the content.

Business Insights :

A. Over the last 20-30 years, there is fluctuation in the movies released per year.

B. The United States is the country at the top for producing Movies compared to TV shows.

C. India is second for producing Movies.

D. Japan and The United Kingdom are the top countries to produced the TV shows.

E. To launch TV shows month of December, July and August is best period.

F. To launch Movie month of July, January and October is best period.

G. Anupam Kher, Shah Rukh Khan, Naseeruddin Shah are most popular actors who appeared in the most movies.

H. International Movies, Dramas, Comedies are most popular genre as well as it more focused in recent years.

I. The Professor and the Madman is on the top for understanding the content.

Recommendations:

A. Prevent the fluctuation of movies by good contents to movies. Add some good songs which can like by the old and new generation simultaneously. Give inspirational message to the public, so they can watch the movies at any cost.

B. Compared to movies give priority to TV shows. To do this create a TV shows of real incidents that can be useful for new generation such as kids, students, etc. For the country which is having low rating in TV shows, give more ads on social media so people can take interest to watch TV shows.

C. Add TV Shows/ movies in the month of February or May.

D. While creating content, take into consideration the popular actors/directors for that country. Also take into account the director-actor combination which is highly recommended.

E. Encourage the people to watch Documentaries and family movies/TV shows.