Importing the necessary libraries

In [2]:

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

```
D:\Anaconda\lib\site-packages\numpy\_distributor_init.py:32: UserWarning: loade
d more than 1 DLL from .libs:
D:\Anaconda\lib\site-packages\numpy\.libs\libopenblas.PYQHXLVVQ7VESDPUVUADXEVJO
BGHJPAY.gfortran-win_amd64.dll
D:\Anaconda\lib\site-packages\numpy\.libs\libopenblas.WCDJNK7YVMPZQ2ME2ZZHJJRJ3
JIKNDB7.gfortran-win_amd64.dll
  stacklevel=1)
```

Loading the data set

In [3]:

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

In [4]:

```
df.head(1)
```

Out[4]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	D

Data Understanding and Prepration

In [5]:

```
#To get the idea about data type and count
df.info()
#To get the frequency and other fetaures of the data
df.describe(include='object').T
```

```
<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
```

Out[5]:

	count	unique	top	freq
show_id	8807	8807	s7403	1
type	8807	2	Movie	6131
title	8807	8807	The Lives of Others	1
director	6173	4528	Rajiv Chilaka	19
cast	7982	7692	David Attenborough	19
country	7976	748	United States	2818
date_added	8797	1767	January 1, 2020	109
rating	8803	17	TV-MA	3207
duration	8804	220	1 Season	1793
listed_in	8807	514	Dramas, International Movies	362
description	8807	8775	Paranormal activity at a lush, abandoned prope...	4

In [6]:

```
(df.isnull().sum()/df.shape[0])*100
```

Out[6]:

```
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
```

Handling the null values

To get to know weather missing values are there and if are how much percentage of each data is missing

Here director column has almost 30% data missing so data imputation is needed we will fillna values with mode of the director column

For cast and country missing column values we will replace it with mode of the column

For rating and duration we can drop the rows as % is too low

For date_added we can replace it with the release_year

In [7]:

```
#For rating and duration we can drop the rows as % is too low
df.dropna(subset=['rating','duration'],inplace=True)
```

In [8]:

```
#fill the director and country having missing value based on the mode of the resp
df['director'].fillna(value=df['director'].mode()[0],inplace=True)
df['country'].fillna(value=df['country'].mode()[0],inplace=True)
```

In [9]:

```
#Handling missing value for cast to replace with the top rated cast
df1=pd.read_csv("netflix.csv")
df2=df1.assign(cast=df1['cast'].str.split(', ')).explode('cast')
top_cast=df2.groupby('cast')['title'].nunique().sort_values().last_valid_index()
df['cast'].fillna(value=top_cast,inplace=True)
```

In [10]:

```
#change release year to date time object
df['release_year'] = pd. to_datetime(df['release_year'],format='%Y')
df['release_year'] = df['release_year'].dt.year
#df.info()
#df.head(1)
```

In [11]:

```
#extract month and year added from date added column
df['date_added'] = pd.to_datetime(df['date_added'])
df['year_added'] = df['date_added'].dt.year
df['month_added']=df['date_added'].dt.month


#replace the missing year added by release year of corresponding row --->
#it will be better rather than doing withmode of the column
def func(x):
    if ((pd.notnull(x.year_added))):
        return x.year_added
    else:
        return x.release_year

df['year_added']=df.apply(func,axis=1)

#replace the month based on mode of the date_added month column
df['month_added'].fillna(value=df['month_added'].mode()[0],inplace=True)
df.drop(['date_added'],axis=1,inplace=True)
```

In [12]:

```
df['year_added'] = df['year_added'].astype('int64')
df['month_added']=df['month_added'].astype('int64')
```

In [13]:

```
df.isnull().any()
```

Out[13]:

```
show_id         False
type            False
title           False
director        False
cast            False
country         False
release_year    False
rating          False
duration        False
listed_in       False
description     False
year_added      False
month_added     False
dtype: bool
```

Spliting the dataset based on type

In [14]:
```
movies_df = df[df['type']=='Movie'].copy()
movies_df.head(1)
```

Out[14]:

	show_id	type	title	director	cast	country	release_year	rating	duration	lis
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Anupam Kher	United States	2020	PG-13	90 min	Docume

In [15]:
```
tv_shows_df = df[df['type']=='TV Show'].copy()
tv_shows_df.head(1)
```

Out[15]:

	show_id	type	title	director	cast	country	release_year	rating	duration	listed_in
1	s2	TV Show	Blood & Water	Rajiv Chilaka	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries

In [16]:
```
movies_df.duration = movies_df.duration.str.replace(' min','').astype(int)
tv_shows_df.rename(columns={'duration':'seasons'}, inplace=True)
tv_shows_df.replace({'seasons':{'1 Season':'1 Seasons'}}, inplace=True)
tv_shows_df.seasons = tv_shows_df.seasons.str.replace(' Seasons','').astype(int)
```

In [17]:
```
movies_df.head(1)
```

Out[17]:

	show_id	type	title	director	cast	country	release_year	rating	duration	listed_
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	Anupam Kher	United States	2020	PG-13	90	Documentari

In [18]:

```
tv_shows_df.head(1)
```

Out[18]:

	show_id	type	title	director	cast	country	release_year	rating	seasons	listed_in
1	s2	TV Show	Blood & Water	Rajiv Chilaka	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021	TV-MA	2	International TV Shows, TV Dramas, TV Mysteries

Exploratory data analysis and visualization

Here we will start with the analysis n complete dataset

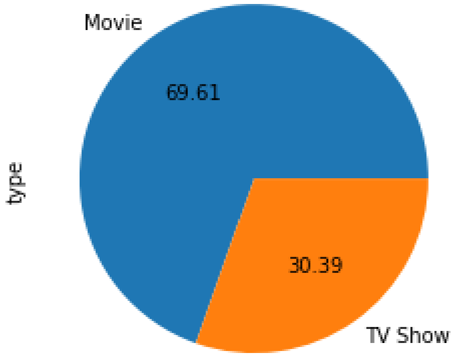
Ratio of TV Shows vs Movies on Netflix

In [19]:

```
#What is the ratio of movies vs TV shows
df['type'].value_counts().plot(kind='pie',autopct='%.2f')
#Based on given dataset about 70 % are movies
```

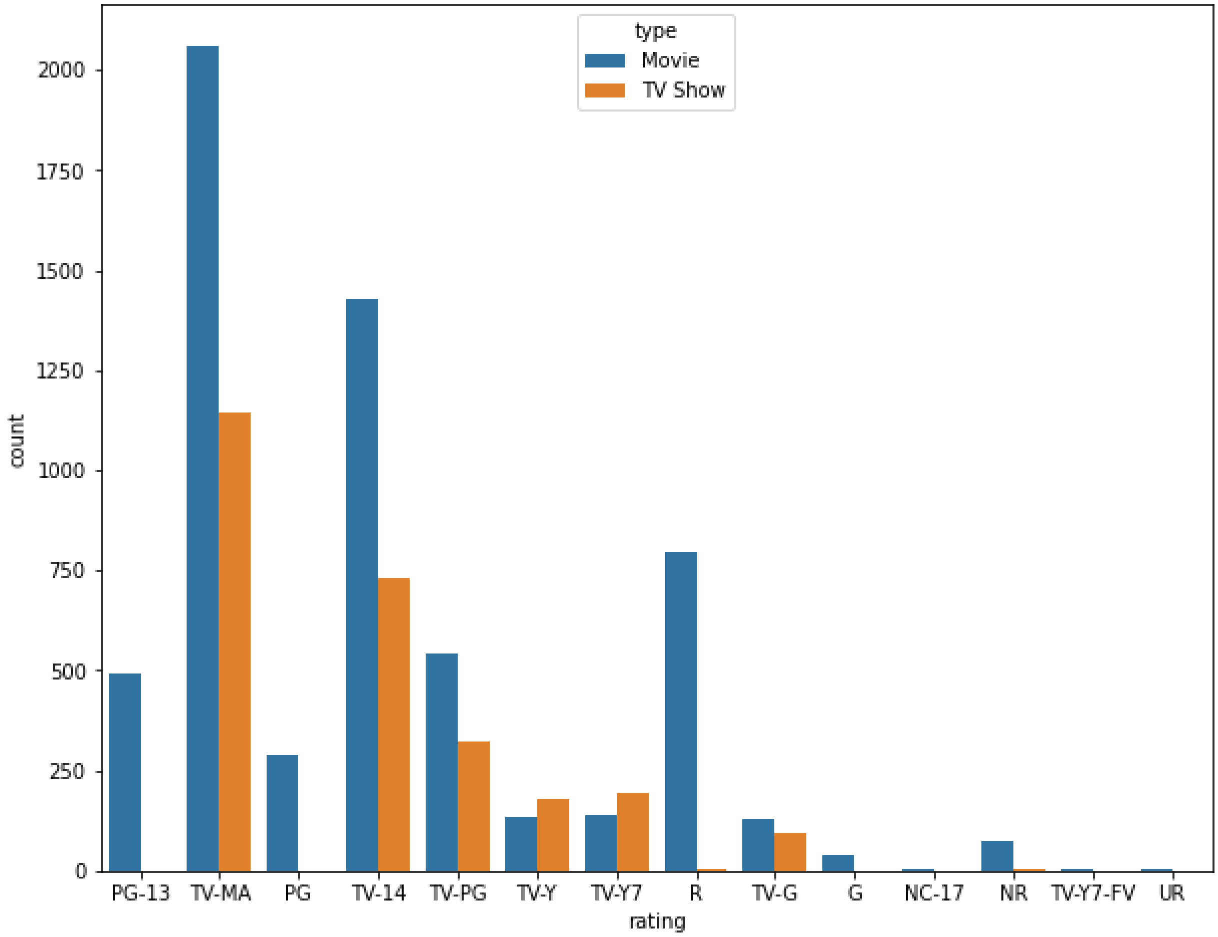
Out[19]:

```
<AxesSubplot:ylabel='type'>
```



```
In [20]: #countplot for the movies and tv shows
         plt.figure(figsize=(10,8))
         sns.countplot(data=df,x='rating',hue='type')
         #It can be infered that maximum number of movies /tv shows fall under TV-MA catog
```

```
Out[20]: <AxesSubplot:xlabel='rating', ylabel='count'>
```

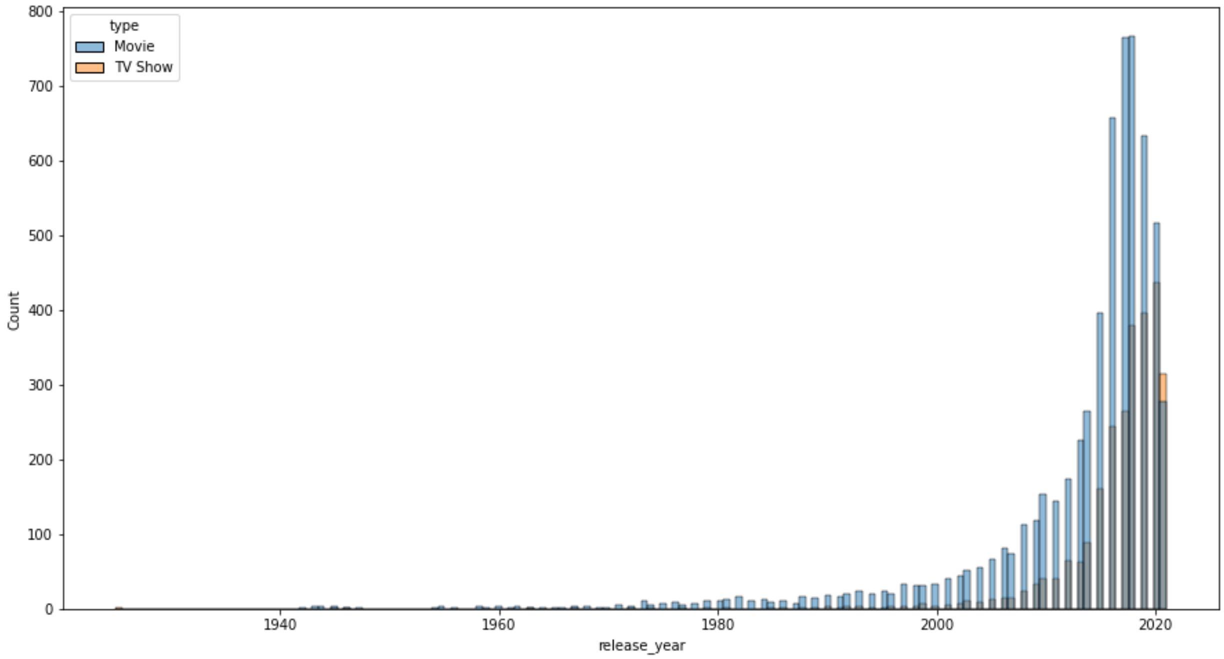


In [21]:

```
#histogram to get the trend over the years for shows/movies
plt.figure(figsize=(15,8))
sns.histplot(data=df,x='release_year',hue='type')
```

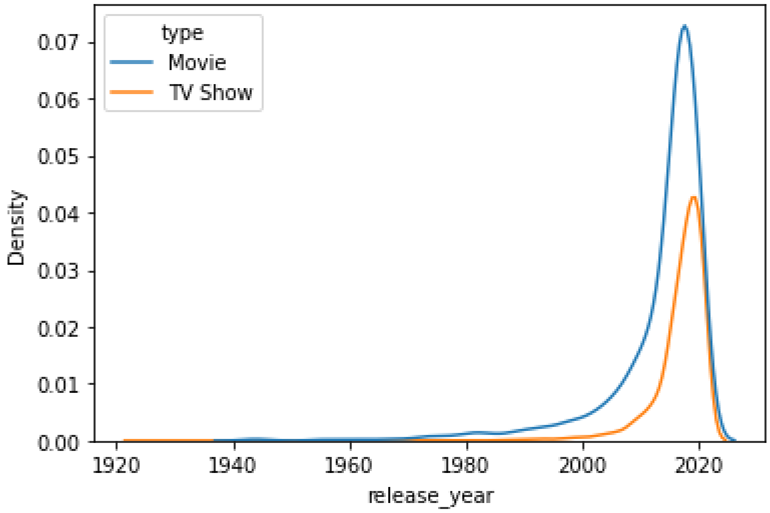
Out[21]:

```
<AxesSubplot:xlabel='release_year', ylabel='Count'>
```



```
In [22]: sns.kdeplot(data=df, x="release_year",hue='type')
```

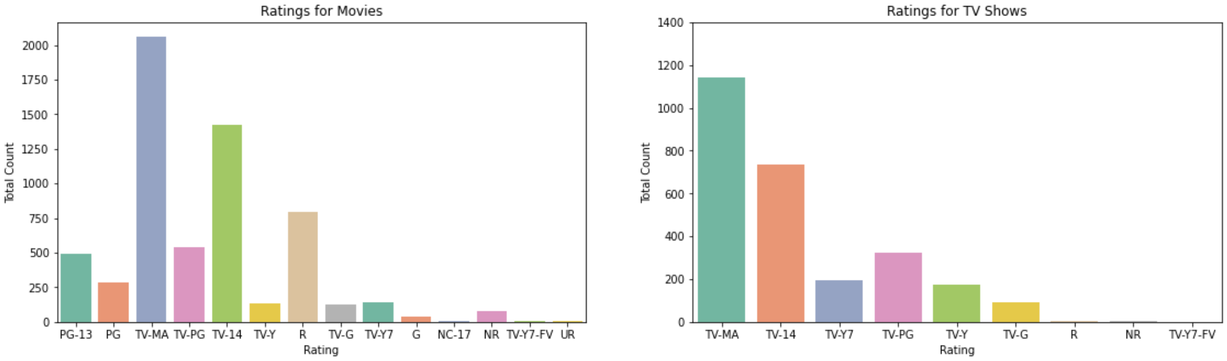
```
Out[22]: <AxesSubplot:xlabel='release_year', ylabel='Density'>
```



It can be infered that it has a increasing trend over the year specially around 2020 there is huge spike for the shows/movies

In [23]:
```
fig, ax = plt.subplots(1,2, figsize=(19, 5))
g1 = sns.countplot(data=movies_df,x='rating',palette="Set2", ax=ax[0]);
g1.set_title("Ratings for Movies")
g1.set_xlabel("Rating")
g1.set_ylabel("Total Count")
g2 = sns.countplot(data=tv_shows_df,x='rating',palette="Set2", ax=ax[1]);
g2.set(yticks=np.arange(0,1600,200))
g2.set_title("Ratings for TV Shows")
g2.set_xlabel("Rating")
g2.set_ylabel("Total Count")
fig.show()
```

```
D:\Anaconda\lib\site-packages\ipykernel_launcher.py:11: UserWarning: Matplotlib
is currently using module://ipykernel.pylab.backend_inline, which is a non-GUI
backend, so cannot show the figure.
  # This is added back by InteractiveShellApp.init_path()
```



Ratings for Movies

Ratings for TV Shows

From the above bar plot for movies vs tv shows it can be infered like TA-MA cateogry contents are more for movies vs tv shows while TV-14 cateogry contents are more for TV shows compared to TV-MA cateogry

Analysis of tv shows and movies over netflix over the year

In [24]:

```
netflix_year = df['year_added'].value_counts().to_frame().reset_index().rename(co
netflix_year = netflix_year[netflix_year.year != 2022]
netflix_year
```

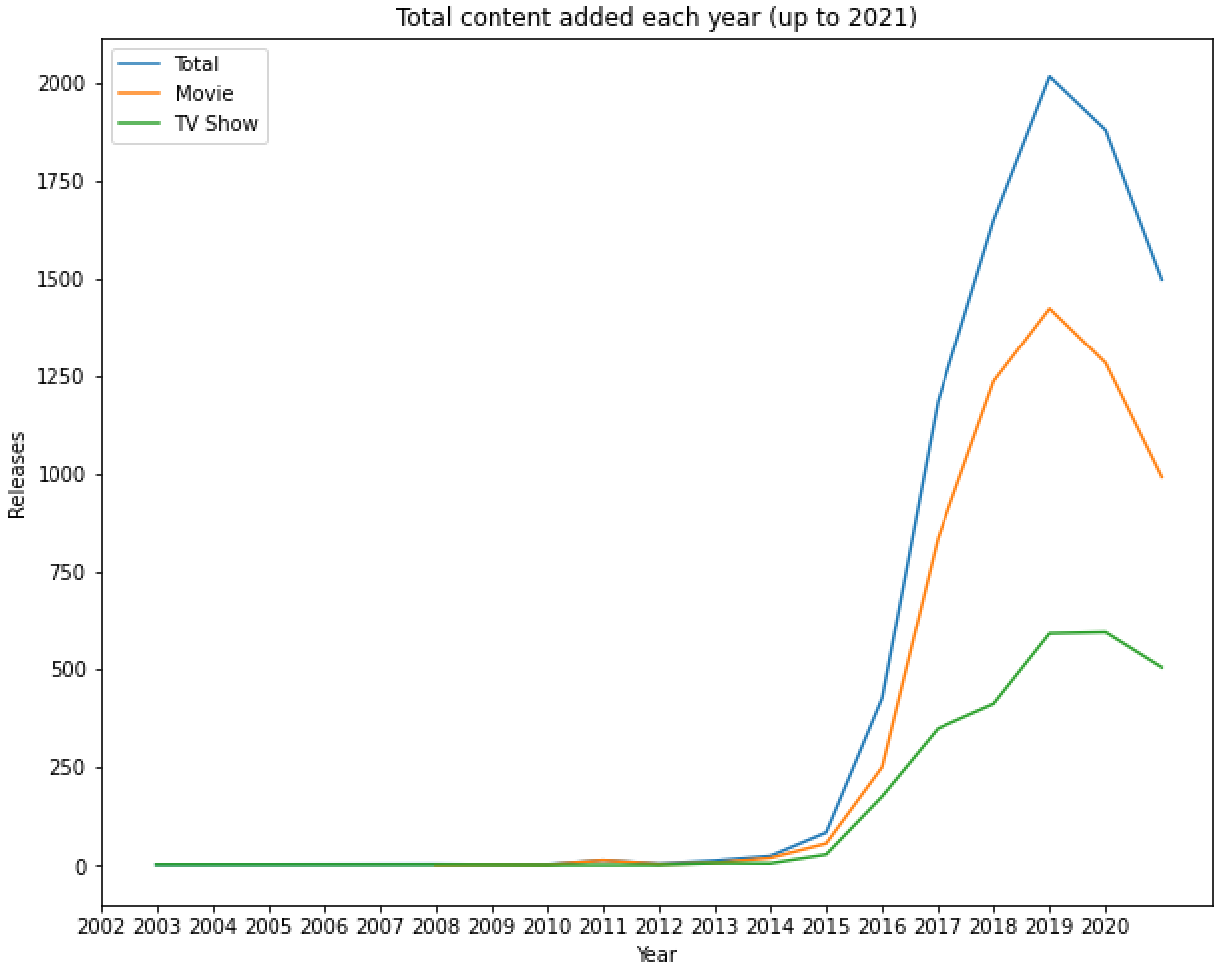
Out[24]:

	year	count
0	2019	2016
1	2020	1879
2	2018	1649
3	2021	1498
4	2017	1185
5	2016	427
6	2015	84
7	2014	24
8	2011	13
9	2013	12
10	2012	4
11	2008	3
12	2003	2
13	2010	2
14	2009	2

In [25]:

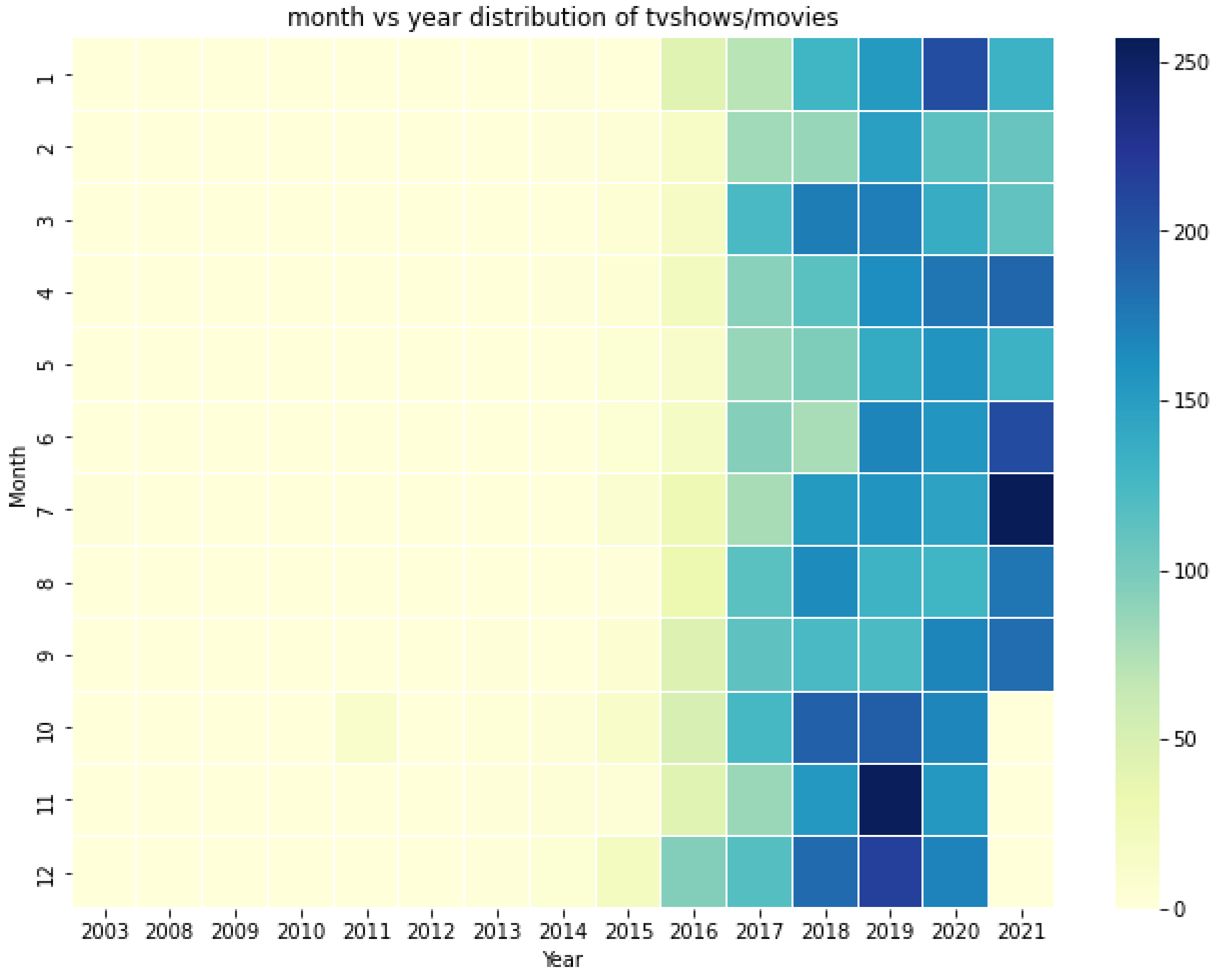
```
netflix_year2 = df[['type','year_added']]
movie_year = netflix_year2[netflix_year2['type']=='Movie'].year_added.value_count
movie_year = movie_year[movie_year.year != 2022]
show_year = netflix_year2[netflix_year2['type']=='TV Show'].year_added.value_coun
show_year = show_year[show_year.year != 2022]
```

In [26]:
```
fig, ax = plt.subplots(figsize=(10, 8))
sns.lineplot(data=netflix_year, x='year', y='count')
sns.lineplot(data=movie_year, x='year', y='count')
sns.lineplot(data=show_year, x='year', y='count')
ax.set_xticks(np.arange(2002, 2021, 1))
plt.title("Total content added each year (up to 2021)")
plt.legend(['Total','Movie','TV Show'])
plt.ylabel("Releases")
plt.xlabel("Year")
plt.show()
```



Based on the above timeline, we can see that netflix grown after the 2015 Since then, the amount of content added has been tremendous. I decided to exclude content added during 2021 since the data does not include a full years worth of data. We can see that there has been a consistent growth in the number of movies on Netflix compared to shows.

In [27]:
```
month_year_df = df.groupby('year_added')['month_added'].value_counts().unstack().
month_year_df
plt.figure(figsize=(11,8))
sns.heatmap(month_year_df, linewidths=0.025, cmap="YlGnBu")
plt.title("month vs year distribution of tvshows/movies")
plt.ylabel("Month")
plt.xlabel("Year")
plt.show()
```



It can be infered like post the december 2016 contents are getting added more and in month of 2021 it has most content added

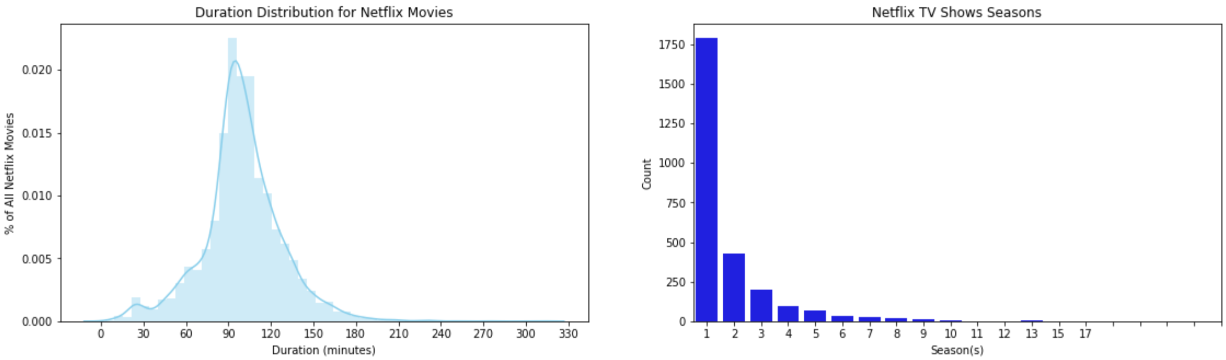
It can also be infered best period for addition around the end of year (oct-dec) from last 3-years data

Netflix seasons and movies durations analysis

In [28]:

```
fig, ax = plt.subplots(1,2, figsize=(19, 5))
g1 = sns.distplot(movies_df.duration, color='skyblue',ax=ax[0],kde=True)
g1.set_xticks(np.arange(0,360,30))
g1.set_title("Duration Distribution for Netflix Movies")
g1.set_ylabel("% of All Netflix Movies")
g1.set_xlabel("Duration (minutes)")
g2 = sns.countplot(tv_shows_df.seasons, color='b',ax=ax[1])
g2.set_xticks(np.arange(0,20,1))
g2.set_title("Netflix TV Shows Seasons")
g2.set_ylabel("Count")
g2.set_xlabel("Season(s)")
fig.show()
```

```
D:\Anaconda\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `di
stplot` is a deprecated function and will be removed in a future version. Pleas
e adapt your code to use either `displot` (a figure-level function with similar
flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
D:\Anaconda\lib\site-packages\seaborn\_decorators.py:43: FutureWarning: Pass th
e following variable as a keyword arg: x. From version 0.12, the only valid pos
itional argument will be `data`, and passing other arguments without an explici
t keyword will result in an error or misinterpretation.
  FutureWarning
D:\Anaconda\lib\site-packages\ipykernel_launcher.py:12: UserWarning: Matplotlib
is currently using module://ipykernel.pylab.backend_inline, which is a non-GUI
backend, so cannot show the figure.
  if sys.path[0] == '':
```

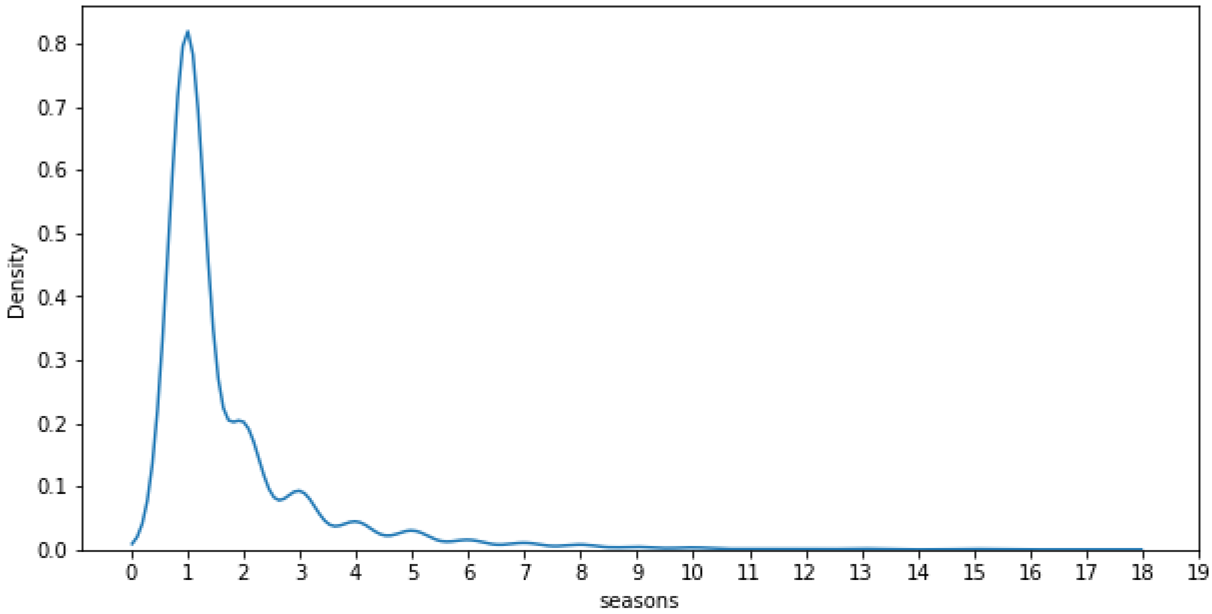


Movies duration is almost like a normally distributed and around most of the movies have duration of around 90 minutes

For TV Shows most of the shows have 1 season only

In [30]:
```
fig1=plt.figure(figsize=(10,5))
g1=sns.kdeplot(data=tv_shows_df,x='seasons')
g1.set_xticks(np.arange(0,20,1))
fig1.show()
```

```
D:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: UserWarning: Matplotlib
is currently using module://ipykernel.pylab.backend_inline, which is a non-GUI
backend, so cannot show the figure.
  after removing the cwd from sys.path.
```



Stastical Summary of Movies and TV shows data frame

In [36]:
```
tv_shows_df['seasons'].describe()
```

Out[36]:
```
count    2674.000000
mean        1.765520
std         1.583206
min         1.000000
25%         1.000000
50%         1.000000
75%         2.000000
max        17.000000
Name: seasons, dtype: float64
```

In [37]:

```
movies_df['duration'].describe()
```

Out[37]:

```
count    6126.000000
mean       99.584884
std        28.283225
min         3.000000
25%        87.000000
50%        98.000000
75%       114.000000
max       312.000000
Name: duration, dtype: float64
```

Average duration of TV shows and movies

In [30]:

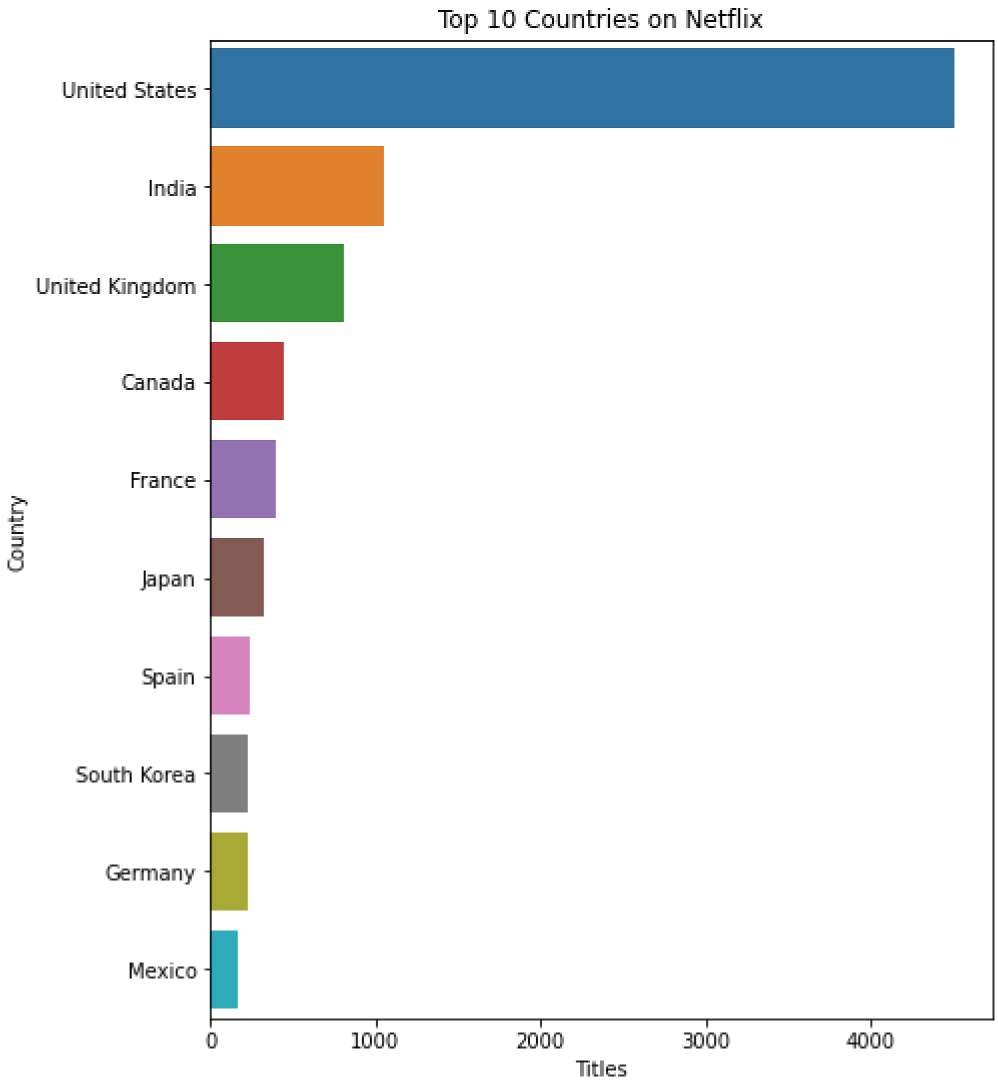
```
print ("Averge duration for movies on netflix",movies_df.duration.mean())
print ("Average no of seasons for tv shows  on netflix ",tv_shows_df.seasons.mean
```

```
Averge duration for movies on netflix 99.58488410055502
Average no of seasons for tv shows  on netflix  1.7655198204936424
```

Top 10 Countries with having max content

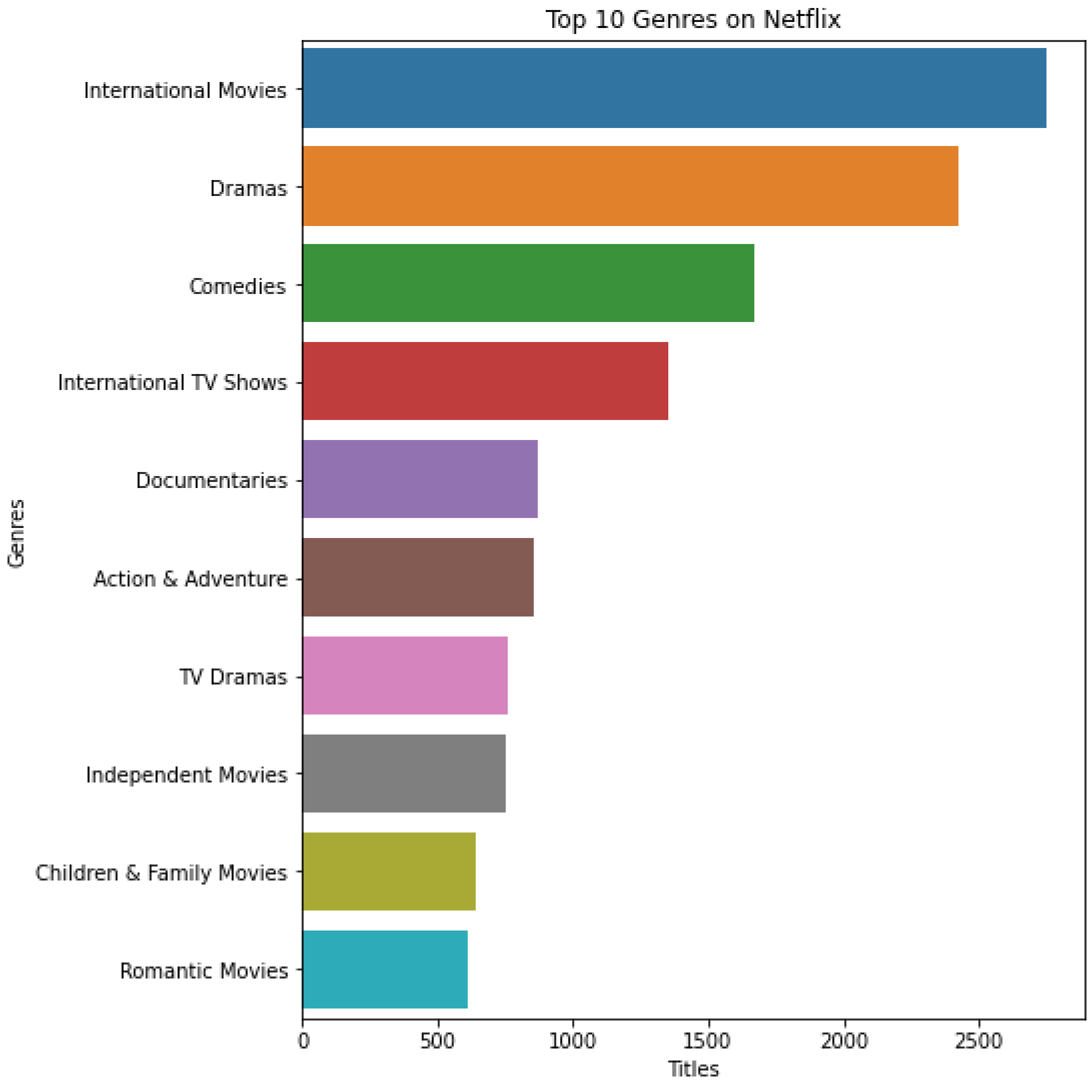
In [31]:

```
filtered_countries = df.set_index('title').country.str.split(', ', expand=True).s
plt.figure(figsize=(7,9))
g = sns.countplot(y = filtered_countries, order=filtered_countries.value_counts()
plt.title('Top 10 Countries on Netflix')
plt.xlabel('Titles')
plt.ylabel('Country')
plt.show()
```



Top 10 Popular Genre

```
In [32]: filtered_genres = df.set_index('title').listed_in.str.split(', ', expand=True).st
         plt.figure(figsize=(7,9))
         g = sns.countplot(y = filtered_genres, order=filtered_genres.value_counts().index
         plt.title('Top 10 Genres on Netflix')
         plt.xlabel('Titles')
         plt.ylabel('Genres')
         plt.show()
```

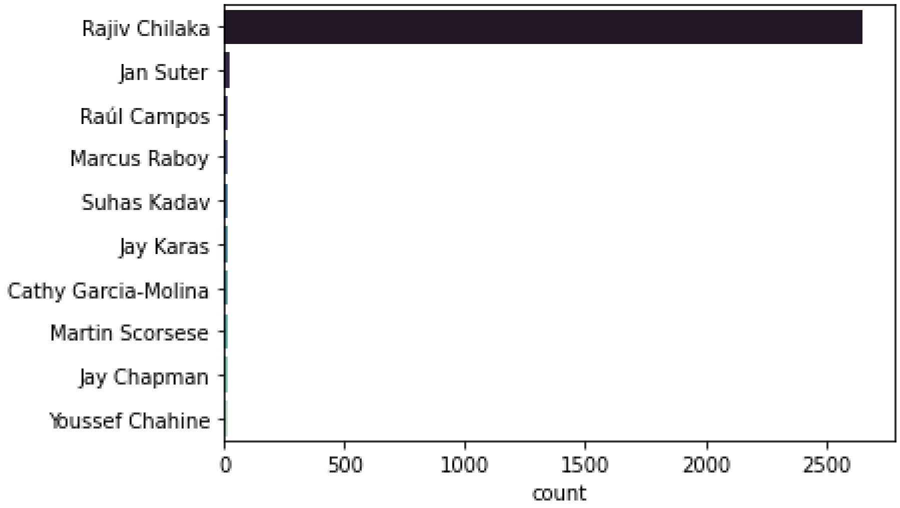


Based on above data ,In terms of genres, international movies are most followed by dramas and comedies.

Even though the United States has the most content available, it looks like Netflix has decided to release a ton of international movies. The reason for this could be that most Netflix subscribers aren't actually in the United States, but rather the majority of viewers are actually international subscribers.

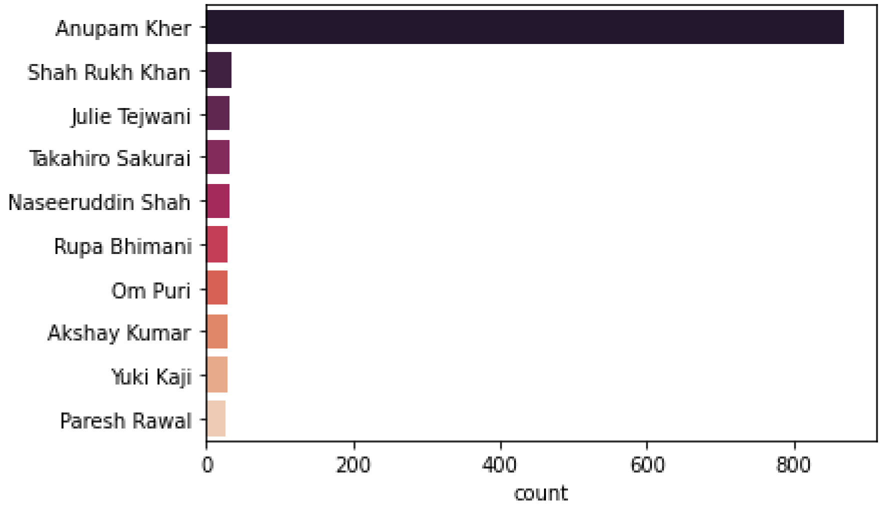
Top 10 directors on Netflix with the most releases

```
In [33]: filtered_directors = df.set_index('title').director.str.split(', ', expand=True).
         sns.countplot(y = filtered_directors, order=filtered_directors.value_counts().ind
         plt.show()
```



Top 10 actors on Netflix

```
In [34]: filtered_cast = df.set_index('title').cast.str.split(', ', expand=True).stack().r
         sns.countplot(y = filtered_cast, order=filtered_cast.value_counts().index[:10], p
         plt.show()
```

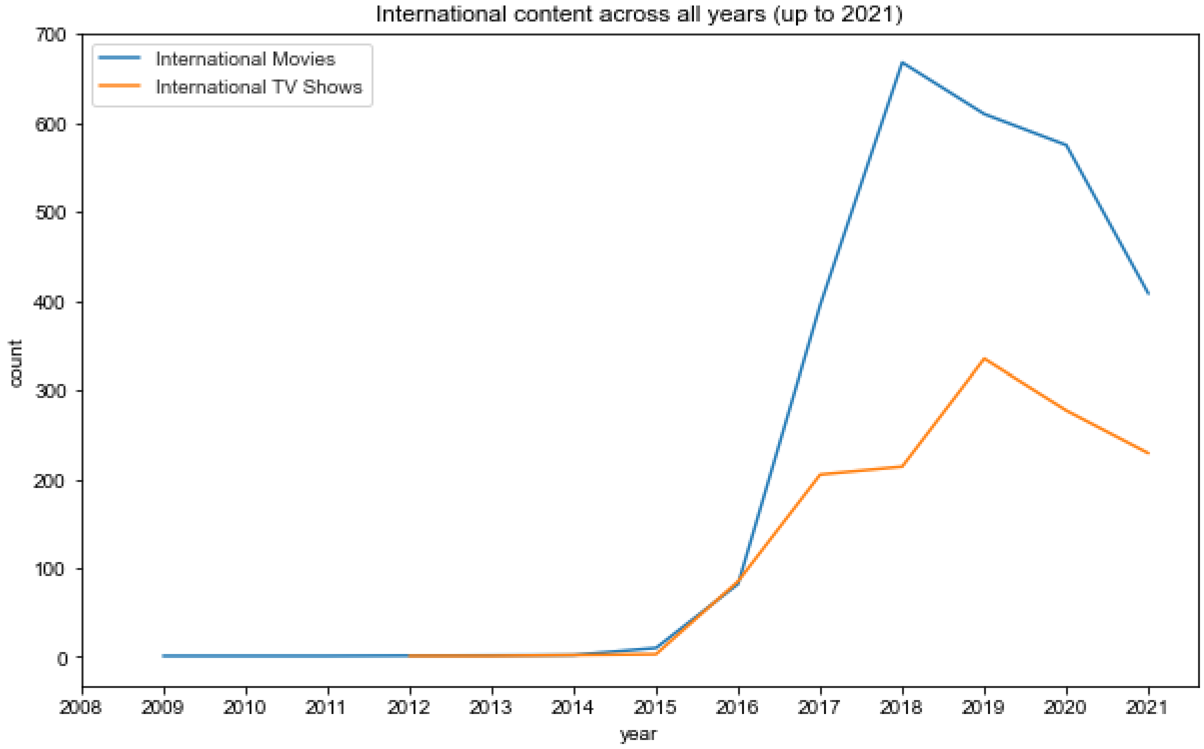


In [35]:

```
international_movies = df[df['listed_in'].str.contains('International Movies')]
intmov_year = international_movies['year_added'].value_counts().to_frame().reset_
intmov_year = intmov_year[intmov_year.year != 2022]

international_shows = df[df['listed_in'].str.contains('International TV Shows')]
intshow_year = international_shows['year_added'].value_counts().to_frame().reset_
intshow_year = intshow_year[intshow_year.year != 2022]

fig, ax = plt.subplots(figsize=(10, 6))
fig.add_gridspec(2, 2)
sns.set_style("whitegrid")
sns.lineplot(data=intmov_year, x='year', y='count')
sns.lineplot(data=intshow_year, x='year', y='count')
ax.set(xticks=np.arange(2008, 2022, 1))
plt.title("International content across all years (up to 2021)")
plt.legend(['International Movies','International TV Shows'])
plt.show()
```



It can be seen there is growth till the 2018 in both international movies and tv shows but there was tv shows still growing till 2019 but after that there is dip in both movies and tv shows

Getting the popular director and actor combination in india for movies

In [36]:
```
new_df = df[(df['country']=='India') & (df['type']=='Movie')].copy()
new_df.reset_index(inplace=True)
df_final=new_df.assign(cast=new_df['cast'].str.split(', ')).explode('cast')
df_final2=df_final.assign(director=new_df['director'].str.split(', ')).explode('d

df_final2.reset_index(inplace=True)

df_cast1=df_final2[['title','director','cast']]
df_cast1.reset_index(inplace=True)

df_cast_final=df_cast1.merge(df_final2, on=['director','cast'], how='inner')

df_cast_final.groupby(by=['director','cast'])['title_x'].nunique().sort_values().
```

Out[36]:
```
director            cast
Karan Johar         Rani Mukerji     4
Rajiv Chilaka       Julie Tejwani    4
Prakash Jha         Ajay Devgn       4
Sooraj R. Barjatya  Mohnish Bahl     4
David Dhawan        Salman Khan      4
Priyadarshan        Rajpal Yadav     4
Rajiv Chilaka       Rajesh Kava      4
Sooraj R. Barjatya  Salman Khan      5
                    Alok Nath        5
David Dhawan        Anupam Kher      6
Name: title_x, dtype: int64
```

David Dhawan and Anupam Kher combo is most popular in indian movies

Get the Count of the total movies or tv shows in comedy Genre

In [37]:
```
comedy_df = df[(df['listed_in'].str.contains('Comedies','TV Comedies'))].copy()
comedy_df['title'].count()
```

Out[37]:
```
2254
```

Popular TV Show director on netflix

In [38]:

```
direct_df=df[df['type']=='TV Show'].copy()
direct_df=direct_df.assign(director=direct_df['director'].str.split(', ')).explod
direct_df['director'].value_counts().head(10)
```

Out[38]:

```
Rajiv Chilaka             2444
Alastair Fothergill          3
Ken Burns                    3
Hsu Fu-chun                  2
Stan Lathan                  2
Gautham Vasudev Menon        2
Jung-ah Im                   2
Shin Won-ho                  2
Joe Berlinger                2
Iginio Straffi               2
Name: director, dtype: int64
```

Top ten kind of genre usually japnese are watching

In [39]:

```
jap_df=df[df['country']=='Japan'].copy()
jap_df=jap_df.assign(listed_in=jap_df['listed_in'].str.split(', ')).explode('list
jap_df['listed_in'].value_counts()[:10]
```

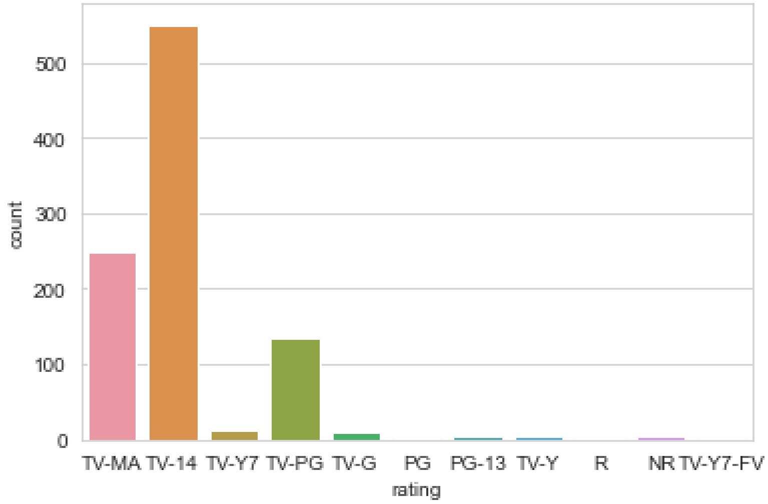
Out[39]:

```
International TV Shows    140
Anime Series              130
International Movies       58
Anime Features             54
Action & Adventure         46
Romantic TV Shows          21
TV Dramas                  20
Crime TV Shows             15
Teen TV Shows              14
Kids' TV                   13
Name: listed_in, dtype: int64
```

Target Audience for the India

```
In [40]: india_df=df[df['country']=='India'].copy()
         #india_df['rating'].value_counts()
         sns.countplot(data=india_df,x='rating')
         #It can be infered like TV-14 category is more
```

```
Out[40]: <AxesSubplot:xlabel='rating', ylabel='count'>
```

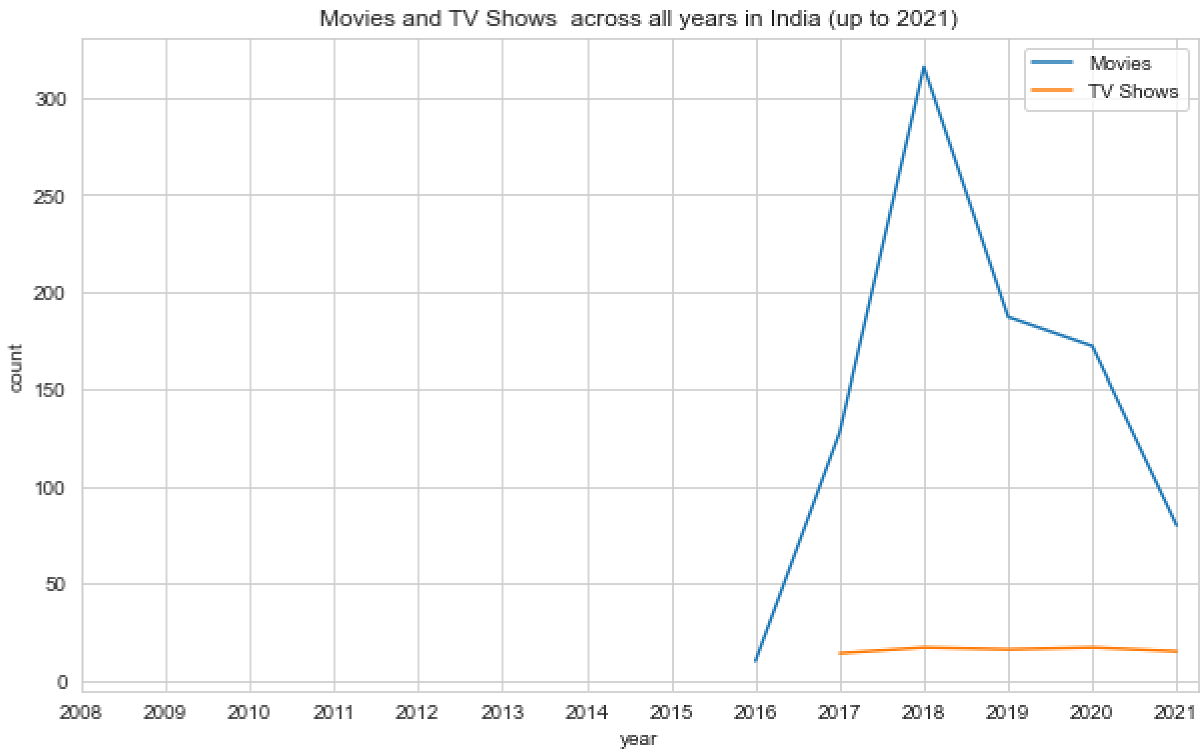


Trend of TV Shows and Movies over the year in India

In [41]:
```
india_movies = df[(df['type']=='Movie') &(df['country']=='India')].copy()
indmov_year = india_movies['year_added'].value_counts().to_frame().reset_index().
indmov_year = indmov_year[indmov_year.year != 2022]

india_shows = df[(df['type']=='TV Show') &(df['country']=='India')].copy()
indshow_year = india_shows['year_added'].value_counts().to_frame().reset_index().
indshow_year = indshow_year[indshow_year.year != 2022]

fig, ax = plt.subplots(figsize=(10, 6))
fig.add_gridspec(2, 2)
sns.set_style("whitegrid")
sns.lineplot(data=indmov_year, x='year', y='count')
sns.lineplot(data=indshow_year, x='year', y='count')
ax.set(xticks=np.arange(2008, 2022, 1))
plt.title("Movies and TV Shows  across all years in India (up to 2021)")
plt.legend(['Movies','TV Shows'])
plt.show()
```

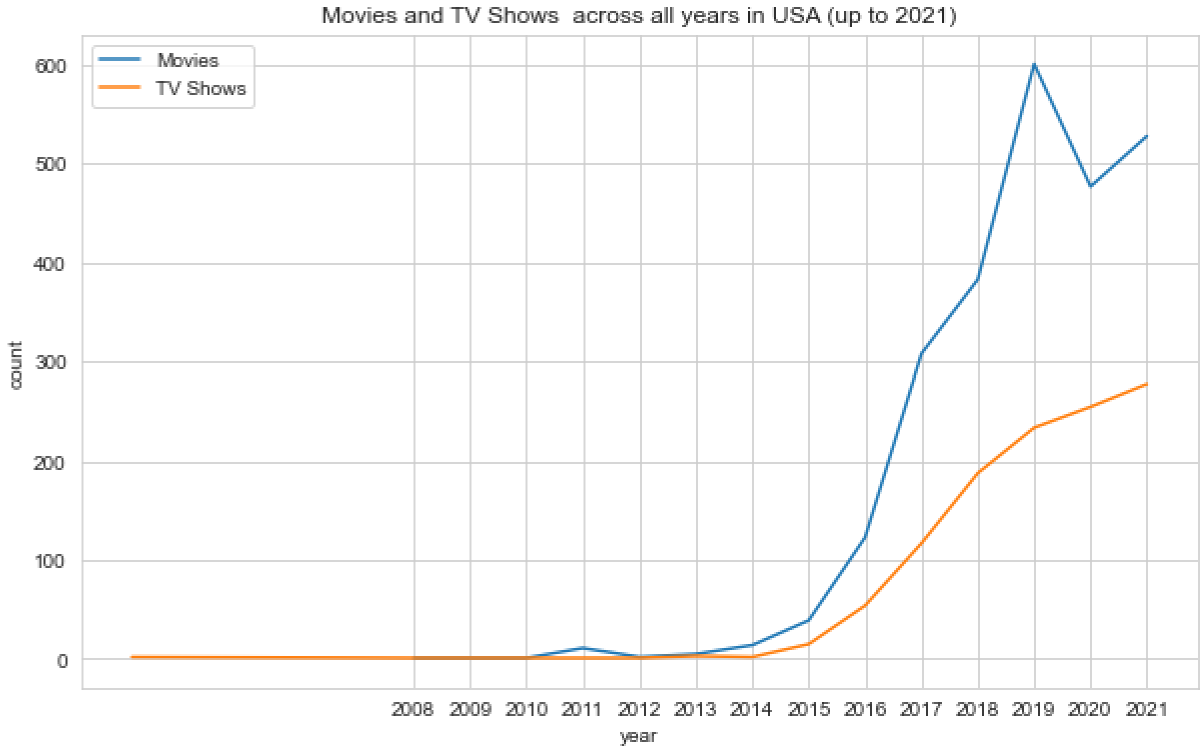


Trend of TV Shows and Movies over the year in USA

In [42]:
```
us_movies = df[(df['type']=='Movie') &(df['country']=='United States')].copy()
usmov_year = us_movies['year_added'].value_counts().to_frame().reset_index().rena
usmov_year = usmov_year[usmov_year.year != 2022]

us_shows = df[(df['type']=='TV Show') &(df['country']=='United States')].copy()
usshow_year = us_shows['year_added'].value_counts().to_frame().reset_index().rena
usshow_year = usshow_year[usshow_year.year != 2022]

fig, ax = plt.subplots(figsize=(10, 6))
fig.add_gridspec(2, 2)
sns.set_style("whitegrid")
sns.lineplot(data=usmov_year, x='year', y='count')
sns.lineplot(data=usshow_year, x='year', y='count')
ax.set(xticks=np.arange(2008, 2022, 1))
plt.title("Movies and TV Shows  across all years in USA (up to 2021)")
plt.legend(['Movies','TV Shows'])
plt.show()
```



Based on the above inference in India TV Shows are getting constantly with less or no change are getting added every year but for movies post 2018 there is constant dip in addition every year so here focus must be how to increase the same .

In USA there was dip in movies addition in year2019-2020 but after that it has increasing seemlessless and also TV Shows addition increasing every year

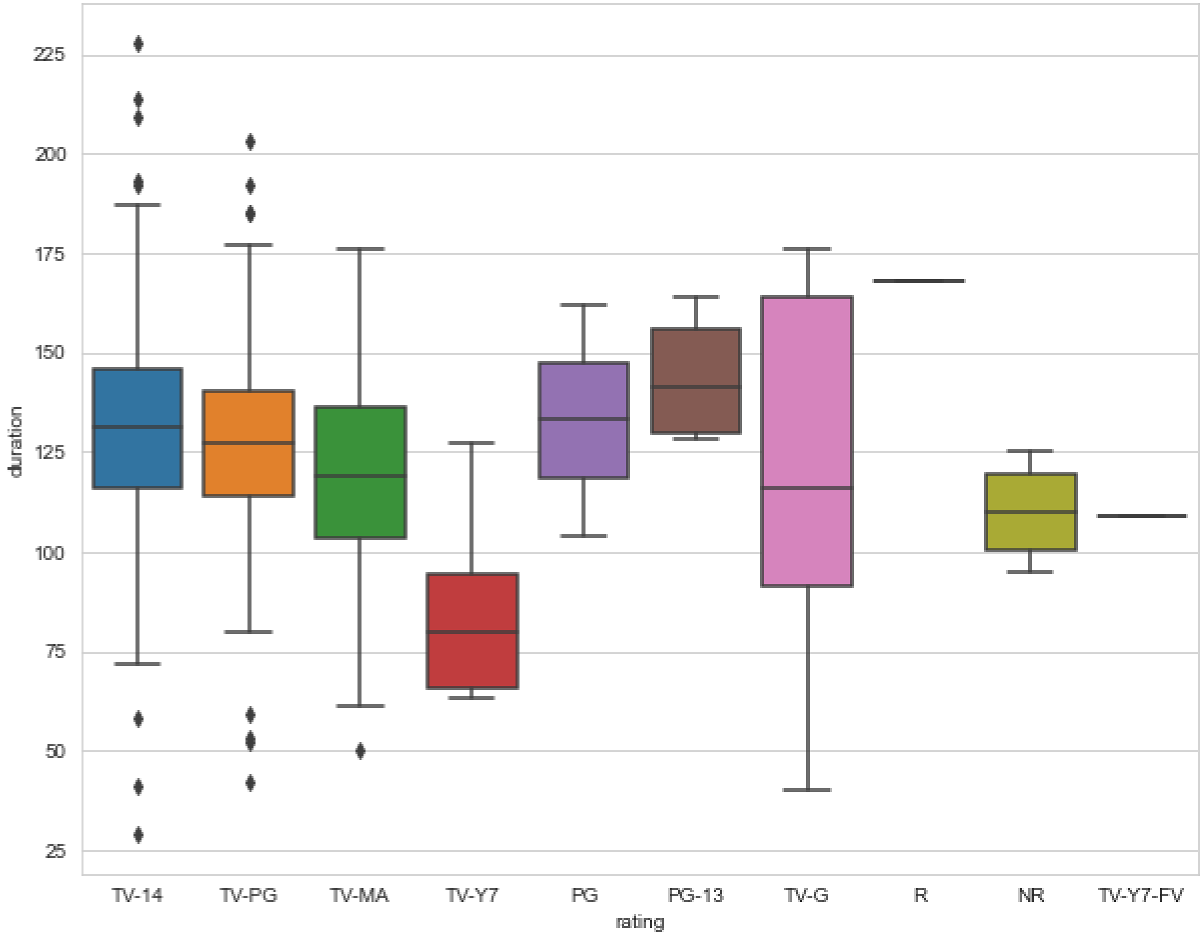
Compare to USA In India addition of TV Shows and Movies are less so Netflix should focus how it can increae it

In [43]:

```
india_mov_df= movies_df[movies_df['country']=='India'].copy()
plt.figure(figsize=(10,8))
sns.boxplot(data=india_mov_df,x='rating',y='duration')
```

Out[43]:

```
<AxesSubplot:xlabel='rating', ylabel='duration'>
```



Analysing directors of india who directed tv shows as well as movies

In [44]:

```
india_df_dir=df[(df['country']=='India')].copy()
india_df_dir=india_df_dir.assign(director=india_df_dir['director'].str.split(', '
res_df=india_df_dir[['type','director']]
res_df.reset_index(inplace=True)
list1=res_df[res_df['type']=='TV Show']['director'].unique()
list2=res_df[res_df['type']=='Movie']['director'].unique()
print(list(set(list1) & set(list2)))
```

```
['Sarjun', 'B. V. Nandini Reddy', 'Priyadarshan', 'Karthik Subbaraj', 'Bejoy Na
mbiar', 'Soumendra Padhi', 'Rajiv Chilaka']
```

Analysing cast of india who worked in tv shows as well as movies

In [45]:

```
india_df_act=df[(df['country']=='India')].copy()
india_df_act=india_df_act.assign(cast=india_df_act['cast'].str.split(', ')).explo
res_df_act=india_df_act[['type','cast']]
res_df_act.reset_index(inplace=True)
list1=res_df_act[res_df_act['type']=='TV Show']['cast'].unique()
list2=res_df_act[res_df_act['type']=='Movie']['cast'].unique()
print(list(set(list1) & set(list2)))
print("=======================================")
print ("Count of actors of India who worked in both tv shows and movies: ",len(li
```

```
['Vatsal Dubey', 'Kishore Kumar G.', 'Luke Kenny', 'Ravi Kishan', 'Shahana Gosw
ami', 'Gashmeer Mahajani', 'Adarsh Gourav', 'Mahesh Manjrekar', 'Manish Chaudha
ry', 'Siddharth Menon', 'Hassan Zaidi', 'Kashmira Irani', 'Eesha Rebba', 'Akash
Khurana', 'Karuna Pandey', 'Yashaswini Dayama', 'Jigna Bhardwaj', 'Neena Gupt
a', 'Ashok Selvan', 'Suparna Marwah', 'Plabita Borthakur', 'Seema Biswas', 'Raj
it Kapoor', 'Suriya', 'Swati Rajput', 'Ashish Verma', 'Vikas Kumar', 'Mukul De
v', 'Mona Singh', 'Amala Paul', 'Sharad Kelkar', 'Anisha Ambrose', 'Rasika Duga
l', 'Neil Bhoopalam', 'Purab Kohli', 'Huma Qureshi', 'Akshay Oberoi', 'Satya De
v', 'Arif Zakaria', 'Shruti Seth', 'Kavin Dave', 'Karanvir Malhotra', 'Rahul Bo
se', 'Siddarth', 'Mallika Dua', 'Alam Khan', 'Jigna Bharadhwaj', 'Mandira Bed
i', 'Viineet Kumar', 'Yogi Babu', 'Puru Chibber', 'Julie Tejwani', 'Vasundhara
Kaul', 'Chandan Anand', 'Sheeba Chaddha', 'Dibyendu Bhattacharya', 'Chaitanya S
harma', 'Vikramjeet Virk', 'Rajesh Kawa', 'Sanghmitra Hitaishi', 'Manav Gohil',
'Anamaya Verma', 'Adil Hussain', 'Sonal Kaushal', 'Arvind Swamy', 'Bipasha Bas
u', 'Rannvijay Singh', 'Danish Husain', 'Barkha Singh', 'Emraan Hashmi', 'Ragin
i Khanna', 'Manoj Pahwa', 'Sanjeeda Sheikh', 'Rupa Bhimani', 'Danish Hussain',
'Vijay Varma', 'Suchitra Pillai', 'Vinay Pathak', 'Gulfam Khan', 'Rajesh Kava',
'Anurag Arora', 'Arun Kumar', 'Simran', 'Aruna Irani', 'Syna Anand', 'Siddhart
h', 'Rohit Saraf', 'Ashwath Bhatt', 'Vir Das', 'Shefali Shah', 'Swapnil', 'Pray
aga Martin', 'Manav Kaul', 'Nayan Shukla', 'Mohit Raina', 'Jishu Sengupta', 'Sa
njay Suri', 'Kavi Shastri', 'Raza Murad', 'Aaditi Pohankar', 'Akarsh Khurana',
'Anupam Kher', 'Satyadeep Misra', 'Sabina Malik', 'Kishore', 'Pujitha Ponnada',
'Rahul Bagga', 'Kirti Kulhari', 'Pooja Bhatt', 'Anjali', 'Revathy', 'Apoorva Ar
ora', 'Rajesh Sharma', 'Ratna Pathak Shah', 'Vijay Sethupathi', 'Anish Kuruvill
a', 'Siddhant Karnick', 'Kalki Koechlin', 'Sarita Joshi', 'Sameer Kochhar', 'Re
gina Cassandra', 'Radhika Apte', 'Pankaj Tripathi', 'Ranvir Shorey', 'Vishesh B
ansal', 'Mayur More', 'Amrita Bagchi', 'Nedumudi Venu', 'Sobhita Dhulipala', 'M
uskkaan Jaferi', 'Manjot Singh', 'Jitendra Kumar', 'Rahul Khanna', 'Suhail Nayy
ar', 'Mona Ambegaonkar', 'Rajesh Tailang', 'Mousam', 'Parvathy', 'Ratnabali Bha
ttacharjee', 'Suhasini Mulay', 'Bobby Simha', 'Amit Sial', 'Manchu Lakshmi', 'G
eetika Tyagi', 'Srinivas Avasarala', 'Prasanna', 'Sameer Dharmadhikari', 'Giris
h Kulkarni', 'Anupam Bhattacharya', 'Prakash Raj', 'Jitendra Joshi', 'Inaamulha
q', 'Pushtii Shakti', 'Mithila Palkar', 'Indraneil Sengupta', 'Amol Parashar',
'Jaideep Ahlawat', 'Jagapathi Babu', 'Amruta Subhash']
=======================================
Count of actors of India who worked in both tv shows and movies:  159
```

Final Inference or conclusion

It's clear that Netflix has grown over the years. We can see it from the data that the company took certain approaches in their marketing strategy to break into new markets around the world. Netflix now focusing more on international market as there are quite good amount

of susbscribers around the world . In this case, we can see that a good amount of international movies and TV shows were added over the years as part of Netflix's global expansion. Here Netflix should focus outside USA and other countries where addition of TV Shows and Movies are less. It is quite clear that country like India is growing market for Netflix so Netflix need to focus more in growing business in countries like India .