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 AND METRIC

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.

Here the basic Moto is to increase the Business.

INITIAL DATA EXPLORATION

In [1]:

```
## Importing Libraries

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

In [2]:

```
df = pd.read_csv('netflixnew.txt')
```

In [3]:

```
df.shape
```

Out[3]:

```
(8807, 12)
```

In [4]:

```
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(ll)
..,.,................. ,,,..... ,... 0-,i::, 0.   1/D
```

In [5]:

```
df.head(2)
```

Out[5]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson ls 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	Arna 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...

Insights -

- The Datset consist of 8807 records and 12 Columns.
- Columns like Director, Cast, Country, Date added, Rating, Duration have few null Values.
- Release Year Column has int data type, and rest all columns have object Data type.

DATA PREPROCESSING

In [6]:

```
def process_comma_separated_column(df, col):

    This method will expand columns which have comma separated values

    constraint= df[col].apply(lambda x:[i.strip() for i in str(x).split(',')]).tolist()
    df_new = pd.DataFrame(constraint, index= df['title'])
    df_new = df_new.stack()
    df_new = pd.DataFrame(df_new)
    df_new.reset_index(inplace = True)
    df_new.drop(axis= 1, columns= ['level_1'], inplace = True)
    return df new
```

Preprocessing Cast Column

In [7]:

```
df_cast = process_comma_separated_column(df, 'cast')
df_cast.rename(columns = {0:'cast'}, inplace = True)
df_cast.loc[df_cast['cast']=='nan', 'cast']=np.nan
df_cast.head(2)
```

Out[7]:

	title	cast
0	Dick Johnson Is Dead	NaN
1	Blood & Water	Arna Qamata

Preprocessing Listed_In Column

In [8]:

```
df_listedin = process_comma_separated_column(df, 'listed_in')
  df_listedin.rename(columns = {0:'listed_in'}, inplace  = True)
  df_listedin.head(2)
```

Out[8]:

	title	listed_in
0	Dick Johnson Is Dead	Documentaries
	Blood & Water	International TV Shows

Preprocessing Director Column

In [9]:

```
df_director = process_comma_separated_column(df, 'director')
df_director.rename(columns = {0: 'director'}, inplace = True)
df _director. loc[df_director ['director']==' nan', 'director'] =np. nan
df_director.head(2)
```

Out[9]:

	title	director
0	Dick Johnson Is Dead	Kirsten Johnson
	Blood & Water	NaN

Preprocessing Country Column

In [10]:

```
df_country = process_comma_separated_column(df, 'country')
df_country.rename(columns = {0: 'country'}, inplace = True)
df _country. lac [df_country ['country']==' nan', 'country'] =np. nan
df _country. lac [df_country ['country']==' ', 'country' ]=np. nan
df_country.head(2)
```

Out[10]:

	title	country
0	Dick Johnson Is Dead	United States
1	Blood & Water	South Africa

Combining all the Data Frames

In [11]:

```
df.drop([ 'director', 'cast', 'country', 'listed_in', 'description'], axis= 1, inplace = True)
df = pd.merge(left = df, right = df_cast, on = 'title', how = 'left')
df = pd.merge( left = df, right = df_listedin, on = 'title', how = 'left')
df = pd.merge(left = df, right = df_director, on = 'title', how = 'left')
df = pd.merge(left = df, right = df_country, on = 'title', how= 'left')
df.drop_duplicates(inplace = True)
```

Final Data Frame

In [12]:

```
df.head(2)
```

Out[12]:

	show_id	type	title	date_added	release_year	rating	duration	cast	listed_in	director	country
0	s1	Movie	Dick Johnson Is Dead	September 25, 2021	2020	PG-13	90 min	NaN	Documentaries	Kirsten Johnson	United States
1	s2	TV Show	Blood & Water	September 24, 2021	2021	TV-MA	2 Seasons	Arna Qamata	International TV Shows	NaN	South Africa

In [13]:

```
df.to_csv('Netflix_final.csv')
```

In [14]:

```
df = pd.read_csv('Netflix_final.csv')
```

In [15]:

```
df.drop('Unnamed: 0', axis= 1, inplace  =True)
```

In [16]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Rangeindex: 202010 entries, 0 to 202009
Data columns (total 11 columns):
 #   Column        Non-Null Count   Dtype
```

```
                     ----------------
 0   show id       202010 non-null  object
 1   type          202010 non-null  object
 2   title         202010 non-null  object
 3   date added    201852 non-null  object
 4   release_year  202010 non-null  int64
 5   rating        201943 non-null  object
 6   duration      202007 non-null  object
 7   cast          199861 non-null  object
 8   listed in     202010 non-null  object
 9   director      151367 non-null  object
 10  country       190007 non-null  object
dtypes: int64(1), object(10)
memory usage: 17.0+ MB
```

Dealing with null values

There are various ways of dealing with Null values -

- Dropping the rows with Null values
- Filling the null values with Mean, Median, Mode
- Treating the missing Values as separate Category

In This Case Sudy, I choose to treat the Missing values as a Separate Category, becuase by imputing them we might end up getting very different Analytical Insights.

Hence I replaced all the missing values as 'Unknown'

In [17]:

```
df['director'] = df['director'].fillna('Unknown')
df['country'] = df['country'].fillna('Unknown')
df['cast'] = df['cast'].fillna('Unknown')
df['rating'] = df['rating'].fillna('Unknown')
```

The Durations column has missing values only for Movies Data

In [18]:

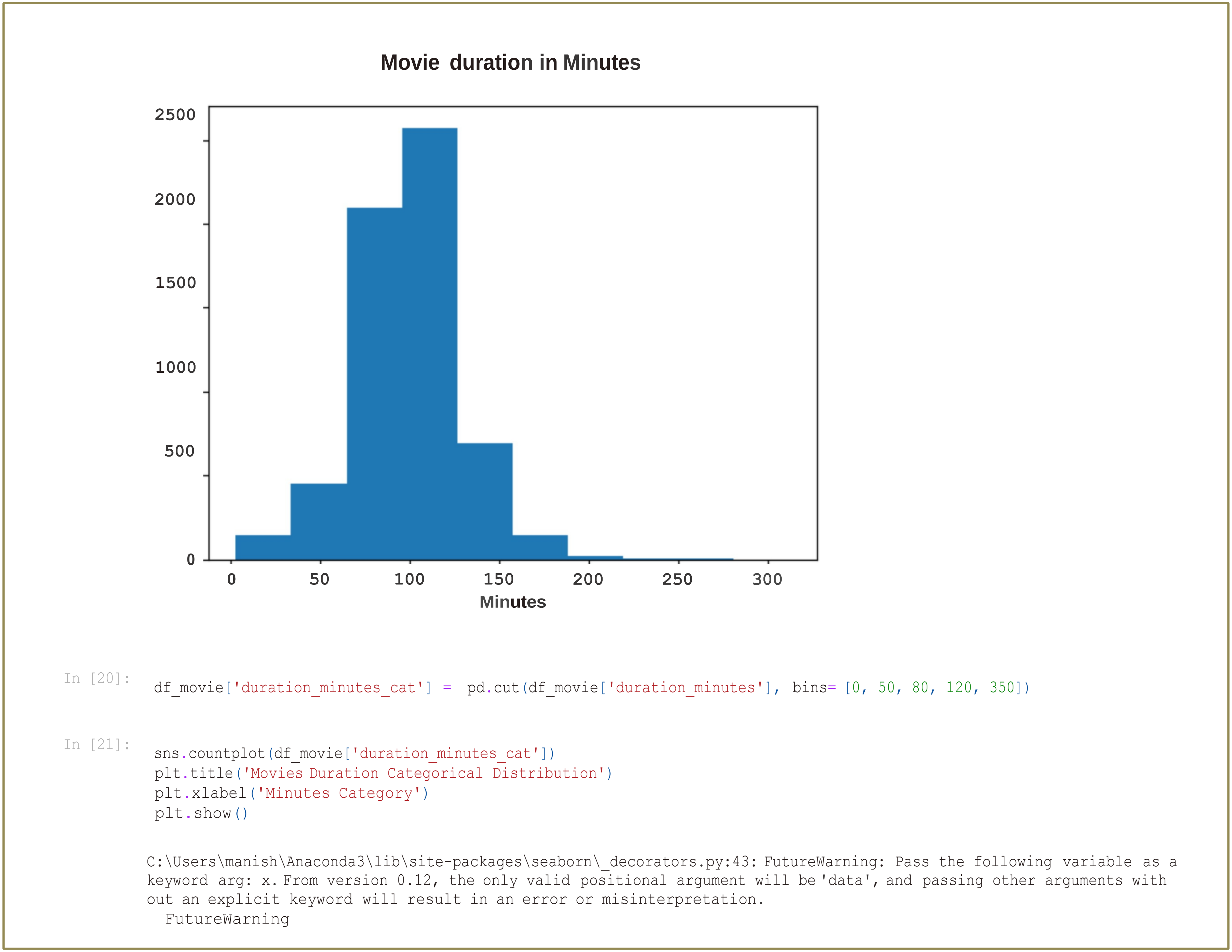
```
 ##  Extracting the Movie related Dataframe
df_movie  = df[df['type'] == 'Movie']
df_movie  =  df_movie ['title', 'duration']].drop_duplicates()

##  Extracting the mean of Movie Duration, for imputing the null values
mean duration= df_movie['duration'].str.split(' ', expand= True)[0].astype(float).mean()

##  Extracting the number of minutes
df_movie['duration_minutes']    = df_movie['duration'].str.split(' '   expand= True)[0].astype(float)

##  Imputing the null with Mean value
df_movie['duration_minutes'].fillna(mean_duration, inplace  = True)
```

DATA EXPLORATION

Question - The duration (in minutes) of most of the movies present on Netflix is between..

In [19]:

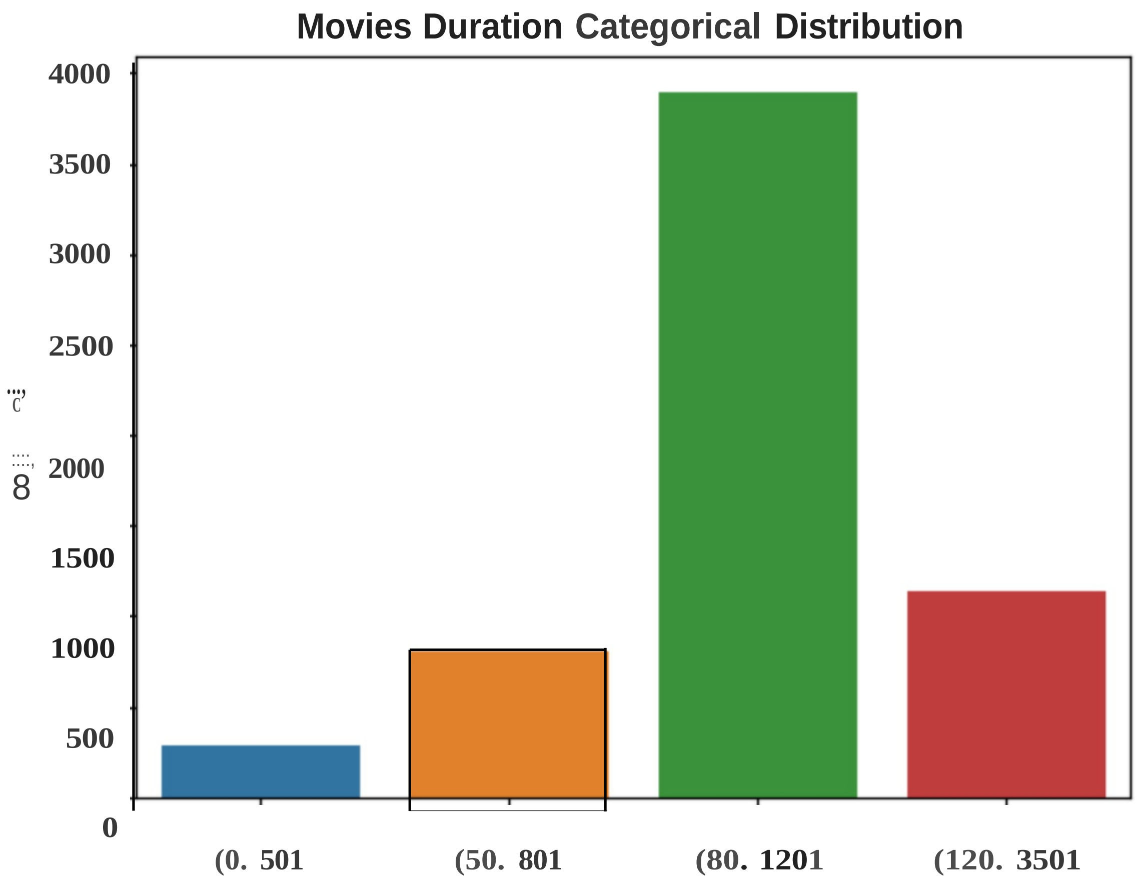
```
 plt.hist(df_movie['duration_minutes'])
plt.title('Movie duration  in  Minutes')
plt.xlabel('Minutes')
plt.show()
```



```
In [20]:  df_movie['duration_minutes_cat'] =  pd.cut(df_movie['duration_minutes'], bins= [0, 50, 80, 120, 350])
```

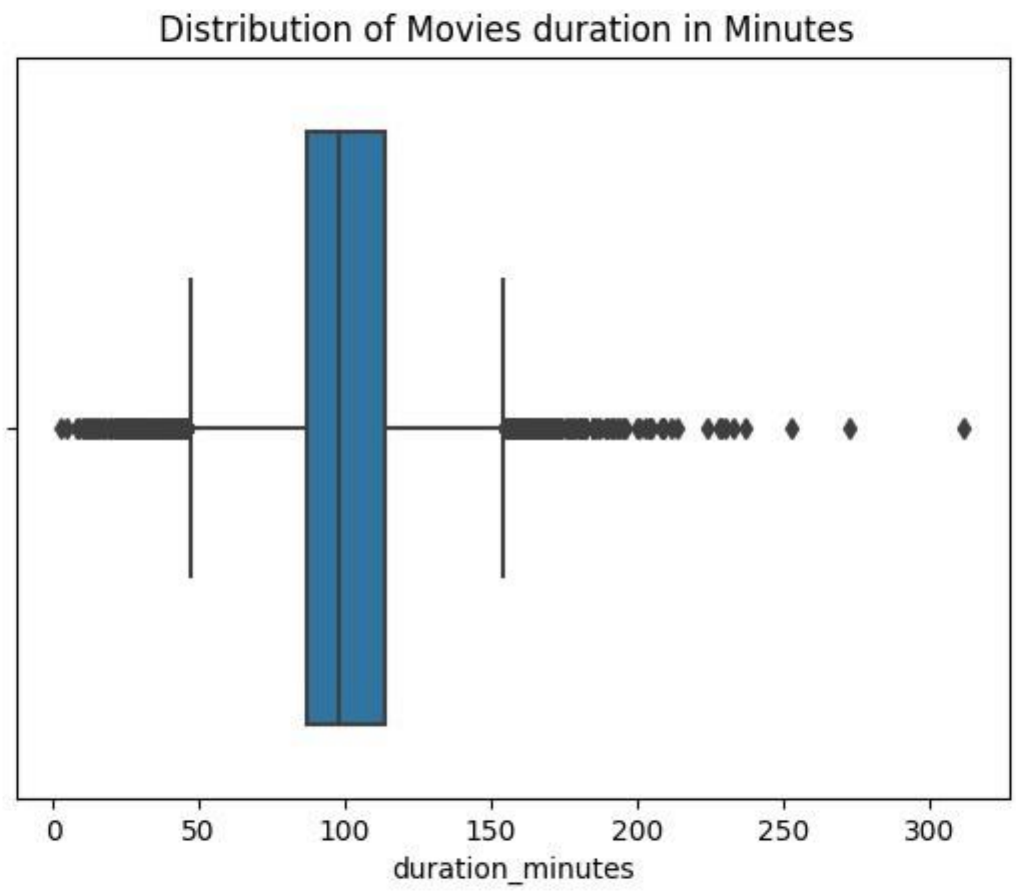
```
In [21]:  sns.countplot(df_movie['duration_minutes_cat'])
          plt.title('Movies Duration Categorical Distribution')
          plt.xlabel('Minutes Category')
          plt.show()
```

```
C:\Users\manish\Anaconda3\lib\site-packages\seaborn\_decorators.py:43: FutureWarning: Pass the following variable as a
keyword arg: x. From version 0.12, the only valid positional argument will be 'data', and passing other arguments with
out an explicit keyword will result in an error or misinterpretation.
  FutureWarning
```



In [22]:

```
sns.boxplot(df_movie['duration_minutes'])
plt.title('Distribution of Movies duration in Minutes ')
plt.show()
```



Insights -

- Most of the Movies have duration of around 100 Minutes.
- There are few Movies which are either very small and very lengthy, which are marked as outlier in box plot as shown above

Business Recomnedations

Most of the movies have duration in range of 80-120 Minutes (as seen in the Bar plot). The business could experiment by promoting short films, because in recent times Reels (short video format) is becoming popular on other platforms

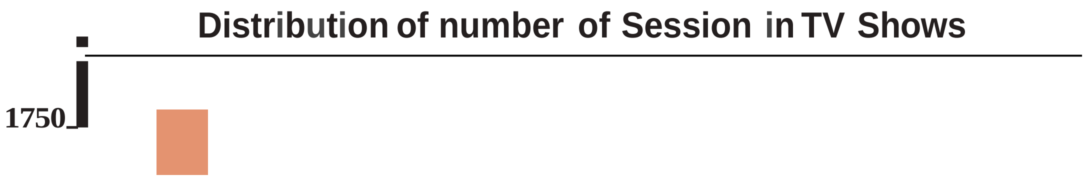
Question - The Number of Sessions a TV Show has on Netflix..

```
In [23]:  df_tvshow = df[df['type'] == 'TV Show']
          df_tvshow = df_tvshow[['title', 'duration']].drop_duplicates()
          median_shows = df_tvshow['duration'].str.split(' ', expand= True)[0].astype(int).median()
          print('The Median of number of Sesions on Netflix =', median_shows)
```

```
The Median of number of Sesions on Netflix = 1.0
```

```
In [24]:  sns.countplot(df_tvshow['duration'])
          plt.xticks(rotation = 45)
          plt.title('Distribution of number of Session in TV Shows')
          plt.show()
```

```
C:\Users\manish\Anaconda3\lib\site-packages\seaborn\_decorators.py:43: FutureWarning: Pass the following variable as a
keyword arg: x. From version 0.12, the only valid positional argument will be 'data', and passing other arguments with
out an explicit keyword will result in an error or misinterpretation.
  FutureWarning
```



Insights -

- Most of the TV Shows have 1 Season, followed by 2 and 3 Season.
- There are very few TV Shows which have greater than 3 Season.

DATA EXPLORATION

Question - What is the percentage of TV Shows and Movies Overall? (Comparison of tv shows vs. movies)

In [25]:

```
df_type_title = df[['type', 'title']].drop_duplicates()
df_type_title['type'].value_counts(normalize = True)
```

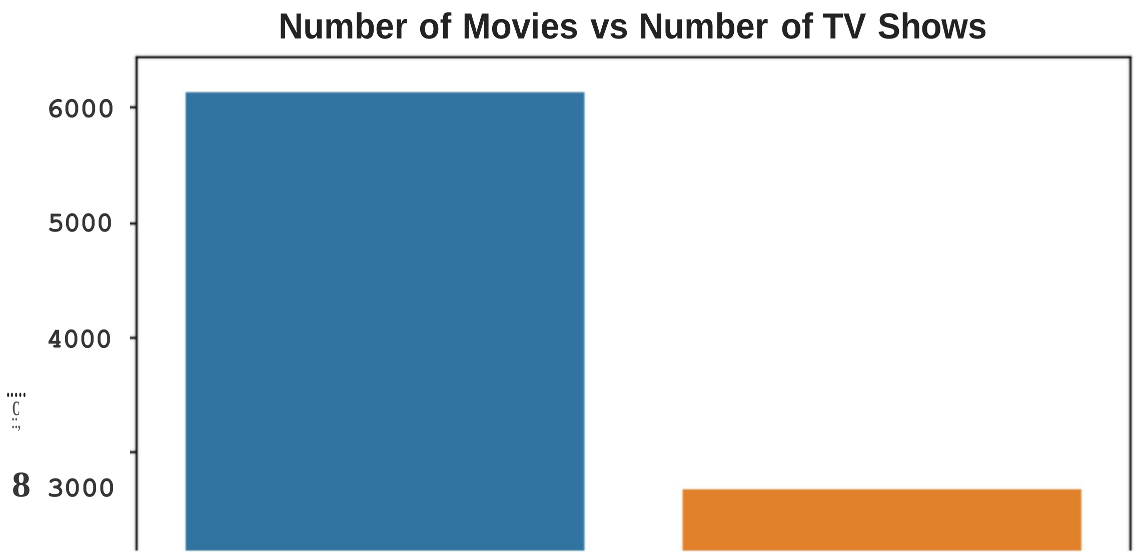
Out[25]:

```
Movie      0.696151
TV Show    0.303849
Name: type, dtype: float64
```

In [26]:

```
sns.countplot(df_type_title['type'])
plt.title('Number of Movies vs Number of TV Shows')
plt.show()
```

```
C:\Users\manish\Anaconda3\lib\site-packages\seaborn\_decorators.py:43: FutureWarning: Pass the following variable as a
keyword arg: x. From version 0.12, the only valid positional argument will be 'data', and passing other arguments with
out an explicit keyword will result in an error or misinterpretation.
  FutureWarning
```



Insights -

69.61% of Shows are Movies and 30.38% of Shows are TV Shows

Question - What are the top countries with most number of releases of TV Shows and Movies ?

Movies

In [27]:

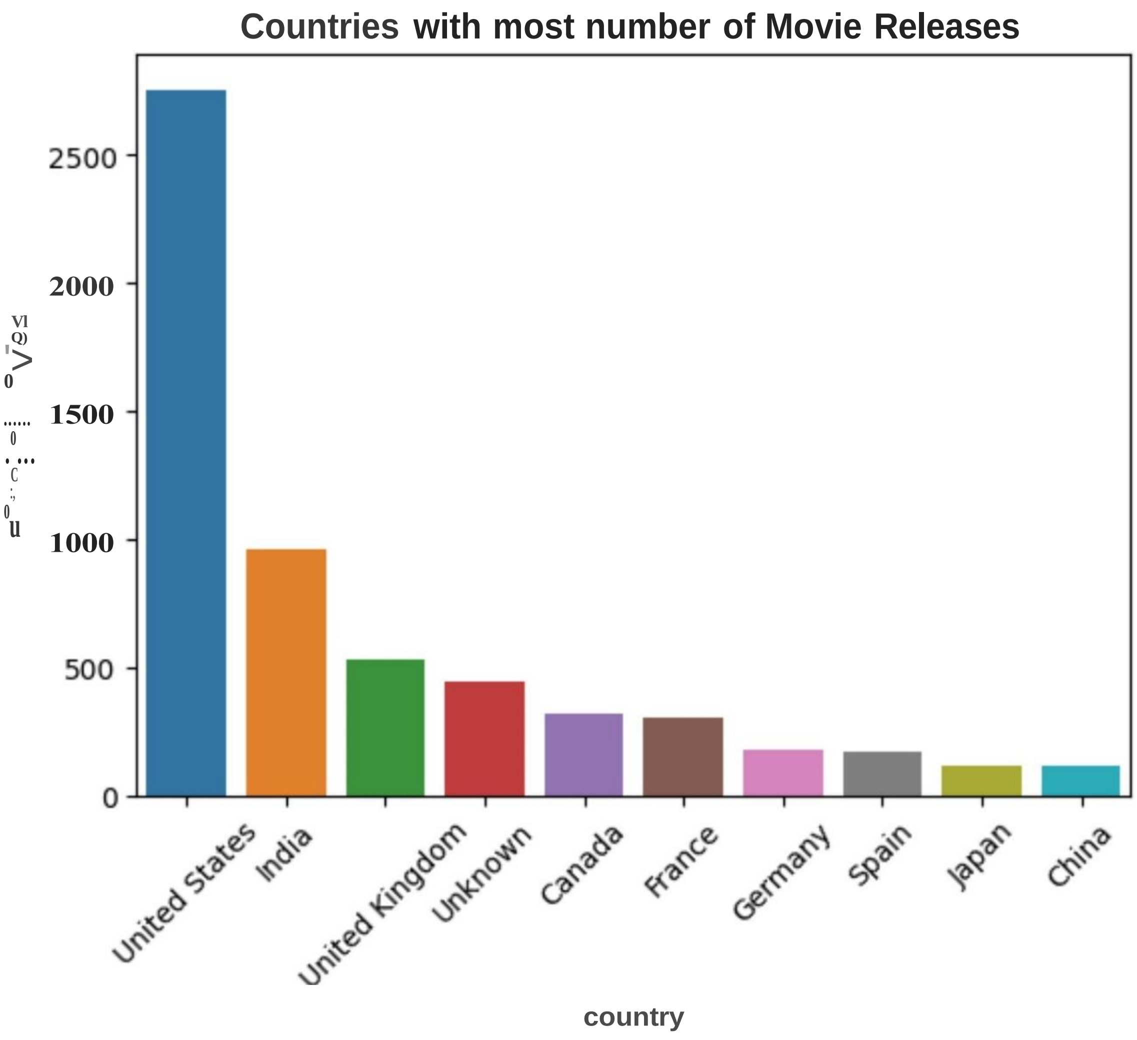
```
df_movies = df.loc[df['type'] == 'Movie']
df_movies_country = df_movies[['title','country']].drop_duplicates()
d = df_movies_country.groupby('country').count().reset_index().sort_values(by = 'title', ascending= False).head(10)
d.columns = ['country', 'count']
d
```

Out[27]:

	country	count
110	United States	2752
41	India	962
109	United Kingdom	534
111	Unknown	446
18	Canada	319

	country	count
32	France	303
34	Germany	182
97	Spain	171
,tQ	l::an::an	11 a

```
In [28]: sns.barplot(data = d, x = d['country'], y = d['count'])
         plt.xticks(rotation = 45)
         plt.title('Countries with most number of Movie Releases')
         plt.ylabel('Count of Movies')
         plt.show()
```



TV Shows

In [29]:

```
df_tvshows = df.loc[df['type'] == 'TV Show']
df_tvshows_country = df_tvshows[['title','country']].drop_duplicates()
d  =  df_tvshows_country.groupby('country').count().reset_index().sort_values(by = 'title', ascending= False).head(10)
d.columns = ['country', 'count']
d
```

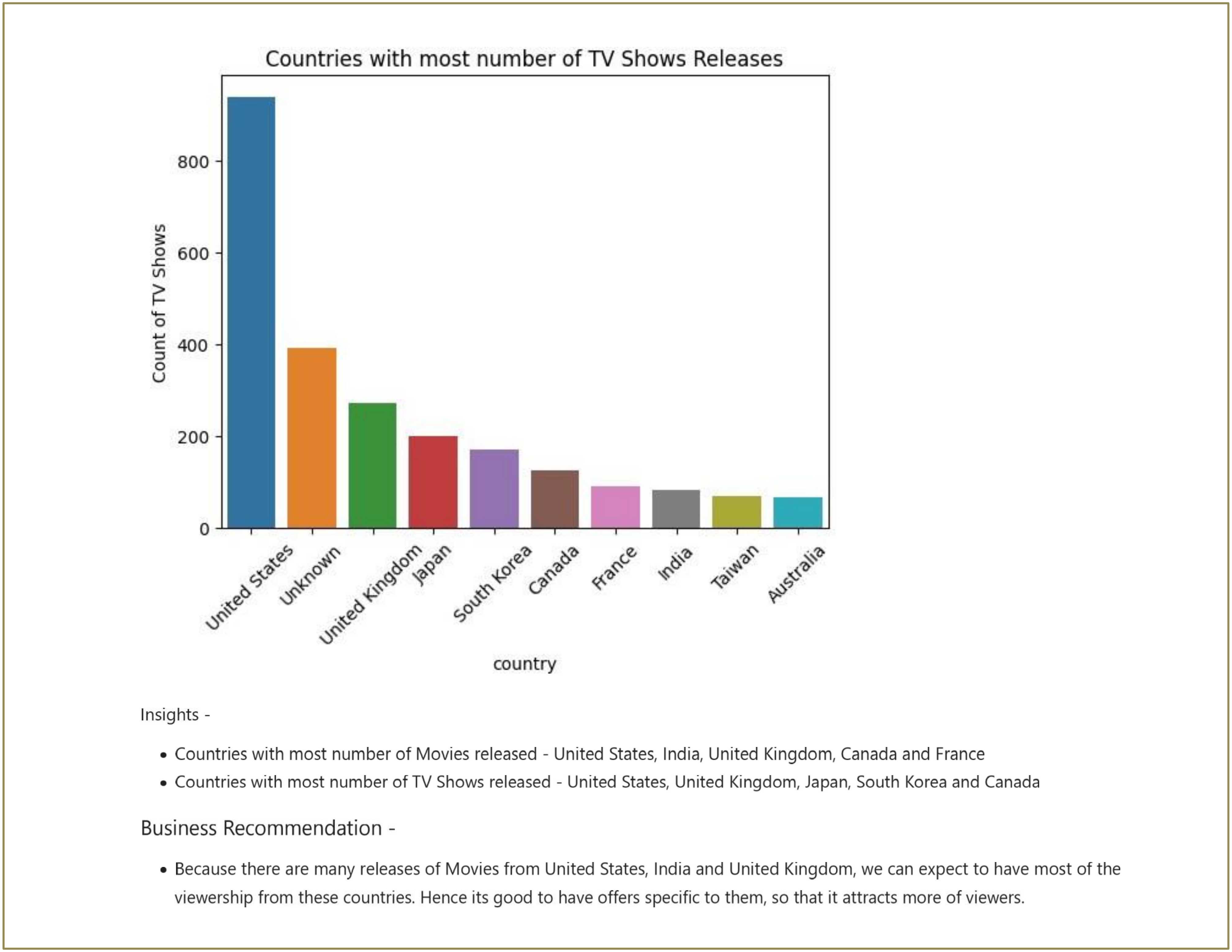
Out[29]:

country	count

	country	count
62	United States	938
63	Unknown	392
61	United Kingdom	272
29	Japan	199
51	South Korea	170
7	Canada	126
18	France	90
24	India	84
56	Taiwan	70

In [30]:

```
sns.barplot(data = d, x = d['country'], y = d['count'])
plt.xticks(rotation = 45)
plt.title('Countries with most number of TV Shows Releases')
plt.ylabel('Count of TV Shows')
plt.show()
```



Insights -

- Countries with most number of Movies released - United States, India, United Kingdom, Canada and France
- Countries with most number of TV Shows released - United States, United Kingdom, Japan, South Korea and Canada

Business Recommendation -

- Because there are many releases of Movies from United States, India and United Kingdom, we can expect to have most of the viewership from these countries. Hence its good to have offers specific to them, so that it attracts more of viewers.

Question - What are the top countries with most number of releases in the past 5 years?

In [31]:

```
df_country_releases = df[['title', 'country', 'release_year']].drop_duplicates()
df_releases = pd.crosstab(df_country_releases['country'], columns= df_country_releases['release_year']).reset_index()
df_releases['average_releases'] = df_releases.mean(axis=1).round(2)
```

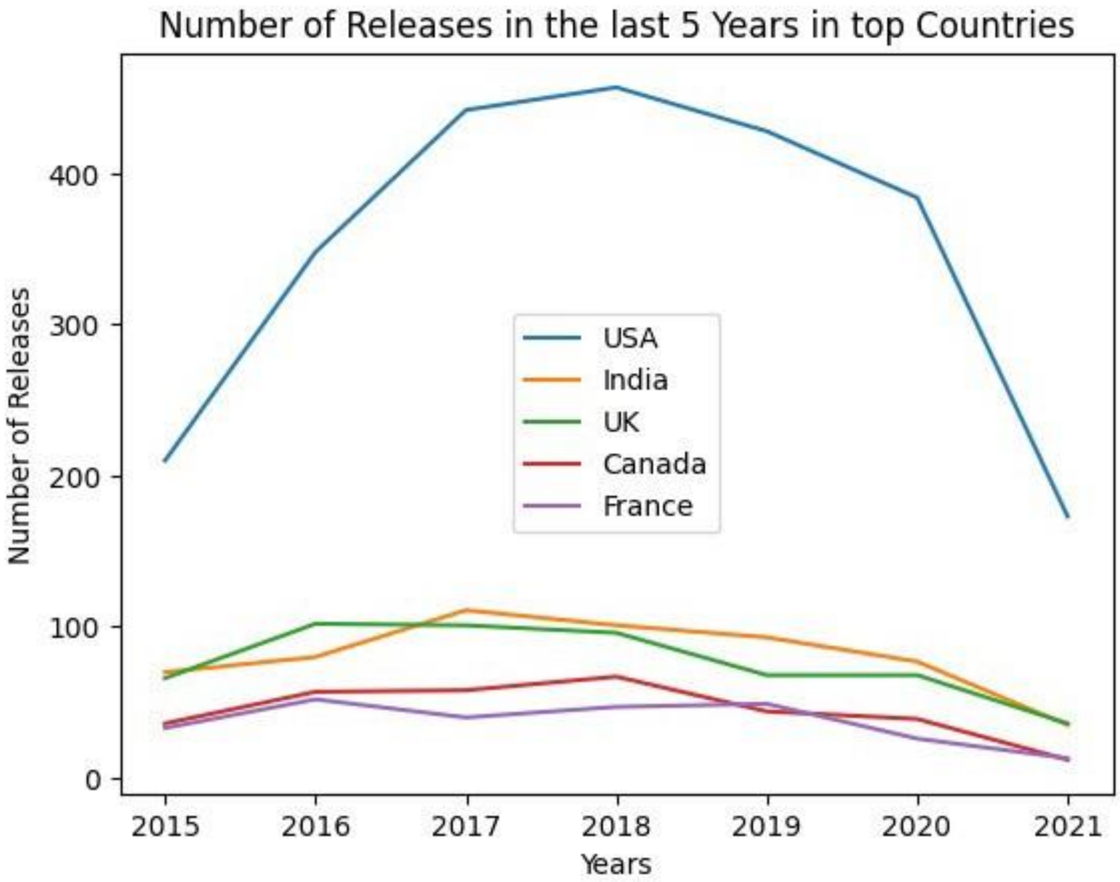
In [32]:

```
d =  df_releases.sort_values(by  = 'average_releases', ascending =  False).head(10).reset_index().drop('index',  axis  = 1)
d
```

Out[32]:

release_year	country	2015	2016	2017	2018	2019	2020	2021	average_releases
0	United States	210	348	442	457	428	384	173	348.86
1	Unknown	44	64	66	111	117	102	210	102.00
2	India	70	80	111	101	93	77	35	81.00
3	United Kingdom	66	102	101	96	68	68	36	76.71
4	Canada	36	57	58	67	44	39	12	44.71
5	France	33	52	40	47	49	26	13	37.14
6	Spain	16	31	33	46	32	31	16	29.29
7	Japan	16	25	37	49	36	24	15	28.86
8	South Korea	16	36	33	34	27	31	20	28.14
9	Mexico	9	23	20	25	25	23	13	19.71

```
In [33]:
yl = d[[2015, 2016, 2017, 2018, 2019, 2020, 2021]].loc[0].values #USA
y2  =  d[[2015, 2016, 2017, 2018, 2019, 2020, 2021]].loc[2].values #India
y3  =  d[[2015, 2016, 2017, 2018, 2019, 2020, 2021]].loc[3].values #UK
y4 = d[[2015, 2016, 2017, 2018, 2019, 2020, 2021]].loc[4].values #Canada
y5  =  d[[2015, 2016, 2017, 2018, 2019, 2020, 2021]].loc[5].values #France
x = [2015, 2016, 2017, 2018, 2019, 2020, 2021]

sns.lineplot(x = x, y = yl, label = 'USA')
sns.lineplot(x = x, y = y2, label = 'India')
sns.lineplot(x = x, y = y3, label = 'UK')
sns.lineplot(x = x, y = y4, label = 'Canada')
sns.lineplot(x = x, y = y5, label = 'France')
plt.title('Number of Releases in the last 5 Years  in top Countries')
plt.ylabel('Number of Releases')
plt.xlabel('Years')
plt.show()
```



Insights -

- United States tops the list with most number of avareage releases in the last five years, followed by India and United Kingdom

Business Recomendation

- The Number of releases has a slight decreasing trend. If this continues, Netflix might have lesser content to release over its channel.
- So its recommended to try to increases the releases, this can be done by sponsoring and Producing Movies.

Question - Number of releases per year over the last years 20 - 30 year

In [34]:

```
plt.rcParams["figure.figsize"] = (15,6)
```

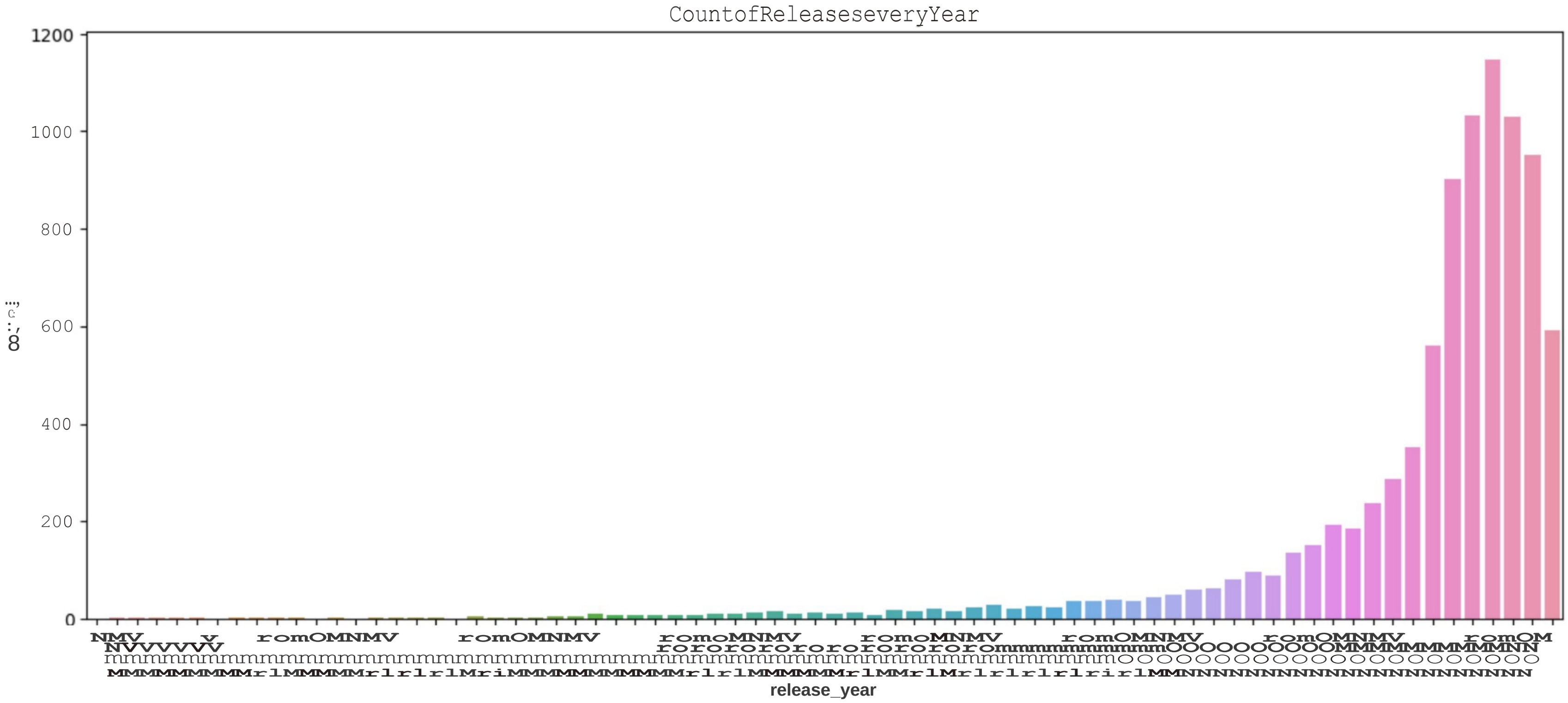
In [35]:

```
df_title_year = df[['title','release_year', 'type']].drop_duplicates()
```

In [36]:

```
sns.countplot(df_title_year['release_year'])
plt.xticks(rotation=90)
plt.title('Count of Releases every Year')
plt.show()
```

```
C:\Users\manish\Anaconda3\lib\site-packages\seaborn\_decorators.py:43: FutureWarning: Pass the following variable as a
keyword arg: x. From version 0.12, the only valid positional argument will be 'data', and passing other arguments with
out an explicit keyword will result in an error or misinterpretation.
  FutureWarning
```



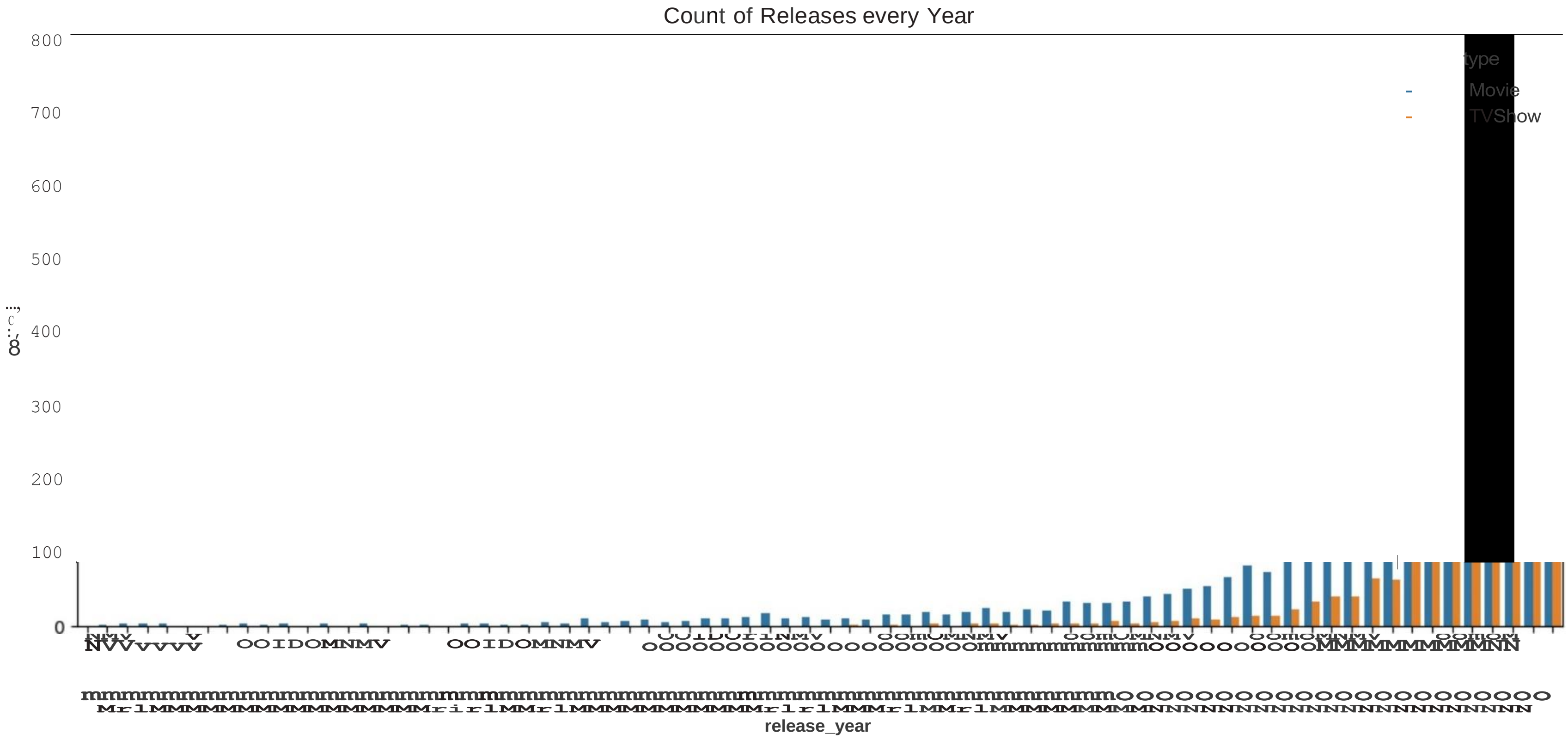
Insights -

- The Number of releases every year increased very gradually in the past. A good jump in growth was seen from 2015.
- The Number of releases every year started to decrease afer reaching the peak in the year 2018. The number of releases decreased

In [37]:

```
sns.countplot(df_title_year['release_year'], hue= df['type'])
plt.xticks(rotation=90)
plt.title('Count of Releases every Year')
plt.show()
```

```
C:\Users\manish\Anaconda3\lib\site-packages\seaborn\_decorators.py:43: FutureWarning: Pass the following variable as a
keyword arg: x. From version 0.12, the only valid positional argument will be 'data', and passing other arguments with
out an explicit keyword will result in an error or misinterpretation.
  FutureWarning
```



Insights -

- The number of movies released every year was always greater than number of TV series until 2020. But this trend changed and more number of TV Series were released than Movies in the year 2021.

- The most probable reason could be that after pandamic, the Movie theaters were not operational over a long period, and the OTT culture saw a good growth during this period, giving rise to more number of TV shows getting released.

Question - What is the best time to launch a TV show and Movie?

In [39]:

```python
df_title_dateAdded = df[['title','date_added', 'type']].drop_duplicates()
```

In [40]:

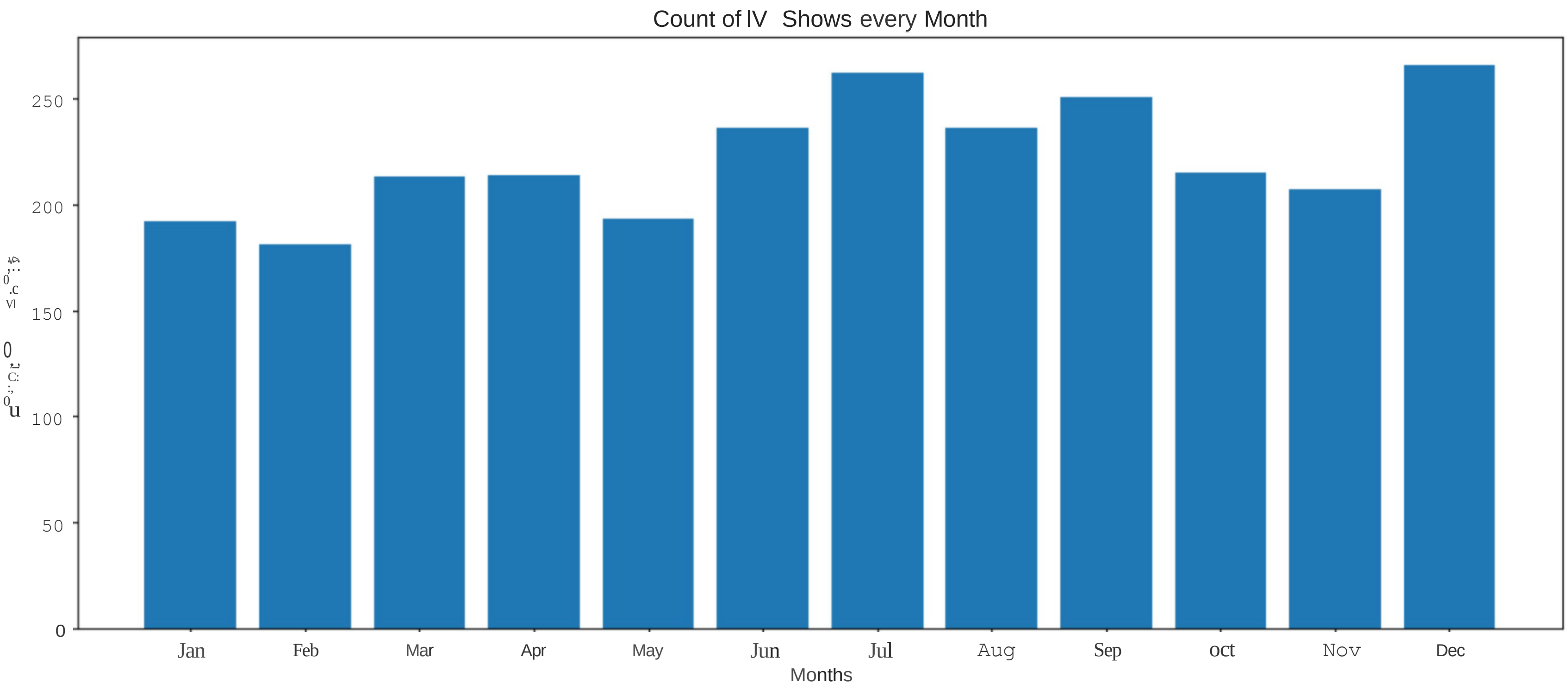
```python
# Extracting the month from date_added
df_title_dateAdded['date_added'] = pd.to_datetime(df_title_dateAdded['date_added'])
df_title_dateAdded['month_added'] = df_title_dateAdded['date_added'].dt.month
df_title_dateAdded.head(2)
```

Out[40]:

	title	date_added	type	month_added
0	Dick Johnson Is Dead	2021-09-25	Movie	9.0
	Blood & Water	2021-09-24	TV Show	9.0

In [41]:

```python
df_tvshow = df_title_dateAdded.loc[df_title_dateAdded['type']=='TV Show', :]
month_added = df_tvshow['month_added'].value_counts().reset_index().sort_values('index')
x = month_added['index']
y = month_added['month_added'].values

plt.bar(x, height= y)
plt.xticks([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'O
plt.title('Count of TV Shows every Month')
plt.xlabel('Months')
plt.ylabel('Count of TV Shows')
plt.show()
```



In [43]:

```
d = df_tvshow['month_added'].value_counts().reset_index()
 d.columns = ['Month', 'TVShows Added']
 d
```

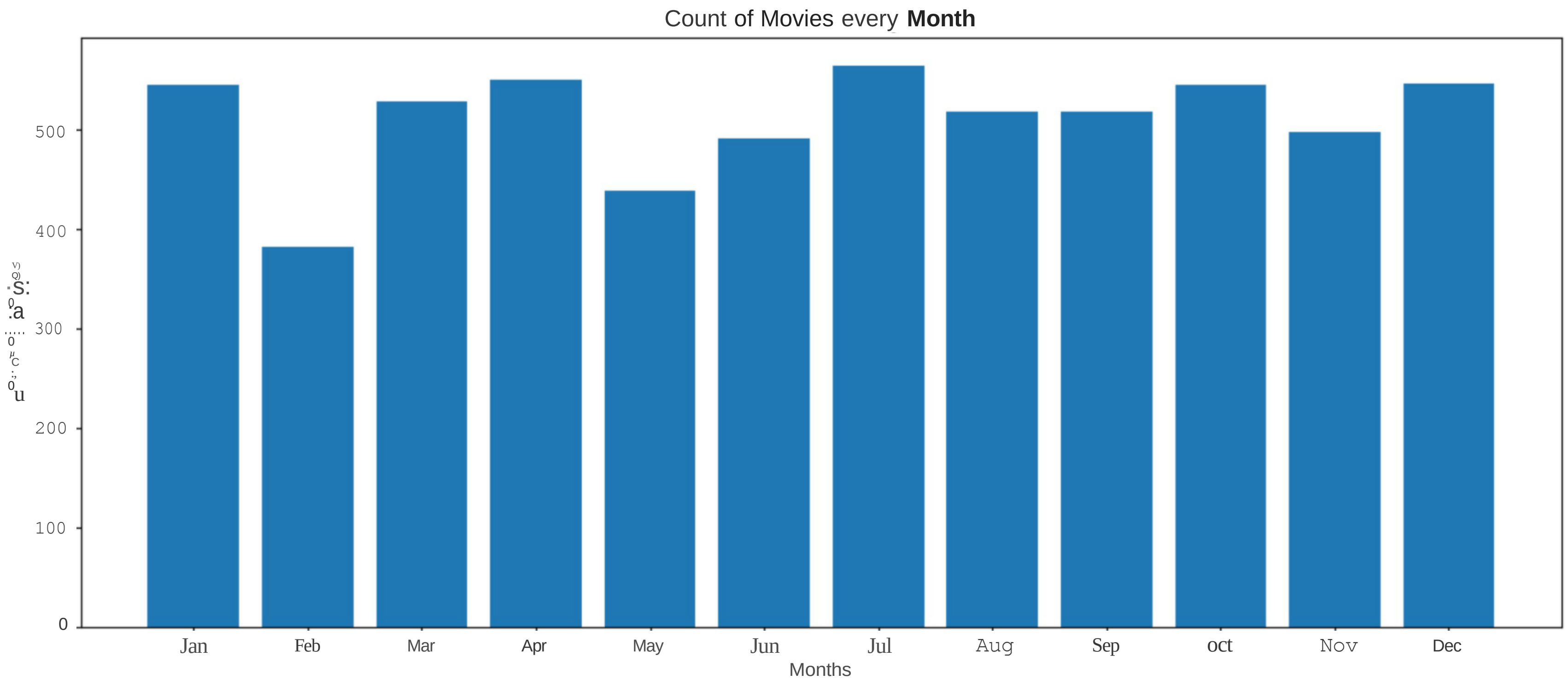
Out[43]:

	Month	TVShows Added
0	12.0	266
1	7.0	262
2	9.0	251
3	6.0	236
4	8.0	236
5	10.0	215
6	4.0	214

	Month	TVShows Added
7	3.0	213
8	11.0	207
9	5.0	193
1n	1 n	1Q?

In [44]:

```
 df movie= df_title_dateAdded.loc[df_title_dateAdded['type']=='Movie', :]
month_added = df_movie['month_added'].value_counts().reset_index().sort_values('index')
x = month_added['index']
y = month_added['month_added'].values

plt.bar(x, height= y)
plt.xticks([1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'O
plt.title('Count of Movies every Month')
plt.xlabel('Months')
plt.ylabel('Count of Movies')
plt.show()
```



In [45]:

```
d = df_movie['month_added'].value_counts().reset_index()
 d.columns = ['Month', 'Movies Added']
 d
```

Out[45]:

	Month	Movies Added
0	7.0	565
1	4.0	550
2	12.0	547
3	1.0	546
4	10.0	545
5	3.0	529
6	9.0	519

	Month	Movies Added
7	8.0	519
8	11.0	498
9	6.0	492
1n	c; n	

Insights -

- In Case of TV Shows, most of them are added in the month of December, July and September
- In Case of Movies, most of them are added in the month of July, April and December
- It can be observed that most of them are added in the midst of Summer and Winter session

Business Recommendation

- Its recommended to add shows when people have lesser options to watch, like in the month of February the number of Movies and shows added are less
- Other good time to add shows is when people have more lessure time to watch. In countries like USA, people get holidays during December. So for USA, Decemeber can be a good option to add Shows to Netflix

Question - Top Actors of TV Shows

In [47]:

```
df_tvshow = df.loc[df['type']=='TV Show', :]
df_tvshow_cast = df_tvshow[['cast', 'title']].drop_duplicates()
##Hereby using iloc, I made sure that Unknown dosent appear
d = df_tvshow_cast.groupby('cast').count().reset_index().sort_values(by = 'title', ascending= False).iloc[l:].head(10
d.columns = ['cast', 'count']
d
```

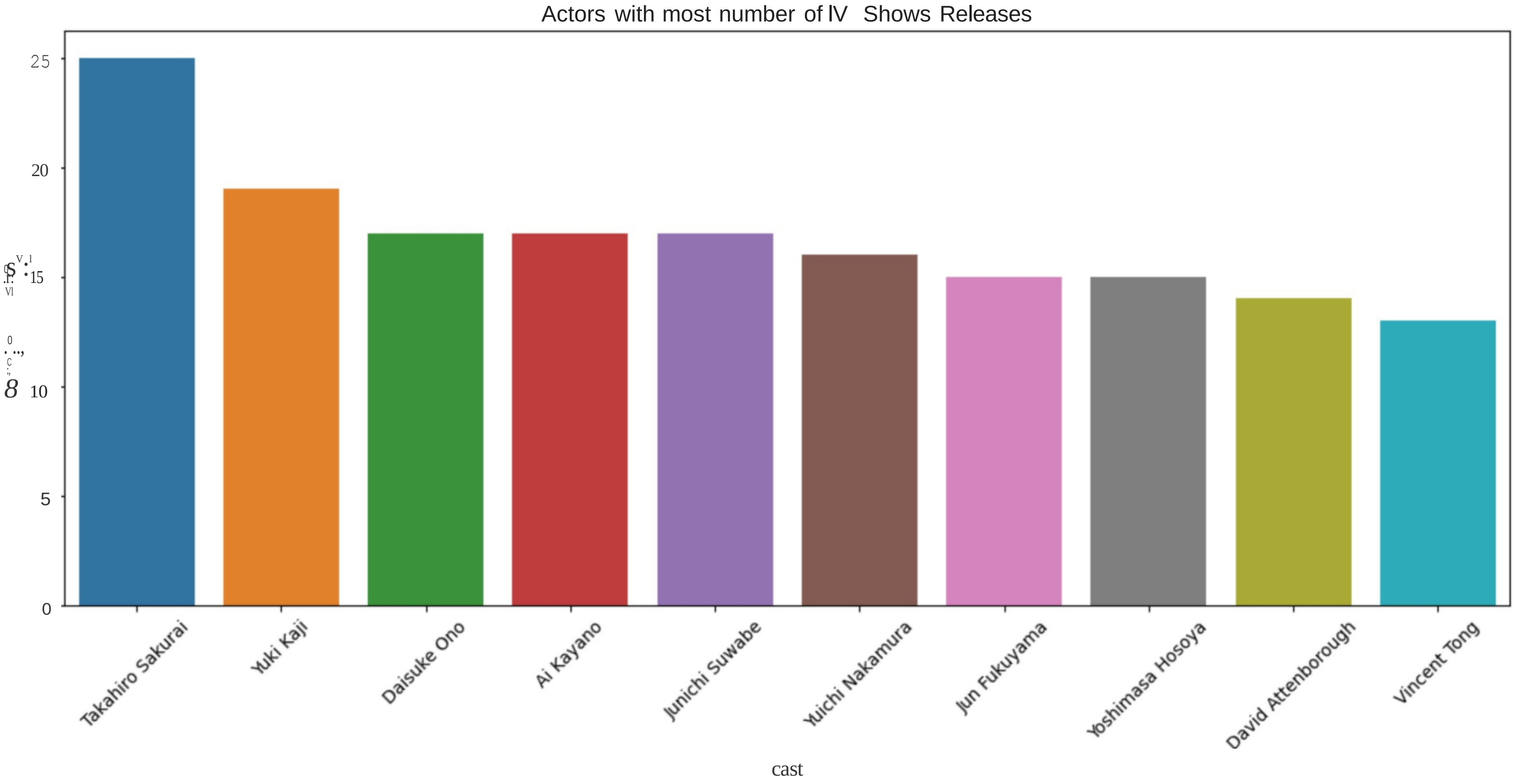
Out[47]:

	cast	count
13230	Takahiro Sakurai	25
14581	Yuki Kaji	19
2873	Daisuke Ono	17
251	Ai Kayano	17

	cast	count
6804	Junichi Suwabe	17
14565	Yuichi Nakamura	16
6761	Jun Fukuyama	15
14497	Yoshimasa Hosoya	15

In [48]:

```
sns.barplot(data = d, x = d['cast'], y = d['count'])
plt.xticks(rotation = 45)
plt.title('Actors with most number of TV Shows Releases')
plt.ylabel('Count of TV Shows')
plt.show()
```



Insights

- Here Cast Data is missing for many Records.
- Among the existing data Takahiro Sakurai, Yuki Kaji and Daisuke Ono are the top Cast who acted in most number of TV Shows.

Question - Top Actors of Movies

In [49]:

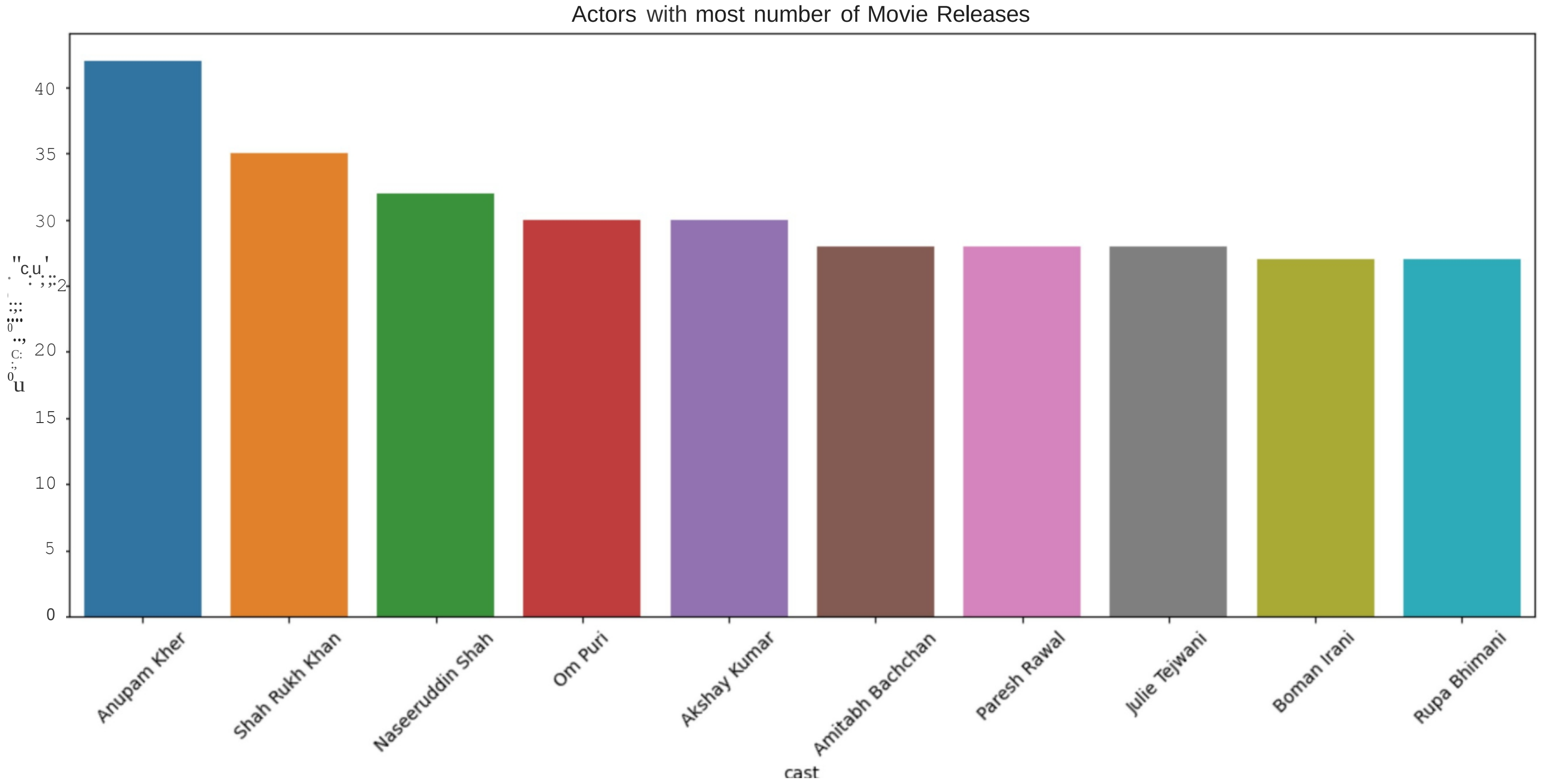
```
df_movies = df.loc[df['type']=='Movie', :]
df_movies_cast = df_movies[['cast', 'title']].drop_duplicates()
##Hereby using iloc, I made sure that Unknown dosent appear
d = df_movies_cast.groupby('cast').count().reset_index().sort_values(by = 'title', ascending= False).iloc[1:].head(10
d.columns = ['cast', 'count']
d
```

Out [49]:

	cast	count
2104	Anupam Kher	42
21781	Shah Rukh Khan	35
17193	Naseeruddin Shah	32
18064	Om Puri	30
637	Akshay Kumar	30
1312	Amitabh Bachchan	28
18329	Paresh Rawal	28
12031	Julie Tejwani	28
3353	Boman Irani	27
20692	Rupa Bhimani	27

In [51]:

```
sns.barplot(data = d, x = d['cast'], y = d[ 'count' ])
plt.xticks(rotation = 45)
plt.title('Actors with most number of Movie Releases')
plt.ylabel('Count of Movies')
plt. show()
```



Insights

- Here Cast Data is missing for many Records.
- Among the existing data Anupam Kher, Shah Rukh Khan and Naseeruddin Shah are the top Cast who acted in most number of Movies.

Question - Top Directors of TV Shows

In [52]:

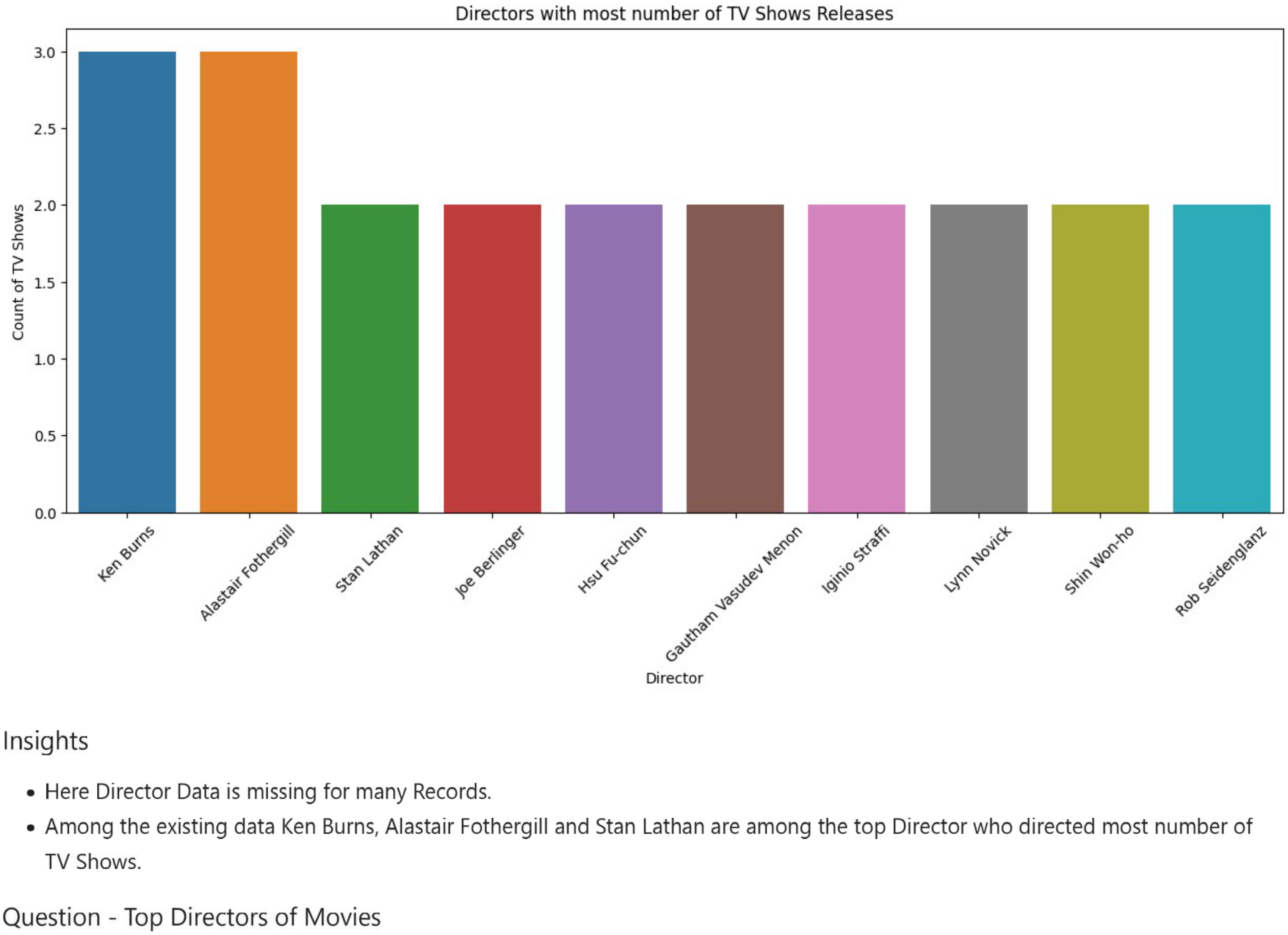
```
df_tvshow = df.loc[df['type']=='TV Show', :]
df_tvshow_director = df_tvshow[['director', 'title']].drop_duplicates()
d = df_tvshow_director.groupby('director').count().reset_index().sort_values(by = 'title', ascending= False).iloc[1:]
d.columns = ['Director', 'Count']
d
```

Out[52]:

	Director	Count
146	Ken Burns	3
8	Alastair Fothergill	3
259	Stan Lathan	2
128	Joe Berlinger	2
100	Hsu Fu-chun	2
84	Gautham Vasudev Menon	2
103	Iginio Straffi	2
168	Lynn Novick	2
251	Shin Won-ho	2
235	Rob Seidenglanz	2

In [54]:

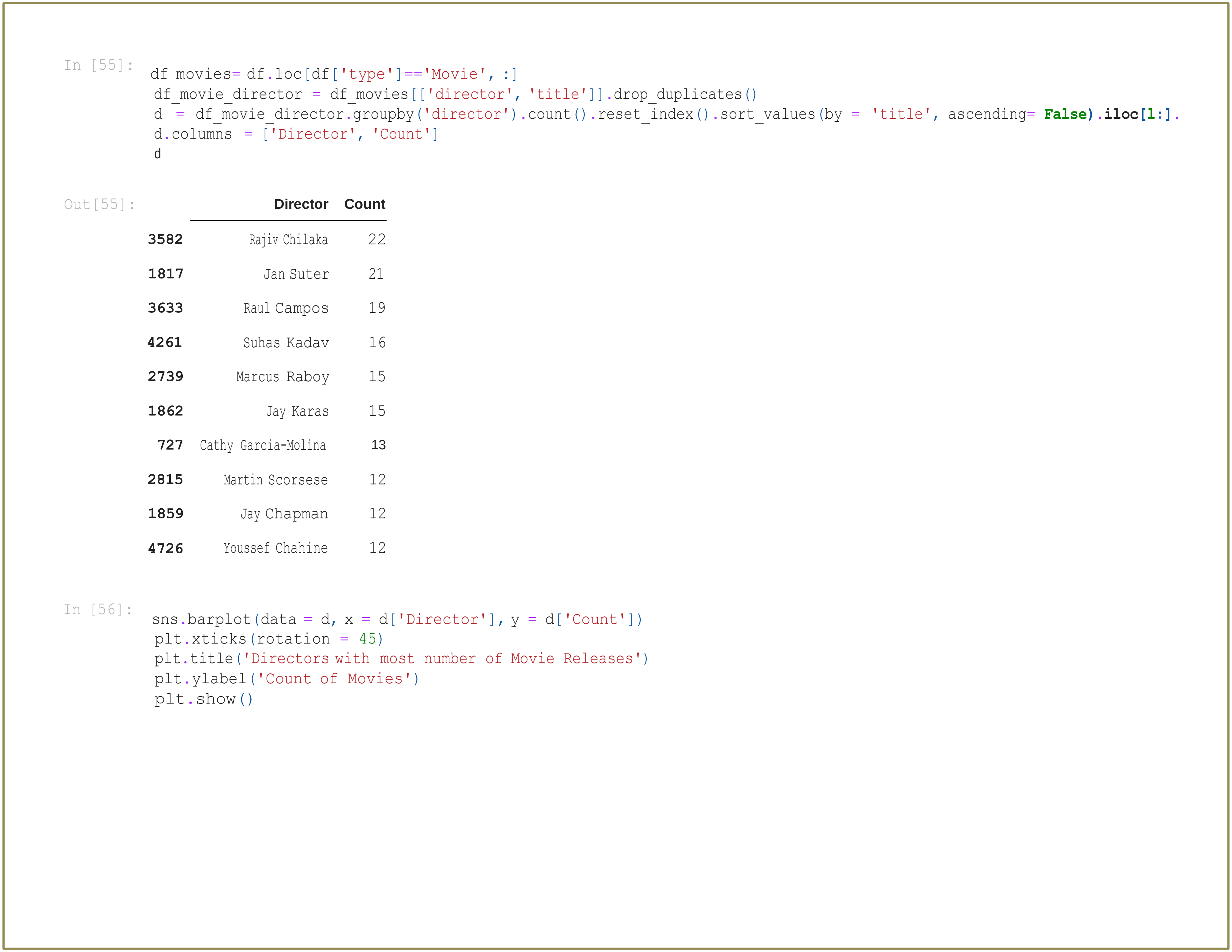
```
sns.barplot(data = d, x = d['Director'], y = d['Count'])
 plt.xticks(rotation = 45)
 plt.title('Directors with most number of TV Shows Releases')
 plt.ylabel('Count of TV Shows')
 plt.show()
```



Insights

- Here Director Data is missing for many Records.
- Among the existing data Ken Burns, Alastair Fothergill and Stan Lathan are among the top Director who directed most number of TV Shows.

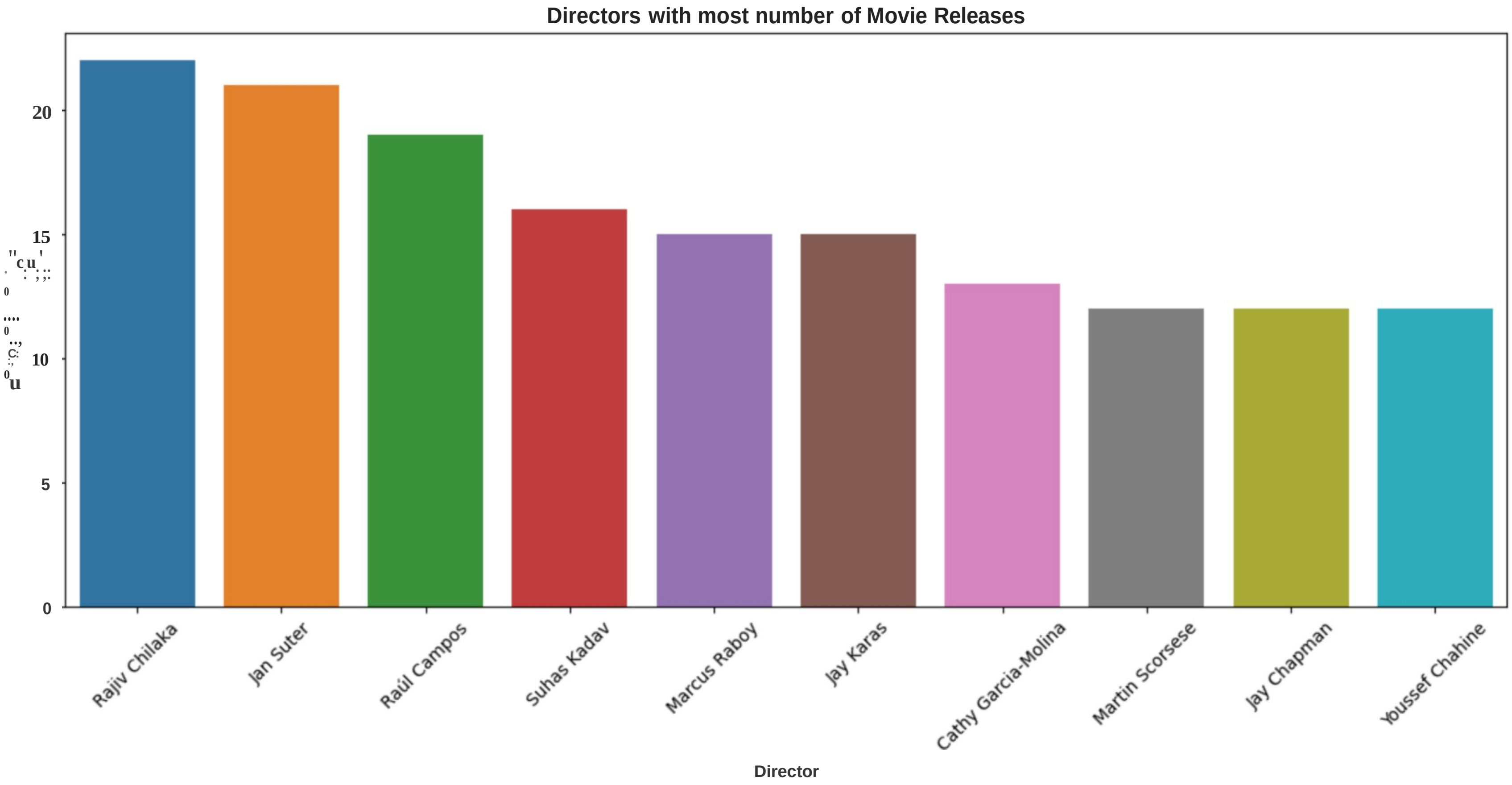
Question - Top Directors of Movies



```
In [55]: df movies= df.loc[df['type']=='Movie', :]
         df_movie_director = df_movies[['director', 'title']].drop_duplicates()
         d  = df_movie_director.groupby('director').count().reset_index().sort_values(by =  'title', ascending= False).iloc[1:].
         d.columns  = ['Director', 'Count']
         d
```

```
Out[55]:
                  Director  Count
         3582   Rajiv Chilaka    22
         1817       Jan Suter    21
         3633     Raul Campos    19
         4261     Suhas Kadav    16
         2739    Marcus Raboy    15
         1862       Jay Karas    15
          727 Cathy Garcia-Molina  13
         2815 Martin Scorsese    12
         1859     Jay Chapman    12
         4726 Youssef Chahine    12
```

```
In [56]: sns.barplot(data = d, x = d['Director'], y = d['Count'])
         plt.xticks(rotation = 45)
         plt.title('Directors with most number of Movie Releases')
         plt.ylabel('Count of Movies')
         plt.show()
```



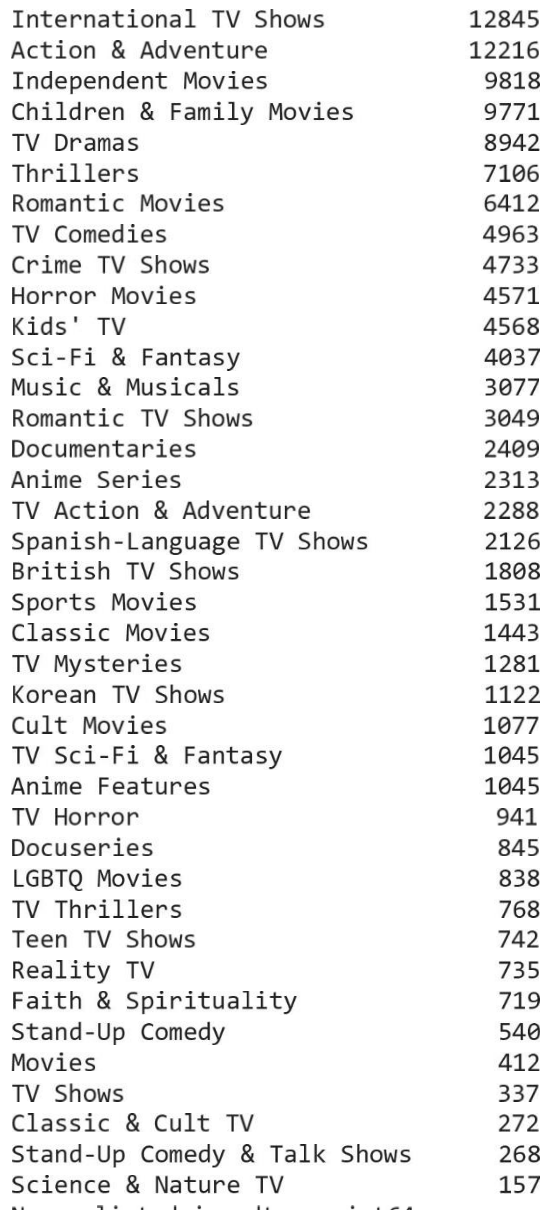
Insights

- Here Director Data is missing for many Records.
- Among the existing data Rajiv Chilaka, Jan Suter and Raul Campos are among the top Directors who directed most number of Movies.

Question - What are all the Genres present on the Netflix Platform ?

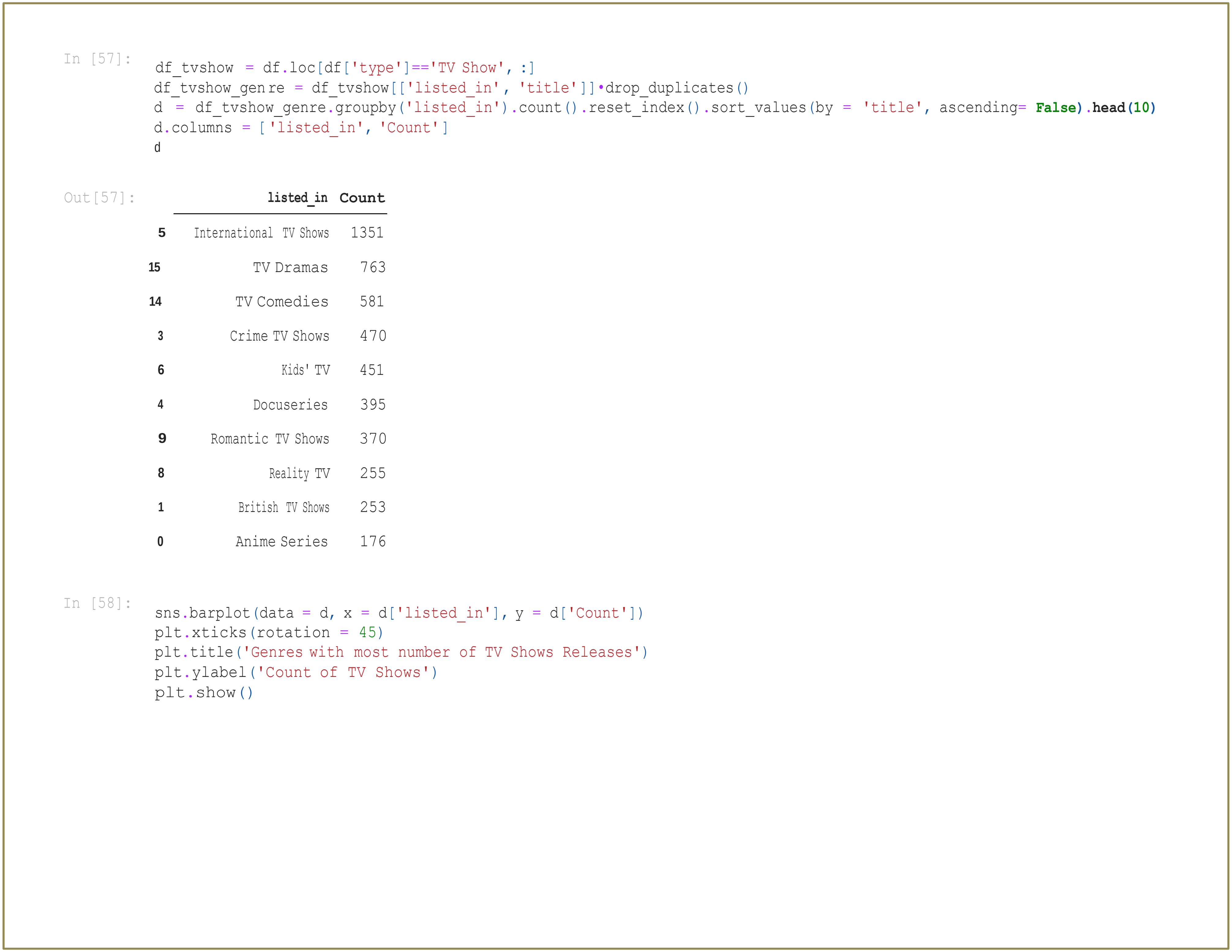
```
In [92]:  df['listed_in'].value_counts()
```

```
Out[92]: Dramas                           29787
         International Movies             28224
         Comedies                         20829
```



```
International TV Shows          12845
Action & Adventure              12216
Independent Movies               9818
Children & Family Movies         9771
TV Dramas                        8942
Thrillers                        7106
Romantic Movies                  6412
TV Comedies                      4963
Crime TV Shows                   4733
Horror Movies                    4571
Kids' TV                         4568
Sci-Fi & Fantasy                 4037
Music & Musicals                 3077
Romantic TV Shows                3049
Documentaries                    2409
Anime Series                     2313
TV Action & Adventure            2288
Spanish-Language TV Shows        2126
British TV Shows                 1808
Sports Movies                    1531
Classic Movies                   1443
TV Mysteries                     1281
Korean TV Shows                  1122
Cult Movies                      1077
TV Sci-Fi & Fantasy              1045
Anime Features                   1045
TV Horror                         941
Docuseries                        845
LGBTQ Movies                      838
TV Thrillers                      768
Teen TV Shows                     742
Reality TV                        735
Faith & Spirituality              719
Stand-Up Comedy                   540
Movies                            412
TV Shows                          337
Classic & Cult TV                 272
Stand-Up Comedy & Talk Shows      268
Science & Nature TV               157
Name: listed_in, dtype: int64
```

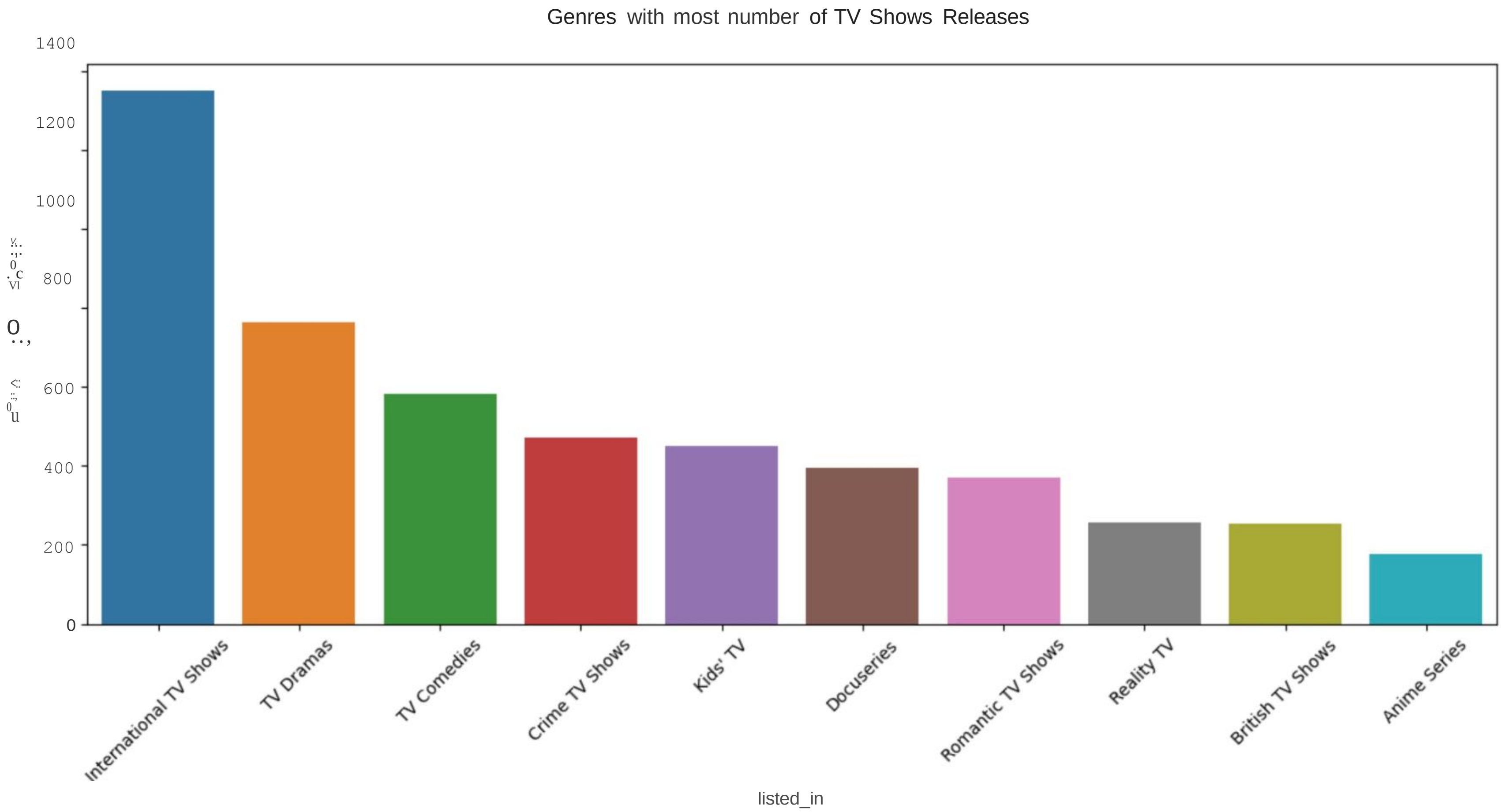
Question - Top Genre of TV Shows



```
In [57]:  df_tvshow = df.loc[df['type']=='TV Show', :]
          df_tvshow_gen re = df_tvshow[['listed_in', 'title']]•drop_duplicates()
          d  = df_tvshow_genre.groupby('listed_in').count().reset_index().sort_values(by = 'title', ascending= False).head(10)
          d.columns = ['listed_in', 'Count']
          d
```

```
Out[57]:
             listed_in  Count
 5  International TV Shows  1351
15           TV Dramas   763
14         TV Comedies   581
 3       Crime TV Shows   470
 6             Kids' TV   451
 4          Docuseries   395
 9     Romantic TV Shows   370
 8           Reality TV   255
 1       British TV Shows   253
 0        Anime Series   176
```

```
In [58]:  sns.barplot(data = d, x = d['listed_in'], y = d['Count'])
          plt.xticks(rotation = 45)
          plt.title('Genres with most number of TV Shows Releases')
          plt.ylabel('Count of TV Shows')
          plt.show()
```



Question - Top Genre of Movie

In [59]:

```
df_movie = df.loc[df['type']=='Movie', :]
df_movie_genre = df_movie[['listed_in', 'title']].drop_duplicates()
d  = df_movie_genre.groupby('listed_in').count().reset_index().sort_values(by = 'title', ascending= False).head(10)
d.columns= ['listed_in', 'Count']
d
```

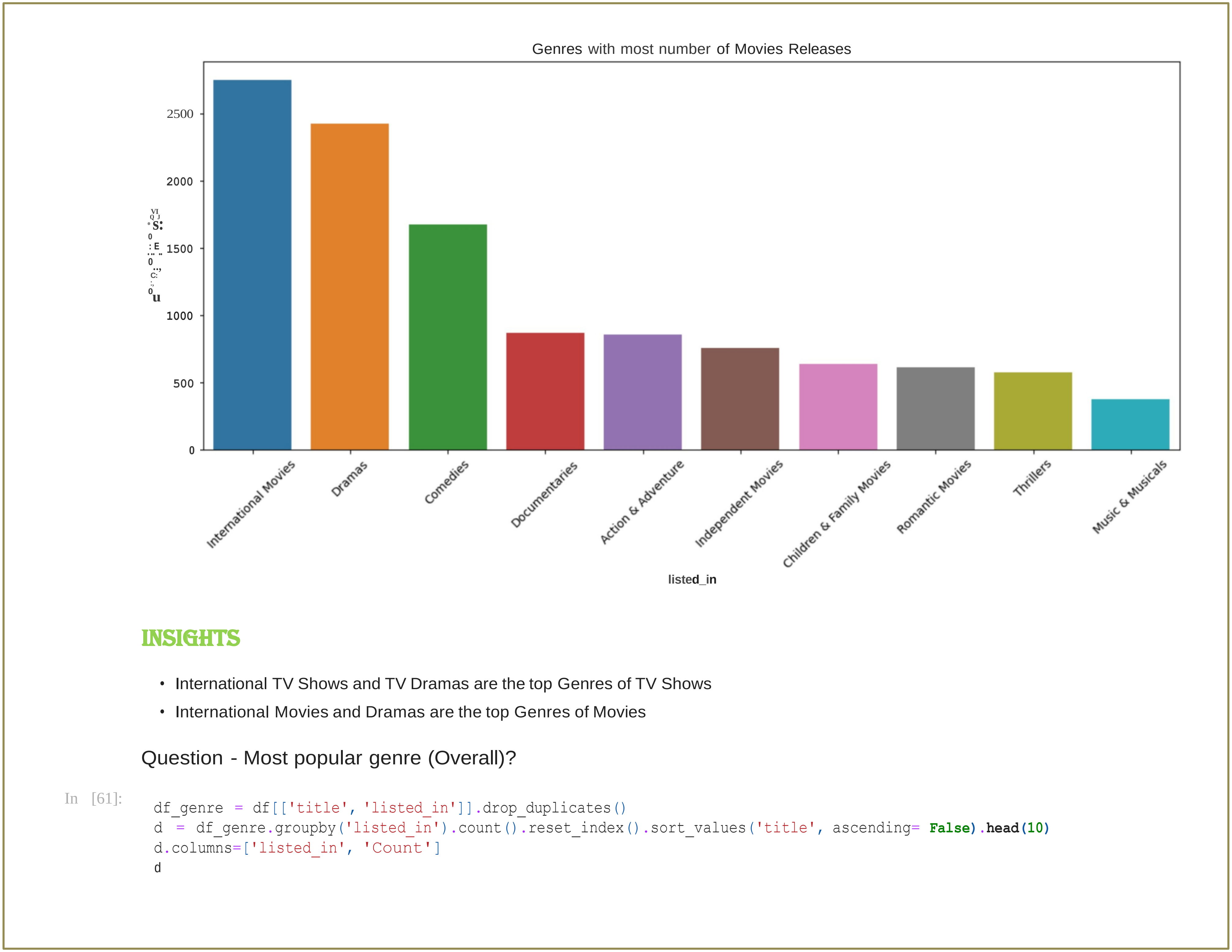
Out[59]:

	listed_in	Count
11	International Movies	2752
7	Dramas	2427

	listed_in	Count
4	Comedies	1674
6	Documentaries	869
0	Action & Adventure	859
10	Independent Movies	756
2	Children & Family Movies	641
15	Romantic Movies	616
10	Th,;11 ,r	r:.77

In [60]:

```
sns.barplot(data = d, x = d['listed_in'], y = d['Count'])
plt.xticks(rotation = 45)
plt.title('Genres with most number of Movies Releases')
plt.ylabel('Count of Movies')
plt.show()
```



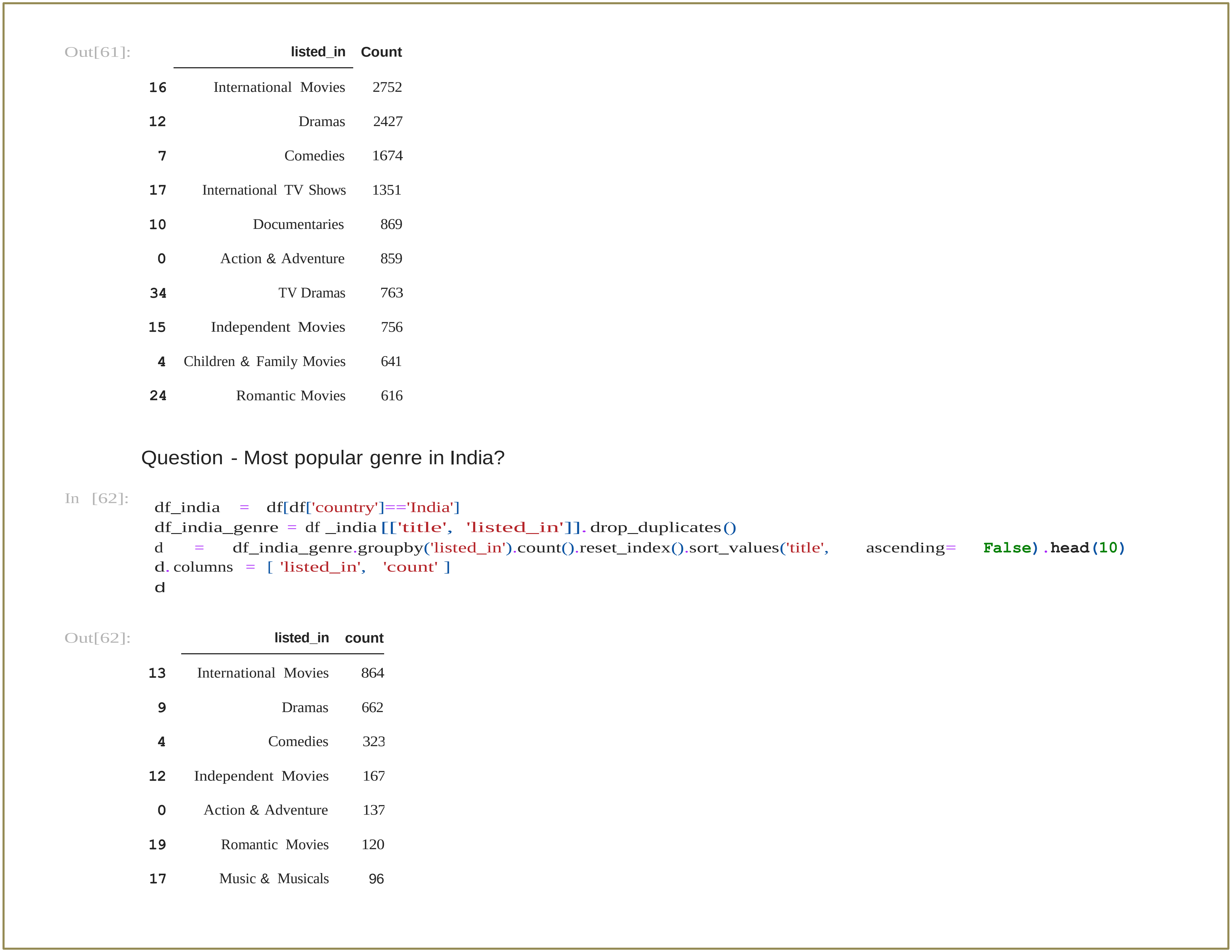
INSIGHTS

- International TV Shows and TV Dramas are the top Genres of TV Shows
- International Movies and Dramas are the top Genres of Movies

Question - Most popular genre (Overall)?

In [61]:

```
df_genre = df[['title', 'listed_in']].drop_duplicates()
d  = df_genre.groupby('listed_in').count().reset_index().sort_values('title', ascending= False).head(10)
d.columns=['listed_in', 'Count']
d
```



Out[61]:

	listed_in	Count
16	International Movies	2752
12	Dramas	2427
7	Comedies	1674
17	International TV Shows	1351
10	Documentaries	869
0	Action & Adventure	859
34	TV Dramas	763
15	Independent Movies	756
4	Children & Family Movies	641
24	Romantic Movies	616

Question - Most popular genre in India?

In [62]:

```
df_india = df[df['country']=='India']
df_india_genre = df _india [['title', 'listed_in']]. drop_duplicates()
d = df_india_genre.groupby('listed_in').count().reset_index().sort_values('title', ascending= False).head(10)
d. columns = [ 'listed_in', 'count' ]
d
```

Out[62]:

	listed_in	count
13	International Movies	864
9	Dramas	662
4	Comedies	323
12	Independent Movies	167
0	Action & Adventure	137
19	Romantic Movies	120
17	Music & Musicals	96

	listed_in	count
34	Thrillers	92
14	International TV Shows	66

Insights -

- Most popular genres across the whole Netflix Platform - 'International Movies', 'Dramas', 'Comedies', 'International TV Shows', 'Documentaries', 'Action & Adventure'

Business Insights -

Most of the Movies and Shows are from International Movies, International TV Shows and Dramas Genres. We can infer that Netflix has good number of viewers who watch International shows, and they are not very specific to their Regional Shows.

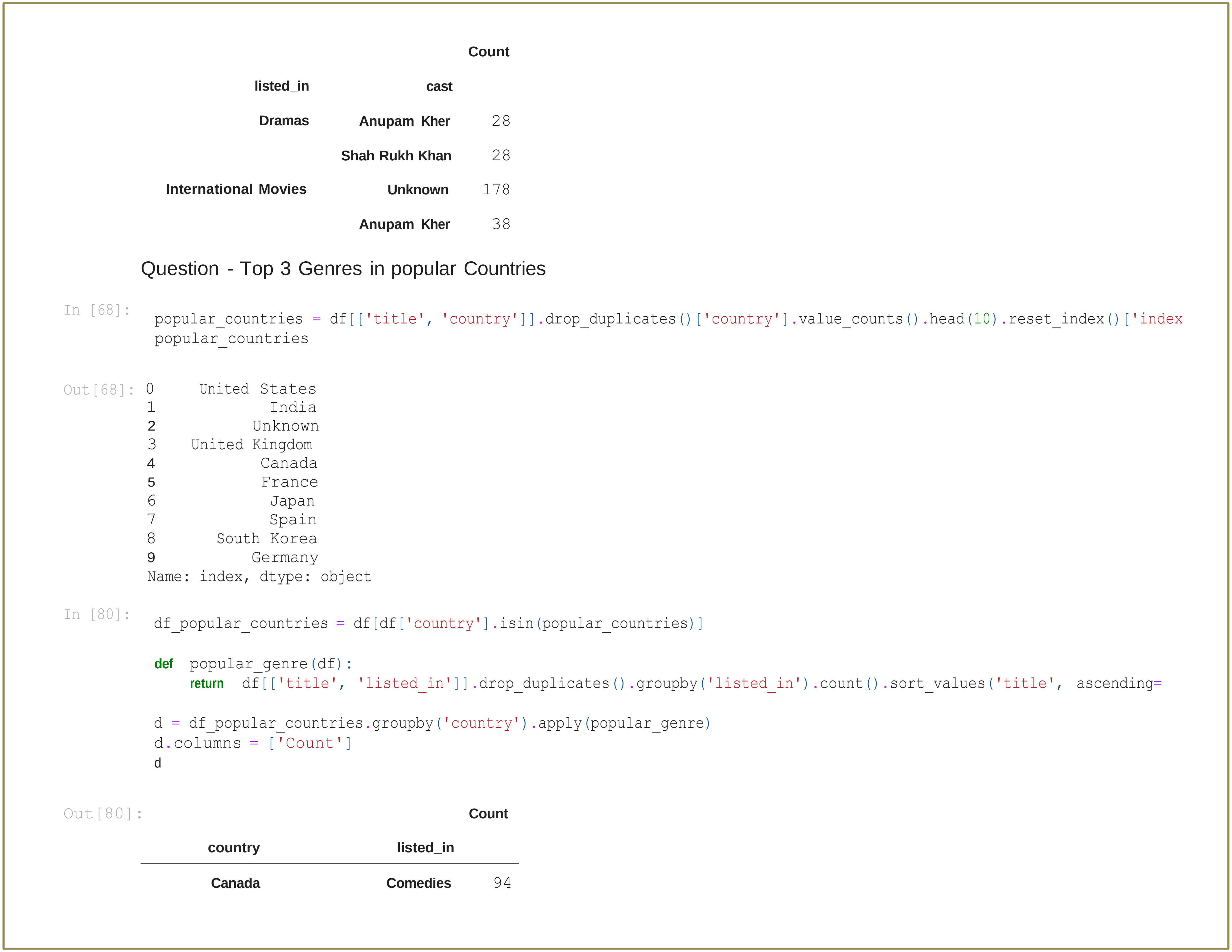
Question - Top 2 Actors who worked the most in Popular Genre

In [78]:

```
df_genre = df[df['listed_in'].isin(['International Movies', 'Dramas', 'Comedies', 'International TV Shows', 'Documenta

def popular_cast(df):
    return df[['title', 'cast']].drop_duplicates().groupby('cast').count().sort_values('title', ascending=False).hea

d = df_genre.groupby('listed_in').apply(popular_cast)

d.columns = ['Count']
d
```

Out[78]:

		Count
listed_in	**cast**	
Action & Adventure	**Bruce Willis**	13
	Amitabh Bachchan	12
Comedies	**Anupam Kher**	20
	Paresh Rawal	18
Documentaries	**Unknown**	424
	Samuel West	10



		Count
listed_in	cast	
Dramas	Anupam Kher	28
	Shah Rukh Khan	28
International Movies	Unknown	178
	Anupam Kher	38

Question - Top 3 Genres in popular Countries

In [68]:

```
popular_countries = df[['title', 'country']].drop_duplicates()['country'].value_counts().head(10).reset_index()['index
popular_countries
```

Out[68]:

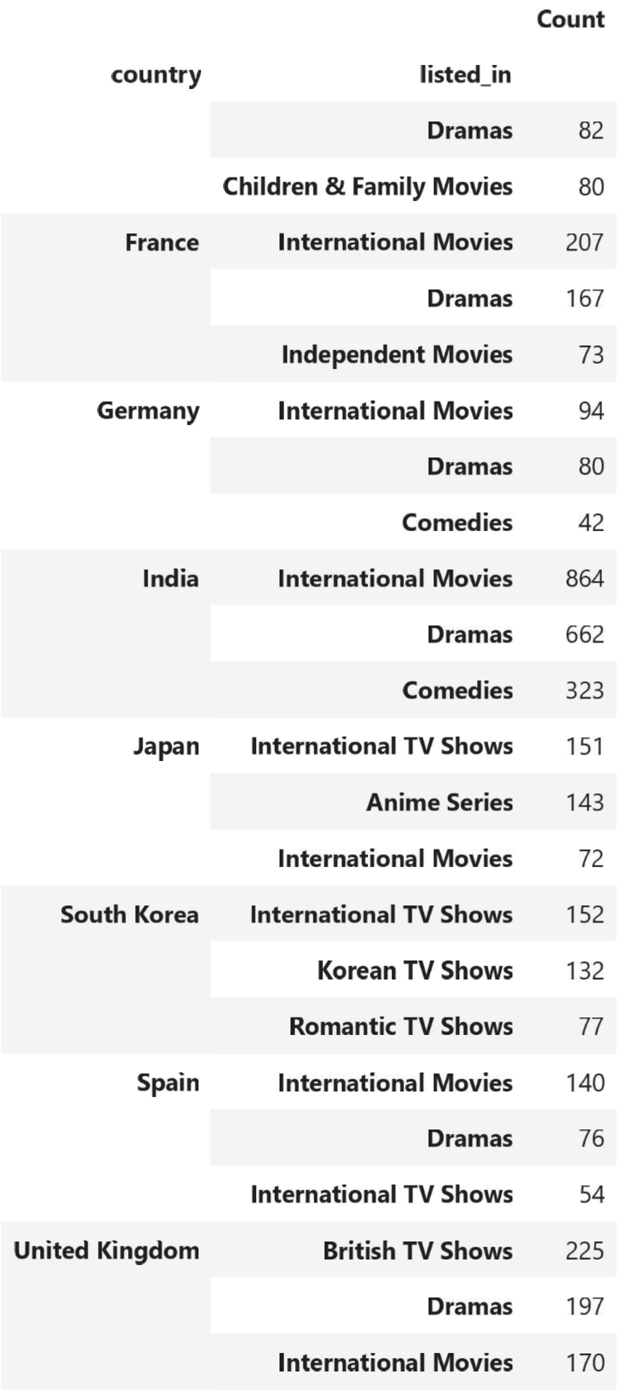
```
0     United States
1             India
2           Unknown
3    United Kingdom
4            Canada
5            France
6             Japan
7             Spain
8       South Korea
9           Germany
Name: index, dtype: object
```

In [80]:

```
df_popular_countries = df[df['country'].isin(popular_countries)]

def popular_genre(df):
    return df[['title', 'listed_in']].drop_duplicates().groupby('listed_in').count().sort_values('title', ascending=

d = df_popular_countries.groupby('country').apply(popular_genre)
d.columns = ['Count']
d
```

Out[80]:

		Count
country	listed_in	
Canada	Comedies	94



country	listed_in	Count
	Dramas	82
	Children & Family Movies	80
France	International Movies	207
	Dramas	167
	Independent Movies	73
Germany	International Movies	94
	Dramas	80
	Comedies	42
India	International Movies	864
	Dramas	662
	Comedies	323
Japan	International TV Shows	151
	Anime Series	143
	International Movies	72
South Korea	International TV Shows	152
	Korean TV Shows	132
	Romantic TV Shows	77
Spain	International Movies	140
	Dramas	76
	International TV Shows	54
United Kingdom	British TV Shows	225
	Dramas	197
	International Movies	170

		Count
country	**listed_in**	
United States	**Dramas**	835

Insights -

- International Movies and Dramas are top 2 in most of the countries
- Comedies is also one of the Popular Genres.
- In countries like Japan - Anime Series and in South Korea - Korean TV Shows are popular genres.

Question - What are top genres in different years?

In [85]:

```
df_recent_years = df[df['release_year'].isin([2021, 2020, 2019, 2018, 2017, 2016, 2015, 2014, 2013, 2012, 2011, 2010])

def popular_rated(df):
    return df[['title', 'listed_in']].drop_duplicates().groupby('listed_in').count().reset_index().sort_values('title'

d = df_recent_years.groupby('release_year').apply(popular_rated).sort_values('release_year', ascending= False)#.drop(

d.columns = ['listed_in', 'count']
d
```

Out[85]:

		listed_in	count
release_year			
2021	**13**	International Movies	141
	14	International TV Shows	149
2020	**15**	International TV Shows	214
	14	International Movies	239
2019	**10**	Dramas	243
	14	International Movies	282
2018	**12**	Dramas	304
	16	International Movies	340

release_year		listed_in	count
2017	**11**	Dramas	285
	15	International Movies	328
2016	**11**	Dramas	265
	15	International Movies	305
2015	**10**	Dramas	180
	14	International Movies	210
2014	**9**	Dramas	104
	13	International Movies	127
2013	**11**	Dramas	83
	15	International Movies	121
2012	**10**	Dramas	66
	14	International Movies	80
2011	**14**	International Movies	55
	10	Dramas	60

Insights

- International Movies and Dramas are the most Popular genre until 2019
- From 2020 International TV Shows became one of the popular genre

Question - Most popular actor-director pair for movies across India?

In [86]:

```
df_india = df[df['country']=='India']
 df_cast_director  = df_india[['title','cast', 'director']].drop_duplicates()
 df_cast_director.groupby(['cast',  'director']).count().reset_index().sort_values('title',  ascending=  False).head(10)
```

Out[86]:

cast	director	title

	cast	director	title
7531	Unknown	Unknown	18
817	Anupam Kher	David Dhawan	6
5912	Salman Khan	Sooraj R. Barjatya	5
402	Alok Nath	Sooraj R. Barjatya	5
2811	Julie Tejwani	Rajiv Chilaka	4
5327	Rajpal Yadav	Priyadarshan	4
259	Ajay Devgn	Prakash Jha	4
3851	Mithun Chakraborty	Umesh Mehra	4

The most popular pair is - Anupam Kher and David Dhawan

Question - Most of the movies are Rated as ?

In [87]:

```
df_rating = df[['rating', 'title']].drop_duplicates()
df_rating.groupby('rating').count().sort_values('title', ascending=False).head(10)
```

Out[87]:

rating	title
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

title

Question - What are most of the movies rated as in top countries ?

In [88]:

```
df_popular_countries = df[df['country'].isin(popular_countries)]

def  popular_rated(df):
    return df[['title', 'rating']].drop_duplicates().groupby('rating').count().reset_index().sort_values('title', asce

df_popular_countries.groupby('country').apply(popular_rated)
```

Out[88]:

		rating	title
country			
Canada	**8**	TV-MA	107
	5	R	79
	6	TV-14	49
France	**8**	TV-MA	163
	5	R	57
	6	TV-14	48
Germany	**7**	TV-MA	79
	4	R	43
	3	PG-13	31
India	**4**	TV-14	572
	6	TV-MA	266
	7	TV-PG	144
Japan	**6**	TV-MA	101
	4	TV-14	99
	7	TV-PG	50
South Korea	**7**	TV-MA	92

		rating	title
country			
	5	TV-14	86
	8	TV-PG	19
Spain	**8**	TV-MA	170
	6	TV-14	18
	5	R	13
United Kingdom	**7**	TV-MA	253
	4	R	145
	5	TV-14	103
United States	**11**	TV-MA	1101
	8	R	660
	9	TV-14	497
Unknown	**5**	TV-MA	281

INSIGHTS

- If we consider on a whole, most of the Movies are rated as TV MA. This is for meant for Matured Audience (17 + age group).
- Other popular Category is TV-14 which ages under 14.

BUSINESS INSIGHTS

- We have most the shows for age group 17+ and under 14
- We can consider these two groups as main Target audience and make more relevant content.